

A Comprehensive Analysis of Deep Learning for Interference Suppression, Sample and Model Complexity in Wireless Systems

Taiwo R. Oyedare

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Jeffrey H. Reed, Chair

Daniel J. Jakubisin, Co-chair

Carl B. Dietrich

Yaling Yang

Xi Chen

Lingjia Liu

February 6, 2024

Arlington, Virginia

Keywords: Deep Learning, Wireless Communications, Model Complexity, Interference
Suppression

Copyright 2024, Taiwo R. Oyedare

A Comprehensive Analysis of Deep Learning for Interference Suppression, Sample and Model Complexity in Wireless Systems

Taiwo R. Oyedare

ABSTRACT

The wireless spectrum is limited and the demand for its use is increasing due to technological advancements in wireless communication, resulting in persistent interference issues. Despite progress in addressing interference, it remains a challenge for effective spectrum usage, particularly in the use of license-free and managed shared bands and other opportunistic spectrum access solutions. Therefore, efficient and interference-resistant spectrum usage schemes are critical. In the past, most interference solutions have relied on avoidance techniques and expert system-based mitigation approaches. Recently, researchers have utilized artificial intelligence/machine learning techniques at the physical (PHY) layer, particularly deep learning, which suppress or compensate for the interfering signal rather than simply avoiding it. In addition, deep learning has been utilized by researchers in recent years to address various difficult problems in wireless communications such as, transmitter classification, interference classification and modulation recognition, amongst others. To this end, this dissertation presents a thorough analysis of deep learning techniques for interference classification and suppression, and it thoroughly examines complexity (sample and model) issues that arise from using deep learning. First, we address the knowledge gap in the literature with respect to the state-of-the-art in deep learning-based interference suppression. To account for the limitations of deep learning-based interference suppression techniques, we discuss several challenges, including lack of interpretability, the stochastic nature of the

wireless channel, issues with open set recognition (OSR) and challenges with implementation. We also provide a technical discussion of the prominent deep learning algorithms proposed in the literature and also offer guidelines for their successful implementation. Next, we investigate convolutional neural network (CNN) architectures for interference and transmitter classification tasks. In particular, we utilize a CNN architecture to classify interference, investigate model complexity of CNN architectures for classifying homogeneous and heterogeneous devices and then examine their impact on test accuracy. Next, we explore the issues with sample size and sample quality with regards to the training data in deep learning. In doing this, we also propose a rule-of-thumb for transmitter classification using CNN based on the findings from our sample complexity study. Finally, in cases where interference cannot be avoided, it is important to suppress such interference. To achieve this, we build upon autoencoder work from other fields to design a convolutional neural network (CNN)-based autoencoder model to suppress interference thereby ensuring coexistence of different wireless technologies in both licensed and unlicensed bands.

A Comprehensive Analysis of Deep Learning for Interference Suppression, Sample and Model Complexity in Wireless Systems

Taiwo R. Oyedare

GENERAL AUDIENCE ABSTRACT

Wireless communication has advanced a lot in recent years, but it is still hard to use the limited amount of available spectrum without interference from other devices. In the past, researchers tried to avoid interference using expert systems. Now, researchers are using artificial intelligence and machine learning, particularly deep learning, to mitigate interference in a different way. Deep learning has also been used to solve other tough problems in wireless communication, such as classifying the type of device transmitting a signal, classifying the signal itself or avoiding it. This dissertation presents a comprehensive review of deep learning techniques for reducing interference in wireless communication. It also leverages a deep learning model called convolutional neural network (CNN) to classify interference, and investigates how the complexity of the CNN affects its performance. It also looks at the relationship between model performance and dataset size (i.e., sample complexity) in wireless communication. Finally, it discusses a CNN-based autoencoder technique to suppress interference in digital amplitude-phase modulation system. All of these techniques are important for making sure different wireless technologies can work together in both licensed and unlicensed bands.

This dissertation is dedicated to my late Father: Titus Olubunmi Oyedare.

Acknowledgments

I would like to express my sincere gratitude to my dissertation advisors, Dr. Jeff Reed and Dr. Jakubisin, for their invaluable guidance and support throughout this research project. Their expertise and guidance have been instrumental in helping me navigate the research process and shape my work into its final form. It is truly an honor to be advised by them. I am very grateful for both of your friendship and counsel all through my time in the PhD program. Their relentless commitment and positive reinforcement has helped shape me into a productive and effective researcher. Also, I would like to thank Dr. Jerry Park for his guidance in my early PhD days. I learnt a lot from my time in the WINSER group.

I would also like to specially thank the members of my dissertation committee, Dr. Carl Dietrich, Dr. Yaling Yang, Dr. Xi Chen and Dr. Lingjia, for their valuable feedback and insights during the research and writing process. I thank each of you for your rigor and attentiveness. Special appreciation also goes to Dr. Vijay Shah and Badhrinarayan Malolan for their collaboration in several experiments and research during this PhD program. I would also like to thank my mentor: Dr. Paul Ampadu, whose mentorship means more than the world to me. Dr. Ampadu's zeal and enthusiasm is second to none. Thank you for all that you do for not just me but several other students in the ECE program.

I would also like to acknowledge U.S. Air Force Research Laboratory (AFRL), Lockheed Martin Corporation and the National Science Foundation (NSF) for their financial support of this research project. Their contribution has made it possible for me to complete this work.

Further, I would like to extend my heartfelt thanks to my wife: *Abisola (a.k.a. Cupcake)*, my daughter: *Zoe*, my twin sister: *Kehinde*, my mum: *Maria*, my brother: *Timmy*, and my

big sister: *Bukola*, for their unwavering support and encouragement. This dissertation would not have been possible without their love and support. I am eternally grateful! Thank you, Cupcake for your support and always hyping me in the dark days of imposter syndrome. You are my God-sent!

Finally, I am deeply grateful to my colleagues and friends who have supported me throughout this journey, particularly Jinshan Liu and Gaurang Naik, who provided invaluable feedback on my work. I will also like to thank Dr. Trey Waller for his support when things became tough. My sincere gratitude also goes to Hilda Reynolds for all her wonderful notes and encouragements. Special and profound gratitude to the all my colleagues, friends and associates in and outside the US: Dr. Ali Reza Famili, Dr. Hanif Rahbari, Hilda Reynolds, Dr. Vireshwar Kumar, Dr. Omotayo Helen Ajao, Dr. Aanuoluwapo Ojelade, Adetoso Afonja, Tolu Adekanye, Dr. Dayo Omosa, Dr. Tosin Ogunmayowa, Dr. Damilola Alegbeleye, Yenum Nduka, Tarun Cousik, Dr. Seyi Percy Falekun, Dr. Oluwafemi Ifetayo, Dr. Annie Patrick, Dr. Pratham Oza, Dr. Tanmaya, Misra, Oluwasegun Adeleye, Niyi Adekanmbi, Pastors Will & Juniece Fillingham, Lucky Agbotiname Imoize, Olumide Esho and Williams Matthew. If I start to mention how each of you have supported me, I would write several pages of acknowledgment. I am thankful for all the chats, insights, ideas, advice, thoughts that we shared together during my PhD study. I want you all to know that I am immensely grateful for knowing each and everyone of you.

Contents

List of Figures	xv
List of Tables	xix
List of Abbreviations	xx
1 Introduction	1
1.1 Interference Suppression Using Deep Learning: Current Approaches and Open Challenges	6
1.1.1 Related Publications	8
1.2 Keep It Simple: CNN Model Complexity Studies for Interference Classification Tasks	8
1.2.1 Related Publications	10
1.3 Sample Complexity Studies in Wireless Communications	11
1.3.1 Estimating the Required Training Dataset Size for Transmitter Classification Using Deep Learning	11
1.3.2 Relationship between Model Performance and Sample Size in Wireless Communications Applications of Deep Learning	13
1.3.3 Detecting Out-of-Distribution Data in Wireless Communications Applications of Deep Learning	14

1.3.4	Related Publications	14
1.4	An End-to-End CNN-AE Interference Suppression System for Wireless Com- munications	15
1.4.1	Related Publications	18
2	Interference Classification and Suppression Using Deep Learning: Current Approaches and Open Challenges	19
2.1	Introduction	19
2.2	Related Works	21
2.3	Technical Background	24
2.3.1	Interference Types and Characteristics	24
2.3.2	Classification of Interference Suppression Techniques	29
2.3.3	Deep Learning Fundamentals	31
2.4	Publicly Available Interference Dataset	38
2.4.1	Radar Interference Dataset	40
2.4.2	Radio Frequency Interference Dataset	40
2.4.3	CRAWDAD Dataset	41
2.4.4	Wireless Interference Signal Dataset	42
2.5	Deep Learning-Based Interference Classification Techniques	42
2.5.1	Contributions for Interference Classification Using Supervised Learning	43

2.5.2	Contributions for Interference Classification Using Semi-Supervised Learning	47
2.6	Deep Learning-Based Interference Suppression Techniques	48
2.6.1	Contributions for Interference Suppression Using CNN	48
2.6.2	Contributions for Interference Suppression Using RNN	54
2.6.3	Contributions for Interference Suppression Using Autoencoders	56
2.6.4	Contributions for Interference Suppression Using DNNs	58
2.7	Comparison Between the Approaches	64
2.7.1	Application/Interference Type	64
2.7.2	Deep Learning Algorithm	64
2.7.3	Dataset Type	66
2.7.4	Performance Metrics	66
2.7.5	Benefits and Drawbacks of Deep Learning-based Interference suppression Schemes	70
2.8	Open Challenges	71
2.8.1	Lack of Interpretability of Deep Learning Algorithms	71
2.8.2	The Stochasticity of the Wireless Channel	71
2.8.3	Need for Time and Model Complexity Studies	73
2.8.4	Lack of Performance Comparison between Online and Offline training	74
2.8.5	Performance Analysis	74

2.8.6	Issues with Open Set Recognition (OSR) or Out-of-Distribution (OOD)	
	Data	76
2.8.7	Challenges with Implementation	77
2.9	Chapter Conclusion	78
3	Keep It Simple: CNN Model Complexity Studies for Wireless Classification Tasks	80
3.1	Overview of Classification Tasks	81
3.1.1	Transmitter Categorization	82
3.1.2	Levels of Classification	83
3.2	Implementation Details of the CNN	84
3.2.1	CNN Parameters for Model Complexity	84
3.2.2	CNN Architecture for Interference Classification	84
3.3	Experimental Setup	85
3.3.1	Dataset Generated For Model Complexity	86
3.3.2	Datasets for Interference Classification	89
3.4	Results	90
3.4.1	Model Complexity	90
3.4.2	Interference Classification	91
3.5	Chapter Conclusion	93

4	Sample Complexity Studies in Wireless Communications	94
4.1	Estimating the Required Training Dataset Size for Transmitter Classification	
	Using Deep Learning	94
4.1.1	Technical Background	95
4.1.2	Dataset Generation	98
4.1.3	Transmitter Classification Performance Prediction Scheme	102
4.1.4	Experimental Setup	107
4.1.5	Evaluation of Results	110
4.1.6	Related Works	117
4.2	Chapter Conclusion	118
5	An End-to-End CNN-AE Interference Suppression System for Wireless Communications	119
5.1	Introduction	119
5.1.1	Interference Suppression in Wireless Communications	121
5.1.2	Study Objectives	122
5.1.3	Contributions	123
5.1.4	Chapter Overview	124
5.2	Related Work	125
5.2.1	Interference suppression techniques	125

5.2.2	Autoencoder-based approaches for signal processing and interference removal	126
5.3	System Model	128
5.3.1	Description of the wireless communication system	128
5.3.2	Description of the wireless communication system	128
5.3.3	Signal of interest (SOI) and interfering signal models	129
5.3.4	Over the Air Recorded Wideband Signal Dataset	132
5.3.5	Assumptions and constraints of the system model	133
5.4	Methodology	134
5.4.1	Training data generation and dataset description	134
5.4.2	Data preprocessing and preparation for training and evaluation . . .	135
5.4.3	CNN-AE Architecture	136
5.4.4	Encoding and decoding blocks	137
5.4.5	Loss function and optimization for the autoencoder	140
5.4.6	Training and validation procedures for the autoencoder	140
5.4.7	Evaluation metrics for measuring autoencoder performance	141
5.5	Results and Discussion	143
5.5.1	Latent Space Representation Visualization Using TSNE	143
5.5.2	Comparison with Traditional Interference Suppression Techniques . .	147
5.5.3	Power Spectral Density	148

5.5.4	BER vs Input SINR & Input SNR	151
5.5.5	EVM vs Input SINR & Input SNR	156
5.5.6	SINR vs Input SINR & Input SNR	160
5.5.7	Key Insights	165
5.5.8	Practical considerations and limitations	166
5.6	Chapter Conclusion	167
6	Conclusion	169
	Bibliography	173

List of Figures

2.1	High Level Structure of the Survey Work	21
2.2	A generic framework of ML-based interference suppression with ACI and CCI	25
2.3	Interference Suppression Techniques	29
2.4	A generic CNN architecture with convolution layers, ReLU activation function, pooling layer interlaced with each other and two fully connected layers	34
2.5	A Generic Autoencoder Architecture	38
2.6	A scalogram plot of the RFI dataset from [169]. (<i>Left: The original SOI with no interference; Center Left: The SOI with multi-continuous wave interference; Center Right: The signal of interest with chirp interference; Right: The signal of interest with continuous wave interference</i>). The figure was reproduced from [169] titled: “ <i>An Efficient Radio Frequency Interference (RFI) Recognition and Characterization Using End-to-End Transfer Learning</i> ”; published as an open access article in the Journal of Applied Sciences.	41
2.7	CNN Architecture proposed by Rock <i>et al.</i> [106] for radar signal denoising (N represents the number of fast-time samples and M represents the number of slow-time samples/ramps). The figure was reproduced from [106] titled: “ <i>Complex Signal Denoising and Interference Mitigation for Automotive Radar Using Convolutional Neural Networks</i> ”; published at the 22nd IEEE International Conference on Information Fusion (FUSION).	50

2.8	A CDF comparison of the sample-wise F1-Score between two CNN models, zeroing, IMAT and Ramp filtering interference mitigation techniques (<i>where x is a real-valued number</i>). The figure was reproduced from [107] titled: “ <i>Resource-Efficient Deep Neural Networks for Automotive Radar Interference Mitigation</i> ”; published in IEEE Journal of Selected Topics in Signal Processing, vol. 15.	53
2.9	A plot showing the 2D FFT result of an RNN-based Interference Mitigation Approach <i>Left: With radar interference; Right: After radar interference has been removed with RNN</i> . The figure was reproduced from [121] titled: “ <i>Automotive Radar Signal Interference Mitigation Using RNN with Self Attention</i> ”; published in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).	55
2.10	Comparison of Autoencoder-based Interference Mitigation technique with Zeroing and Auto-regressive models. The figure was reproduced from [51] titled: “ <i>A DNN Autoencoder for Automotive Radar Interference Mitigation</i> ”; published in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).	58
2.11	Comparison of DNN-based Self Interference Suppression Techniques with non-DNN based techniques. The figure was reproduced from [114] titled: “ <i>On Deep Learning Assisted Self-Interference Estimation in a Full-Duplex Relay Link</i> ”; published in IEEE Wireless Communications Letters.	60
3.1	Overview of Interference Classification	81
3.2	Confusion Matrix for CNN classification of the RFI and CRAWDAD Datasets.	90

3.3	Comparison of Test Accuracy for Homogeneous and Heterogeneous Transmitter Classes	92
4.1	Equipment Setup for the Dataset Generation. (<i>Transmitter: USRP 2921; Receiver: USRP 2921</i>)	99
4.2	Overview of the Transmitter Classification Performance Prediction Scheme .	103
4.3	Comparison of Classification Accuracy Prediction with their corresponding experimental accuracy for Experiment 1 across all Three Transmitter Categories A, B & C	111
4.4	Classification Accuracy Prediction for Experiment 2	114
5.1	Overview of the System Model	128
5.2	Power Spectral Density (PSD) of the Pulse Shaped QPSK signal with different interference types.	131
5.3	Real and Imaginary Plots of <i>CommSignal2</i> Digital Communication Signal . .	133
5.4	CNN-AE architecture consisting of the time domain input signal, the down-sampling encoder (with successive increasing depth of Conv1D, LeakyReLU activation function), latent space, upsampling decoder (with successive decreasing depth of Conv1D, LeakyReLU activation)	138
5.5	TSNE Latent Space Representation of the QPSK Signal with the interference types.	145
5.6	TSNE Latent Space Representation of the FFT-transformed QPSK Signal with the interference types.	146

5.7	PSD Comparison of the Interference Suppression Techniques	150
5.8	BER vs Input SINR Comparison of the Interference Suppression Techniques	152
5.9	BER vs Input SNR Comparison of the Interference Suppression Techniques .	155
5.10	EVM vs Input SINR Comparison of the Interference Suppression Techniques	157
5.11	EVM vs Input SNR Comparison of the Interference Suppression Techniques	159
5.12	SINR vs Input SINR Comparison of the Interference Suppression Techniques	161
5.13	SINR vs SNR Comparison of the Interference Suppression Techniques	164

List of Tables

2.1	Wireless Technologies sharing the 2.4GHz band with their interference avoidance methods	24
2.2	Comparison Between Interference Datasets	43
2.3	Dataset Analysis and Performance Comparison of Selected DL-Based Interference Suppression Techniques	65
2.4	Model Complexity Parameters from Selected DL-based Interference Suppression Techniques	68
2.5	Benefits and Drawbacks of DL-Based Interference Suppression Approaches .	75
3.1	CNN Parameters for Model Complexity	85
3.2	Comparison of Training Accuracy and Testing Accuracy for all Model Complexity and Device Category	86
3.3	CNN Parameters for Interference Classification	89
4.1	CNN Parameters	102
4.2	Transmitter Classification Method	109
4.3	Goodness-of-fit Test for Experiment 1	114
5.1	Parameters of QPSK Generation	135
5.2	Parameters of CNN-AE	136

List of Abbreviations

ACI Adjacent Channel Interference

AI Artificial Intelligence

AUC Area Under the Receiver Operating Characteristics Curve

AWGN Additive White Gaussian Noise

BER Bit Error Rate

BLE Bluetooth Low Energy

BPSK Binary Phase Shift Keying

BSS Blind Source Separation

CCI Co-channel Interference

CDMA Code Division Multiple Access

CNN Convolutional Neural Network Autoencoder

CNN Convolutional Neural Network

Conv Convolution

CS Chirp Sequence

CSMA/CA Carrier Sense Multiple Access/ Collision Avoidance

CTI Cross Technology Interference

CVCNN Complex Valued Convolutional Neural Network

CWI Continuous Wave Interference

dB Decibels

DBP Digital Backpropagation

DFT Discrete Fourier Transform

DL Deep Learning

DNN Deep Neural Network

DSSS Direct Sequence Spread Spectrum

EBPNN Extended Backpropagation Neural Network

EVM Error Vector Magnitude

FCN Fully Convolutional Network

FFT Fast Fourier Transform

FMCW Frequency Modulated Continuous Wave

GAN Generative Adversarial Networks

GEMS Generalized Estimation of Multi-path Signals

GPS global Positioning System

GPU Graphical Processing Unit

GRU Gated Recurrent Unit

IB In-band

IMAT Iterative Method with Adaptive Thresholding

IoT Internet of Things

IQ In-phase Quadrature

LDBP Learned Digital Backpropagation

LQI Link Quality Indicator

LR Linear Regression

LTE-U Long Term Evolution Unlicensed

MAE Mean Absolute Error

MCWI Multi-tone Continuous Wave Interference

MIMO Multiple Input Multiple Output

MLP Multi-layer Perceptron

MMSE Minimum Mean Square Error

MSE Mean Square Error

MSPE Mean Square Prediction Error

NoMA Non-orthogonal Multiple Access

NS3 Network Simulator 3

OFDM Orthogonal Frequency Division Multiplexing

OOB Out-of-band

OOD Out of Distribution

OSR Open Set Recognition

OTA Over The Air

PCA Principal Component Analysis

PDP-WPT Power Distributing Predominance Wavelet Packet Transform

PHY Physical

PPMSE Peak Phase Mean Square Error

PRR Packet Receive Ratio

PSD Power Spectral Density

QPSK Quadrature Phase Shift Keying

ReLU Rectified Linear Unit

RFI Radio Frequency Interference

RNN Recurrent Neural Network

RSSI Received Signal Strength Indicator

SCWI Single-tone Continuous Wave Interference

SDR Software Defined Radio

SEFDM Spectrally Efficient Frequency Division Multiplexing

SER Symbol Error Rate

SINR Signal to Noise plus Interference Ratio

SIR Signal to Interference Ratio

SNR Signal to Noise Ratio

SOI Signal of Interest

SSFM Split-Step Fourier Method

SSIM Structural Similarity

STI Same Technology Interference

t-SNE t-distributed Stochastic Neighbor Embedding

TDMA Time Division Multiple Access

UKF-RNN Unscented Kalman Filter-based Recurrent Neural Network

UMAP Uniform Manifold Approximation and Projection

V2X Vehicle to Everything

VAE Variational Autoencoders

WBAN Wireless Body Area Network

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

Chapter 1

Introduction

The wireless spectrum is limited and the demand for its use is increasing due to technological advancements in wireless communication, resulting in persistent interference issues. Despite progress in addressing interference, it remains a challenge for effective spectrum usage, particularly in the use of license-free and managed shared bands and other opportunistic spectrum access solutions. With the emergence of IoT and 5G technology, we are observing growth in the number of devices that are causing interference to one another whether intentionally or unintentionally. The need to classify interfering sources is pertinent to ensuring quality communication and adherence to spectrum sharing policies as well as also determining interference suppression approaches. Wireless interference has long been studied in the wireless communication field [1, 2, 3]. Wireless interference occurs when a signal of interest is disrupted by another signal as they share the same radio band. Wireless interference tends to degrade signal-to-noise-plus-interference ratio (SINR) performance and significantly disrupt communication between a transmitter and a receiver.

Cognitive radio (CR) technology has been identified as a tool necessary to ensure that both transmitters and receivers make optimal spectrum decisions so as to attenuate the impact of wireless interference. The most popular techniques involve extracting and classifying such interference classes so as to improve the probability of interference-free communication. The extraction and classification process typically involve a human-in-the-loop during run time and domain expert designed automated processes created for specific signal classes, thereby

making the process unscalable when the size of the network increases.

As a result of the unscalability of the manual extraction and classification, model-free approaches that are capable of automating the process have become more attractive to researchers. Convolutional neural network (CNN), a model-free deep learning approach, has demonstrated excellent performance in other domains (e.g. image classification, natural language processing, amongst others). In the wireless domain, CNN-based classifiers have been used for several classification tasks including protocol/interference classification [3, 4, 5, 6], transmitter classification [7], and modulation classification [8], amongst others. At the protocol level, a CNN model has been used to classify protocols or interference sources [4]. For instance authors in [5] proposed a wireless interference classification that utilized a CNN algorithm to classify multiple IEEE 802.15.1, IEEE 802.11 b/g and IEEE 802.15.4 interfering signals in the presence of the signal of interest. At the transmitter level, a CNN model can be used to classify different transmitter (heterogeneous) categories. The heterogeneous transmitter classification is different from the protocol classification because it considers different transmitter categories that may be emitting the same protocol, unlike the protocol or interference classification that does not account for the transmitter category. In addition, a CNN model can also be used to classify specific emitters (homogeneous) within a category (i.e. transmitters of the same model or manufacturer).

Interference classification is a crucial first step towards achieving interference-free or interference-tolerant communications because it helps to identify the source of the interference and enables the development of appropriate mitigation techniques. Once the type of interference is identified through classification, appropriate mitigation techniques can be applied to either avoid or suppress the interference. Avoidance techniques involve adjusting the communication system's parameters, such as frequency band or power level, to operate in a frequency band or at a power level that is less affected by the interference. This can be achieved by us-

ing techniques such as frequency hopping or power control. On the other hand, suppression techniques involve removing the interference from the received signal. This can be achieved by using techniques such as filtering, equalization, interference cancellation or deep learning. The selection of appropriate suppression techniques depends on the type of interference and the characteristics of the communication system. This dissertation explores deep learning based interference suppression techniques for multiple application areas and various interference characteristics. Our investigation reveal that not only do deep learning techniques reduce computational complexity in comparison to non-deep learning suppression techniques, they also outperform non-deep learning techniques in various channel environments as well as different wireless communications applications.

There are several challenges associated with model-free or deep learning approaches for interference suppression in wireless communication systems. Some of these challenges include: complexity, lack of interpretability, training data requirements, overfitting, robustness, amongst others. Complexity can be further broken down into sample and model complexity. Sample complexity of a machine learning algorithm is defined as the number of training samples that are required to achieve a performance target with high probability. Model complexity on the other hand investigates computational intensity which can make deep learning algorithms challenging to implement on resource-constrained devices or in real-time systems. In this dissertation, we address two of these challenges associated viz: (1) sample complexity, and (2) model complexity. It is important to note that some of these challenges have been addressed in some other fields that utilize deep learning (e.g. medical bioinformatics, image processing, natural language processing, amongst others) but have not been investigated in transmitter or interference classification applications.

By addressing these challenges and improving the performance and efficiency of deep learning models, researchers in the communications industry can benefit in several ways. Specifically,

the results from this work serve the wireless communications community by improving the performance of communication systems by reducing and improving signal quality leading to improved coverage, better overall reliability of communications systems and faster data rates. In addition, by reducing interference and improving the performance of communication systems, deep learning models can help to reduce the cost of deploying and maintaining communication networks. This can be especially important in applications where cost is a key factor, such as in rural or remote areas. Furthermore, the development of new techniques and algorithms for interference mitigation using deep learning can drive innovation in the communications industry. This can lead to the development of new technologies and products that can help to meet the evolving needs of users and businesses. Finally, deep learning can help to drive innovation and create new opportunities for growth and development in the industry.

The contributions of this dissertation are as follows:

- Comprehensive survey of the current state and open challenges of deep learning-based interference suppression. To the best of our knowledge, this is the first effort that reviews deep learning based interference suppression techniques for multiple application areas and various interference characteristics.
- Investigation into the relationship between deep learning model complexity and classification difficulty for a set of interference and transmitter classification tasks. This analysis is important for resource-constrained applications like the IoT that is marked by its limited processing power and storage capabilities such that it can successfully utilize lower complexity models in order to perform well.
- Quantification of the relationship between CNN model performance and sample size in wireless communication. We developed a scheme for the prediction of classification

performance and proposed rules-of-thumb for transmitter classification using CNNs based on the classification difficulty.

- Study of a convolutional neural network-based autoencoder (CNN-AE) interference suppression system for wireless communications. In addition, this study also explores the latent space behavior, CNN-AE generalizability, training methodology, and robust evaluation methods. This end-to-end approach eliminates the need for manual interference detection and suppression, simplifying the communication pipeline.

In Chapter 2, we review a wide range of techniques that have used deep learning to suppress interference by learning interference characteristics directly from data, rather than relying on expert systems. A thorough technical discussion of the prominent deep learning algorithms that have been proposed in the literature and a comparison and guidelines regarding their successful implementation in this application are provided. In Chapter 3, we propose a CNN architecture that is able to classify time-varying interference so as to enhance the physical layer's interference resilience, and perform physical layer adaptation. Furthermore, we investigate model complexity of three CNN architectures for classification of two categories of devices: homogeneous devices and heterogeneous devices and we observe how the different complexities of the CNN model impacts the test accuracy. In Chapter 4, an investigation of the relationship between model performance and sample size (also referred to as sample complexity). In the sample complexity study, we generate a dataset for training and testing a CNN for transmitter classification, we developed a scheme for the prediction of classification performance and finally, we propose a rule-of-thumb for transmitter classification using CNN. Finally, in Chapter 5, we investigate an end-to-end convolutional neural network (CNN)-based autoencoder for interference suppression in wireless communication. This is very crucial in ensuring coexistence of different wireless communications technologies in both licensed and unlicensed bands. Coexistence issues in these bands are crucial since

they can result in interference which significantly degrade the quality of communication on these bands. The uniqueness of utilizing deep autoencoders for interference suppression is that it can be done in a scalable manner with little to no human in the loop.

1.1 Interference Suppression Using Deep Learning: Current Approaches and Open Challenges

Before interference can be effectively suppressed, it is important to understand the nature of the interference and classify it into different types. There are several reasons why interference classification should be done before interference suppression. These reasons include: identifying the type of interference, avoiding over-suppression (this can lead to reduced signal quality and potentially a loss of information), and saving resources via minimizing the resources required for suppression when interference is accurately classified. By accurately classifying the interference, it is possible to select the appropriate suppression technique, avoid over-suppression, minimize the impact on other signals, and save resources. All of these factors ultimately contribute to the effectiveness and efficiency of the wireless communication system.

Traditional interference suppression techniques, on the other hand, rely on the interference categories to know which approach to follow. These techniques include physical (PHY) layer schemes, blind source separation schemes (BSS) amongst others. PHY layer schemes rely on suppressing interference via optimizing PHY layer design or providing redundancy to account for interference at the receiver [9, 10, 11, 12, 13]. They investigate optimal PHY layer schemes with baseline methods such as heuristic models, optimization techniques, greedy algorithms, amongst others, to suppress interference. For instance, Carrick *et al.* [12] investigated interference suppression methods that exploit spectral redundancies in orthogonal frequency

division multiplexing (OFDM) signals using time-varying frequency shift filter referred to as *TV-FRESH* filters to suppress linear frequency modulated radar interference. Other works in which an adaptive filter was used to remove interference from a signal without affecting crucial information include [14, 15]. One key attribute of adaptive filters is their ability to adjust the filtering criteria based on the type of interference to be suppressed [14, 16]. Earlier works that utilized these techniques rely on the use of signal properties such as cyclostationary features [17] inherent in the signal, signal envelope [16, 18, 19] and channel bandwidth [20] to identify and suppress interfering signals. As noted by [21], the major issue with PHY layer techniques that optimize the weights of the filter using estimated time-frequency statistics is that time-varying interference result in time-varying weights which tend to reduce the effectiveness of conventional adaptive filters such as Least Mean Squares (LMS). Furthermore, the reliance on domain expertise or specific signal knowledge may result in lack of scalability when considering the variety and the number of wireless devices present in modern wireless environments.

In this work, we seek to address the knowledge gap in the literature with respect to the state-of-the-art in deep learning-based interference suppression. To account for the limitations of deep learning-based interference suppression techniques, we discuss several challenges, including lack of interpretability, the stochastic nature of the wireless channel, issues with open set recognition (OSR) and challenges with implementation in Section 2.8. Our investigation shows that not only do deep learning techniques reduce computational complexity in comparison to non-deep learning suppression techniques, they also outperform non-deep learning techniques in various channel environments as well as different wireless communications applications. We believe that this is a very important study since deep learning is gaining attention in wireless communication research for interference suppression. Some of the key take-ways from this study include an understanding of key characteristics and taxon-

omy of interference-related issues in wireless communications, a systematic review of a wide range of deep learning-based interference suppression as well as the benefits and drawbacks of using deep learning algorithms for interference suppression.

1.1.1 Related Publications

The following publication has resulted from this work.

1. Oyedare, Taiwo, Vijay K. Shah, Daniel J. Jakubisin, and Jeffrey H. Reed. “Interference suppression using deep learning: Current approaches and open challenges.” IEEE Access (2022).

1.2 Keep It Simple: CNN Model Complexity Studies for Interference Classification Tasks

The proliferation of internet of things (IoT), heterogeneous cells, 5G devices, spectrum sharing and other wireless technologies has led to an increase in the number of wireless devices that interfere with each other, either intentionally or unintentionally. Classifying interference is essential for ensuring good communication quality and adhering to spectrum sharing policies. The concept of wireless interference, where a signal of interest is disrupted by another signal sharing the same channel, has been studied extensively in the field of wireless communication [1, 2, 3]. This type of interference can significantly degrade the signal-to-noise-plus-interference ratio (SINR) and disrupt communication between a transmitter and receiver.

Traditionally, many interference classification techniques have relied on rule-based approaches that are not effective when multiple coexisting technologies are in use. As a result, alter-

native classification algorithms that use feature detection or extraction techniques, such as cyclostationary feature detection [17], have been explored. However, these techniques require domain expertise and can result in a complicated solution that is oftentimes not scalable. In recent years, researchers have turned to deep learning techniques to reduce the need for domain expertise [3, 4, 22]. Interference suppression applications have also widely used deep learning [23].

Convolutional neural networks (CNNs), a model-free deep learning approach, have been shown to be effective in various domains such as image classification and natural language processing. CNNs have been used for various classification tasks, including protocol/interference classification [3, 4, 5, 6, 22, 24], transmitter classification [7], and modulation classification [8]. In using CNNs or other deep learning models for classification tasks in wireless communication applications, researchers need to ensure that they have access to high quality datasets and efficient models. While earlier works [4, 5, 22] have focused on improving the classification performance of different deep learning models, the relationship between the size of a dataset, the complexity of the CNN models used, and the difficulty of classification are not analyzed in-depth in prior works. Even though many researchers would tune hyper-parameters of their CNN during the training process, an insight into other factors (such as filter size, number of nodes in the hidden layer, etc) that affect model complexity has not been typically investigated. In resource-constrained applications like the IoT, CNN models that are too complex may not be feasible for classification in real-world situations. This is because IoT is marked by its limited processing power and storage capabilities, which can lead to challenges in terms of performance, security, privacy, and reliability [25, 26]. In order to apply deep learning to IoT applications, it is essential to understand the processes that lead to the selection of hyper-parameters in relation to model complexity, the size of the dataset and the difficulty of classification.

A typical CNN architecture consists of a series of feed-forward layers that apply convolutional filters and pooling operations, followed by fully-connected layers that convert the 2D feature maps produced by the earlier layers into 1D vectors for classification [27]. While CNNs do not require a separate feature extraction step before being used, they can be time-consuming and difficult to train from scratch because they require a large labeled dataset for building and training the model [28]. The complexity of deep learning models can be influenced by various factors, such as the number of layers, number of filters, size of the filters, and number of nodes in the hidden layer. Researchers in the field of deep learning often aim to improve the performance of their models by hyper-parameter tuning and other optimization techniques. In the literature, there has been a focus on improving classification performance through these methods. Although it is important to optimize deep learning models for performance, there has been limited attention in the wireless communication literature on thoroughly analyzing the factors that influence model complexity. This section aims to fill this gap by studying the relationship between model complexity, dataset size, and classification difficulty in a thorough and empirical manner. To the best of our knowledge, this is one of the first studies to examine this relationship.

1.2.1 Related Publications

The following publication has resulted from this work.

- T. Oyedare, V. K. Shah, D. J. Jakubisin and J. H. Reed, “Keep It Simple: CNN Model Complexity Studies for Interference Classification Tasks,” IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hoboken, NJ, USA, 2023, pp. 1-6, doi: 10.1109/INFOCOMWKSHPS57453.2023.10226045.

1.3 Sample Complexity Studies in Wireless Communications

In this section we discuss our work on sample complexity investigation in classification tasks.

1.3.1 Estimating the Required Training Dataset Size for Transmitter Classification Using Deep Learning

Transmitter classification [29] [30] is the process of accurately classifying wireless transmitters based on hardware attributes or features in the transmitted signal. Transmitter classification is also referred to as radio transmitter classification. It is a core technique used in spectrum management, IoT security, etc, to perform difficult transmitter recognition tasks. Before the popularity of deep learning models, wireless transmitter classification algorithms leveraged carefully crafted features that require domain knowledge and precision to classify transmitters. Spectrum management is the term that describes the efficient management of the radio spectrum to maximize spectrum use while minimizing RF interference. It encapsulates spectrum sensing, spectrum monitoring and spectrum forensics, all of which enables cross technology spectrum sharing. Transmitter classification has found wide applications in the spectrum forensic paradigm as it assists in rapidly classifying transmitter(s) that inadvertently misuse spectrum resources. With the abundance of devices introduced into the wireless spectrum, the challenge of classifying these devices become burdensome.

Prior works have shown that the transmitter classification problem can be effectively addressed using convolutional neural networks (CNN). CNN is effective because it is able to extract features from raw data via several hidden layers in a network, thereby making it possible to learn complex non-linear associations between the input data and output classi-

fication labels. CNNs require a huge amount of data and powerful computational resources [31]. Authors in [32] used a CNN algorithm to fingerprint IEEE 802.15.4 devices. Youssef et al. [33] utilized CNN and other machine learning algorithms to identify RF transmitters based on their intrinsic physical characteristics.

While several prior studies have explored the utilization of CNNs for transmitter classification or identification (e.g., [32, 33, 34, 35]), none of them provided any formal discussions on the impact of the training dataset size on classification performance. Like all deep learning techniques, the performance of CNNs is directly impacted by the size of the training dataset. Through the findings presented in this work, we shed light on this understudied problem. Specifically, in this work, we focus on quantitatively analyzing the relationship between the size of the training dataset and classification accuracy.

In computational biology and related fields, researchers have identified ways of estimating the training dataset size needed for acceptable classification performance [36][37][38]. In the wireless communication domain, such an estimation is more challenging because the input to the deep learning algorithm (IQ data and other representation of the transmitted signals) has complex underlying processes and do not have typical average values when compared to the input data used in other domains [39]. Often, the quantity of data needed depends on the complexity of the classification problem and the deep learning algorithm. The relationship between dataset size and performance have been studied in many non-deep neural networks literature [40][41] where features used for the classification are crafted by domain experts. Since the features are manually crafted, the training dataset size is small when compared with deep neural networks where the features are automatically learned by the algorithm. Also, their applications (discrete choice analysis and medical bio-informatics) which uses non-deep machine learning algorithms require smaller training dataset for classification whereas wireless transmitter data with higher complexity requires more training dataset.

In this work, we generate a dataset for training and testing a CNN for transmitter classification, we developed a scheme for the prediction of classification performance and finally, we propose a rule-of-thumb for transmitter classification using CNNs.

1.3.2 Relationship between Model Performance and Sample Size in Wireless Communications Applications of Deep Learning

In this work, we systematically study the *sample complexity* problem associated with recall for each class. This is different from the effort described in the previous section in terms of the datasets utilized, evaluation methodology and the use of model-intrinsic metrics. As far as current research indicates, this is the first investigation into the sample complexity problem in relation to model-intrinsic metrics in deep learning-based classification models and the robust statistical model estimation of those metrics. The sample complexity issue has not received much attention from the deep learning research community, despite being extensively studied in non-deep learning models like logistic regression and linear regression [36, 37]. In [7, 42], sample complexity in deep learning was empirically investigated; however, the findings are limited to fixed testing datasets. Furthermore, when they increase the size of the training dataset, [7, 42] assume that all of the training data are sampled uniformly and independently from the entire training dataset. The impact of altering the percentage of each class's training data when additional training data is added on model performance was not discussed in those works. This work addresses sample complexity in transmitter classification and modulation recognition tasks. Transmitter classification is useful in assessing RF spectrum utilization [43] as well as classifying rogue transmitters in the spectrum sharing ecosystem. Traditional modulation recognition techniques are either too complicated [44] or not robust enough in discriminating signal modulation types. This is why deep learning based modulation recognition have found wide usage in the modulation recognition/classification research.

1.3.3 Detecting Out-of-Distribution Data in Wireless Communications Applications of Deep Learning

Samples from the same distribution as the training data are referred to as *in-distribution* (ID) data, while those from a *substantially distinct* distribution are termed *out-of-distribution* (OOD) data. Previous studies indicate that deep learning models often express high confidence even for entirely unrecognizable [45] or irrelevant inputs [46]. OOD data can make it difficult to analyze results in real-world applications. OOD data may cause unanticipated perception errors in domains crucial to safety and security, like autonomous driving, which could lead to mishaps or system failures. OOD data may not always show up as obvious mistakes in the model, so their impact might go unreported. This work proposes a deep learning-based model that utilizes variational autoencoders to map high-dimensional input data to low-dimensional feature representations in latent space, classifies input data by analyzing feature representations and data reconstruction errors to determine if it is out-of-distribution (OOD) or not. Unlike previous VAE-based models, the approach described in this work integrates class labels during training, allowing simultaneous classification and OOD detection, and its approach of embedding label information in feature representations outperforms models relying solely on reconstruction probabilities. Details of this work can be found in [47].

1.3.4 Related Publications

The following publication resulted from this work.

1. Oyedare, Taiwo, and Jung-Min Jerry Park. “Estimating the required training dataset size for transmitter classification using deep learning.” In 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), pp. 1-10. IEEE,

2019.

2. Liu, Jinshan, Taiwo Oyedare, and Jung-Min Park. “Detecting Out-of-Distribution Data in Wireless Communications Applications of Deep Learning.” *IEEE Transactions on Wireless Communications* 21, no. 4 (2021): 2476-2487.
3. Liu, Jinshan, Taiwo Oyedare, and Jung-Min Park. “Relationship between Model Performance and Sample Size in Wireless Communications Applications of Deep Learning”. Under review at the *IEEE Transaction on Mobile Computing (TMC)*.

1.4 An End-to-End CNN-AE Interference Suppression System for Wireless Communications

We now turn our attention to the task of interference suppression. Interference suppression in wireless systems is crucial for reliable communication amid rising congestion. Varied sources like neighboring devices, environmental factors, and channel constraints degrade signal quality, leading to failures, reduced data rates, and spectral inefficiency, hindering seamless wireless communication [48]. Suppressing interference in wireless systems ensures robust reception, higher throughput, expanded coverage, reduced latency, and enhanced network performance. This enables vital applications like 5G, IoT, autonomous vehicles, and critical services, ensuring reliability and efficiency in wireless communication [49].

In this work, we investigate how autoencoders can be used to suppress interference at the decoder by learning the latent space representation of the input data with interference. Compared to traditional suppression methodologies and also deep learning approaches in the literature, this approach offers a comprehensive and end-to-end solution for interference suppression, where the entire process, from encoding the data to its reconstruction, is seamlessly integrated into a single neural network framework. This study provides several insights.

It eliminates the need for manual interference estimation or preprocessing steps, streamlining the signal processing pipeline. Additionally, it allows the autoencoder to adapt and generalize to varying interference patterns, enhancing its robustness in real-world scenarios where interference characteristics may change unpredictably. Further, it leverages the latent space representation to discern and suppress the interfering tones present within the signal of interest. Through its encoding and decoding processes, the autoencoder effectively disentangles the signal components, enabling the removal of the interference from the original signal. The system efficiently reconstructs the cleaned signal from the latent space representation, thus demonstrating its ability to substantially attenuate or eliminate the interference.

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimensionality reduction technique used for visualizing high-dimensional data in lower-dimensional space [50]. t-SNE is especially good at capturing complex, nonlinear relationships in data, which makes it useful for revealing patterns and structures that might be obscured in higher dimensions. The fundamental idea behind t-SNE is to model the similarity between data points in the high-dimensional space and the low-dimensional space. It provides a visual representation of the algorithm's performance in differentiating between the signal categories. Observing distinct clusters enables a clearer and more accessible visual understanding of the interference effects on the signal of interest (SOI). In addition, our approach eliminates the necessity for explicit interference detection mechanisms, enabling the system to autonomously discern and mitigate the undesired signal disturbances.

The autoencoder architecture proposed in this study utilizes 1D convolutional layers for both the encoding and decoding blocks. The encoder consists of several Conv1D layers with LeakyReLU activation and batch normalization. The convolutional layers are fundamental to CNNs and play a key role in detecting spatial patterns within the input data. These layers use learnable filters to convolve over the input, capturing local features and enabling

the network to recognize hierarchical patterns. In the training process of the autoencoder, the model is trained using a dedicated training dataset, while simultaneously evaluating the model's performance on a separate validation set. The CNN autoencoder was trained utilizing a dataset comprising original SOI with interference, with the target signal set as clean signal (signal free of interference). This training paradigm allowed the autoencoder to learn and discern interference patterns for subsequent suppression, aiming to reconstruct the SOI.

While autoencoders have been explored in the context of interference suppression for automotive radar [51] and in jointly optimizing transmitter and receiver systems to combat interference [52, 53], we found a lack of prior work evaluating autoencoders for receive-side interference suppression for digital communication signals. Also, authors in [54] explored the use of denoising autoencoders for specific interference sources and structures within the Wi-Fi spectrum. Compared to the prior art, this study leverages rigorous empirical analysis to understand how autoencoders can enhance signal quality by suppressing interference, and ultimately improve the reliability and efficiency of wireless communication.

Our results show impressive suppression performance for in-band, out-of-band and a publicly available wideband digital signal dataset in terms of quantitative metrics bit error rate (BER), error vector magnitude (EVM) and signal-to-interference -plus-noise-ratio (SINR). In addition, we also compare our CNN-AE model with baseline models such as matched filtering and adaptive filtering (specifically, the least squares filtering). Furthermore, we also show the performance of CNN-AE as both interference and noise is varied. Our results show that the CNN-AE outperformed both baseline models for all metrics. In addition, this study also explores the latent space behavior, CNN-AE generalizability, training methodology, and robust evaluation methods. Utilizing t-SNE plots, we showcase the CNN-AE's effectiveness in capturing and differentiating interference-induced variations from the desired clean signal

within the latent space. The study’s key insights stem from eliminating the need for manual interference estimation, thereby enhancing the signal processing pipeline. Additionally, it improves adaptability to various interference patterns, strengthening its robustness in dynamic real-world scenarios. Through decreased dependency on explicit interference detection mechanisms, the model autonomously identifies and mitigates undesired signal disruptions. This autonomous interference suppression represents a significant stride forward, showcasing its potential to streamline interference management in wireless communication systems.

1.4.1 Related Publications

The following publication has resulted from this work.

- T. Oyedare, B. Malolan, D. J. Jakubisin and J. H. Reed, “An End-to-End CNN-AE Interference Suppression System for Wireless Communications,” to be submitted to IEEE Transactions on Cognitive Communications and Networking (TCCN).

Chapter 2

Interference Classification and Suppression Using Deep Learning: Current Approaches and Open Challenges

2.1 Introduction

In this work, we seek to address the knowledge gap in the literature with respect to the state-of-the-art in deep learning-based interference classification and suppression. To account for the limitations of deep learning-based interference suppression techniques, we discuss several challenges, including lack of interpretability, the stochastic nature of the wireless channel, issues with open set recognition (OSR) and challenges with implementation in Section 2.8. To the best of our knowledge, this is the first work that explores deep learning based interference suppression techniques for multiple application areas and various interference characteristics. Our investigation shows that not only do deep learning techniques reduce computational complexity in comparison to non-deep learning suppression techniques, they also outperform non-deep learning techniques in various channel environments as well as different wireless communications applications. We believe that this is a very important study since deep

learning is gaining attention in wireless communication research for interference suppression. This work would best serve researchers working on interference suppression issues or the application of deep learning to wireless communications, as well as, wireless spectrum users and decision makers. The work described in this section has been published by the author of this dissertation and can be found in [23]. Our contributions are as follows:

- We identify key characteristics and provide a taxonomy of interference-related issues in wireless communication. This includes key features of adjacent channel interference, co-channel interference, homogeneous and heterogeneous technology interference prevalent in wireless communication applications.
- We provide a review of interference classification techniques that utilized supervised and semi-supervised learning techniques. This study of interference classification is necessary since it precedes interference suppression.
- We review a wide range of techniques that have used deep learning to suppress interference. Techniques include both supervised deep learning (such as convolutional neural networks (CNN), deep neural networks (DNN) and recurrent neural networks (RNN)) and unsupervised techniques such as deep autoencoders. We also discuss the different wireless communication applications for which the deep learning techniques have been utilized.
- We summarize the benefits and drawbacks of using different DL algorithms for interference suppression. This summary also includes the technical approach utilized as well as the benefits and drawbacks associated with them. Table 2.5 has been provided to give a high level summary of the benefits and drawbacks of DL-based interference suppression approaches.

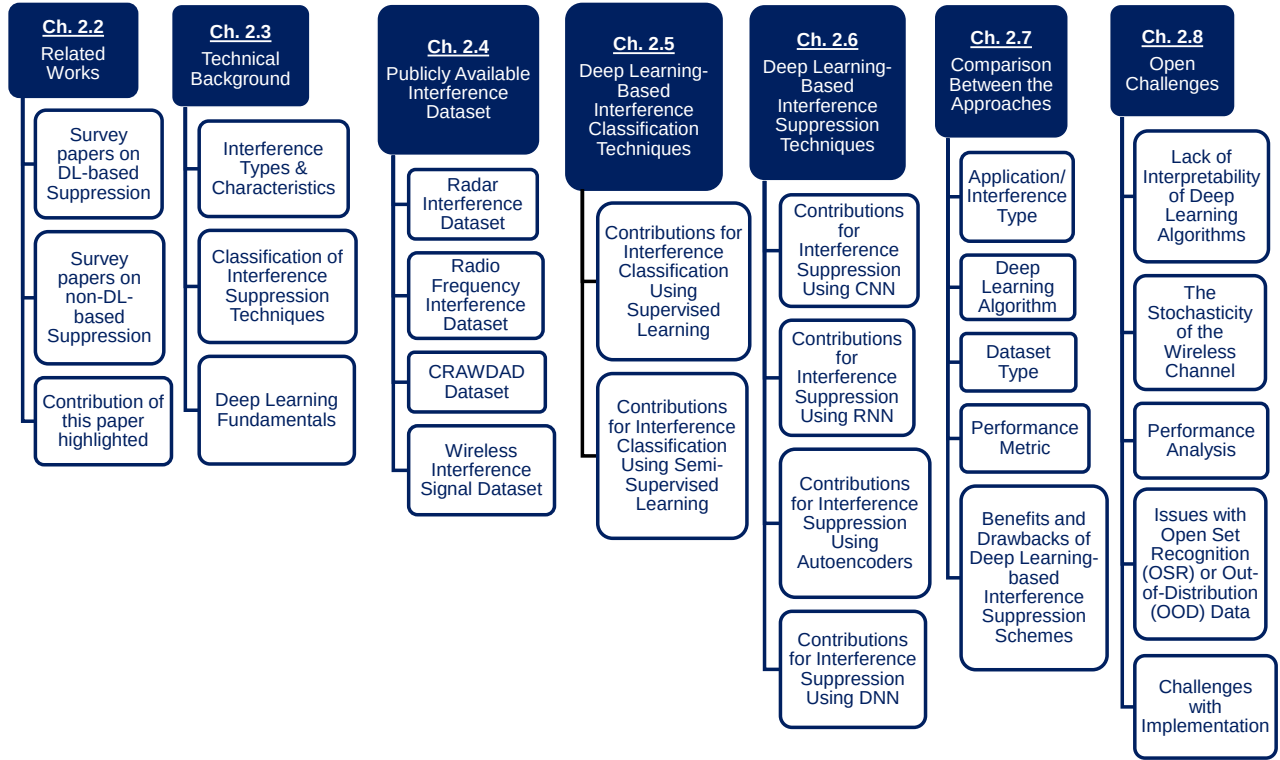


Figure 2.1: High Level Structure of the Survey Work

- We highlight challenges associated with using deep learning for interference suppression.

While some of these challenges are not unique to DL-based interference suppression, we note that they have the capability of slowing down the adoption of DL in interference suppression study.

Fig. 2.1 shows a high level structure of the survey including a breakdown of the topics discussed in each section.

2.2 Related Works

To the best of our knowledge, the current literature lacks a comprehensive study on deep learning-based interference suppression. As at the time of writing this dissertation, we have found only two surveys which consider DL techniques in the context of interference suppres-

sion. In [55], the authors discussed deep learning techniques for radar applications in general and briefly highlighted interference techniques that fell into the deep learning categories. One key distinction of our work from theirs is that the interference types they considered includes only jamming and clutter. We note that this investigation is very important and our work goes a step further by investigating more DL-based interference suppression applications including radars, optical communications, IoT networks, GPS navigation, full duplex systems and wireless sensor networks. Further, we consider many other types of interference such as self-interference, adjacent channel interference, homogeneous and heterogeneous technology interference. Furthermore, the jamming suppression techniques considered in [55] focused on target signal classification and recognition. Our work is fundamentally different from theirs because we focused more on interference suppression techniques and not interference classification techniques. In another work, Chen *et al.* [56] surveyed both coexistence and interference mitigation techniques, including traditional (non-DL) and DL approaches for wireless personal area network (WPAN) and WLAN applications. While this is also another important contribution, the scope of the application considered is different in comparison to the one presented in this survey. Also, we found that their work focused more heavily on non-DL interference suppression and briefly discussed DL-based suppression techniques. Furthermore, in our own work, we discuss several critical interference characteristics and why DL techniques are suitable to suppress them. Also, we point the community to publicly available dataset for interference suppression.

Many of the other surveys [57, 58, 59, 60, 61] on interference suppression reviewed works that utilized signal processing techniques and even non-ML techniques to avoid interference. For instance, Azmi *et. al.* [57] discussed works that used techniques such as transmit power control, dynamic frequency selection, interferer classification schemes [58], time slotted channel hopping schemes amongst others to suppress interference in wireless sensor networks

(WSN). Further, Lu *et al.* [60] reviewed interference mitigation schemes that utilized adaptive array processing in GPS systems. Le *et al.* [61] also surveyed interference mitigation schemes used in wireless body area sensor networks. Kunert *et al.* [59] surveyed six basic techniques that researchers have used to suppress radar interference. They sorted the suppression techniques into domains such as polarization, time, frequency, coding and space. While the techniques discussed in their survey typically require domain expertise to be successful, it is not clear if they will be scalable in applications such as IoT and 5G where there are multiple devices sharing the same frequency channel. Furthermore, one of the approaches discussed by Kunert *et al.* [59] is “communicate and avoid” in various domains including time and frequency. While this is a straightforward approach, it involves multiple steps to achieve and the time to implement in a latency-sensitive applications such as 5G New Radio (NR) or vehicle to everything (V2X) may be unrealistic. Some of these limitations can be tackled using deep learning techniques surveyed in this work.

For automotive radar suppression algorithms, Toth *et al.* [62] investigated state-of-the-art interference suppression techniques including zeroing, ramp filtering, iterative method with adaptive thresholding (IMAT) and time domain parametric interpolation (IVMip). Their study did not include machine learning (ML) or DL approaches that could be leveraged to suppress interference. Also, their application focused primarily on automotive radar interference, whereas our survey encompasses many other application areas. In the same vein, Nabil *et al.* [63], discussed advances in self and mutual interference suppression techniques for wireless networks. They surveyed redundancy techniques such as MIMO technology to suppress mutual interference in scenarios when the receiver does not have complete knowledge of the interfering signal. They also discussed interference cancellation techniques such as Balun transformer, isolation of the transmitter and receiver paths, phase shifting and spatial techniques (enabling an appropriate distance between the transmit antennas).

Our work is different from all the other surveys on interference suppression because we systematically explore suppression approaches that utilize deep learning for many application areas and interference characteristic types while also providing a comparison between them and the major challenges with using deep learning to suppress interference. This crucial information helps to provide key insight for many researchers who work at the intersection of deep learning and interference suppression. To the best of our knowledge, this is the first work that explores deep learning based interference suppression techniques for multiple application areas and various interference characteristics. There is a lot of promising results that show that DL based techniques can be very useful in quickly suppressing interference in many wireless communications applications.

Table 2.1: Wireless Technologies sharing the 2.4GHz band with their interference avoidance methods

Technology	Number of Channels	Interference Method	Avoidance
Wi-Fi (IEEE 802.11b/g/n/ax)	3 20MHz static channels	Collision avoidance	
Bluetooth	79 1 MHz dynamic channels	Frequency hopping	
Wireless USB (including wireless keyboard, mouse, etc)	79 1 MHz dynamic channels	Frequency agility	
IEEE 802.15.4	16 5MHz static channels	Fixed collision avoidance	
Bluetooth Low Energy (BLE)	40 2MHz channels	Frequency hopping	
Cordless Phone	100KHz (Analog Phone); 800KHz (Frequency hopping phone)	Frequency hopping	

2.3 Technical Background

2.3.1 Interference Types and Characteristics

In this section, we discuss the types and characteristics of interference in the wireless communications research.

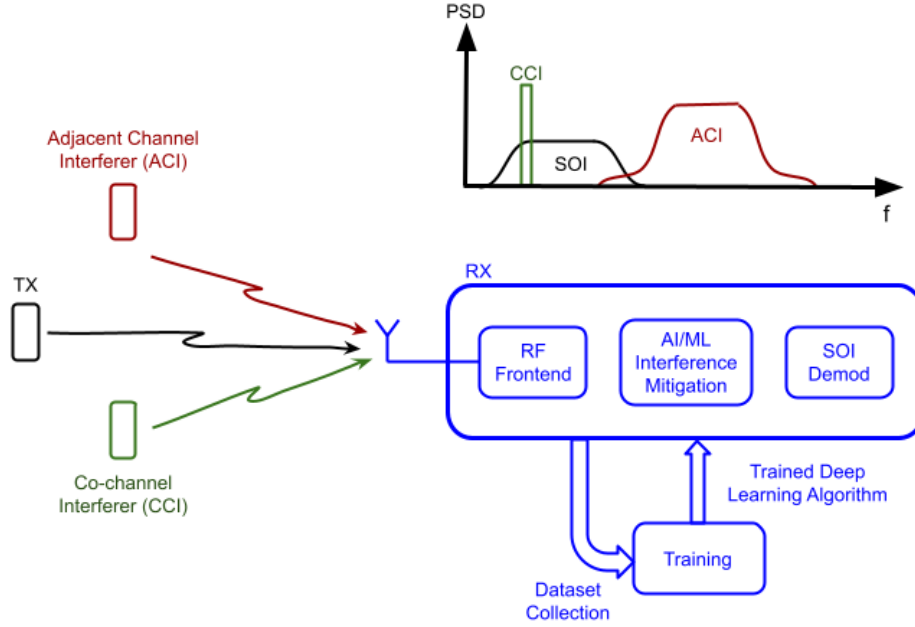


Figure 2.2: A generic framework of ML-based interference suppression with ACI and CCI

2.3.1.1 Adjacent Channel Interference

Adjacent channel interference (ACI) occurs when the power from a signal in a channel adjacent to it causes interference to another signal in a different channel. There are several reasons why ACI may occur, for example in frequency modulated (FM) systems, incomplete filtering may lead to ACI. Furthermore, an inadequate frequency control in either the interfering channel or the reference channel may also lead to ACI. In some cases, interference avoidance schemes can be used to avoid ACI. For instance, while the 2.4 GHz Wi-Fi band supports a total of 11 channels, it is recommended that Wi-Fi users in the United States utilize non-overlapping channels (take for example: channels 1, 6 and 11) [64].

2.3.1.2 Co-channel Interference

Co-channel interference (CCI), on the other hand, occurs when two devices or transmitters that share the same frequency for transmission interfere with one another [65]. In cellular communication applications, CCI is more pronounced when frequency bands are re-used.

Interference avoidance schemes alone are not sufficient to suppress CCI. In this survey, we focus on the more difficult problem of CCI.

Fig. 2.2 shows an illustration of how CCI and ACI interferes with the SOI. It is also important to note that the data collection, offline training and other deep learning components are done at the receiver. We discuss the details of some of the ML interference mitigation schemes in Section 2.6. We provide Fig. 2.2 as a generic representation of the application of machine or deep learning to interference suppression or mitigation.

2.3.1.3 Self Interference

Self interference (SI) is a very common phenomenon in full duplex systems where transmitters and receivers utilize the same frequency band for simultaneous communication. In cognitive radio (CR) radio applications, in-band full duplex transmission and reception on the same frequency at the same time is very useful in doubling spectral efficiency while maximizing the use of available spectrum resources [66, 67]. Techniques such as antenna, analog and digital cancellation have been studied in canceling interference, however, it has been shown that it is very difficult to completely cancel SI due to imperfections such as IQ imbalance, power amplifier non-linearities, amongst others [68].

2.3.1.4 Homogeneous Technology Interference

Homogeneous technology interference, also referred to as same technology interference, occurs when a wireless technology is interfered by a varying number of signals from the same technology. They can occur either in the same channel (co-channel) or in different channels (adjacent channel). In some cases, the number of interfering signals could be more than one, depending on the type of frequency band the dominant signal is utilizing. Such interference is often due to the use of multiple access technologies such as IEEE 802.15.1 and

IEEE 802.15.4 [5]. Farid *et al.* [69] also studied homogeneous technology interference avoidance technique such as CSMA/CA-based approach; interference suppression techniques such as adaptive techniques, graph coloring, TDMA based approaches, and heuristic techniques, amongst others, for wireless body area networks (WBAN)¹. Also, code division multiple-access (CDMA), a multiplexing technology for cellular communication technologies, is also vulnerable to homogeneous technology interference. In spread spectrum CDMA, the spread spectrum systems share the band with other communication systems. To do this successfully, individual users in a spread spectrum CDMA system has a unique binary psuedo-noise (PN) code that helps the receiver to distinguish it from the other users in the system. In the past three decades, several interference suppression techniques, such as linear [70] and non-linear estimation techniques [71], have been proposed to suppress homogeneous technology interference in CDMA systems. We note that the research in homogeneous technology interference has considerably waned in comparison to heterogeneous technology interference discussed in Section 4.4. As a result of this, many of the DL-based interference suppression techniques explored heterogeneous technology interference more than they did homogeneous technology interference.

2.3.1.5 Heterogeneous Technology Interference

Understanding heterogeneous technology interference, also referred to as cross technology interference (CTI), is very crucial to the coexistence of different wireless technologies sharing similar unlicensed frequency bands. CTI occurs when a wireless technology is interfered by signals from the different technology [5]. Classic examples include licensed LTE vs Wi-Fi

¹WBAN is a broad term describing wearable sensors connected in a wireless manner to a hub [69].

[72, 73], unlicensed LTE versions² vs Wi-Fi [74, 75], radar³ vs Wi-Fi [76, 77], dedicated short range communication (DSRC)⁴ vs Wi-Fi [79, 80].

Due to coexistence issues that can lead to interference between these heterogeneous technologies, several studies have proposed several interference suppression techniques [81, 82, 83, 84, 85] to better explore the manner in which these technologies interact. In most cases, CTI exhibit distinct recognizable patterns that interference-aware protocols can leverage [81]. Grunau et. al. [5] showed that the impact of CTI for an IEEE 802.11 b/g signal interfered by IEEE 802.15.1 and found that the IEEE 802.15.1 signals were relatively benign when compared to the converse. This is because both IEEE 802.15.4 and IEEE 802.15.1 signals result in lesser overlapping interference compared to IEEE 802.11 b/g signal because they have narrower bandwidth. However, for low-power wireless devices, CTI can disproportionately impact their coexistence with other devices when sharing the same band [82].

Table 2.1 shows wireless technologies such as IEEE 802.11 (the standard that defines Wi-Fi), Bluetooth, Bluetooth low energy (BLE), wireless USB, IEEE 802.15.4⁵, and others sharing the 2.4 GHz band. Despite the fact that these technologies have different power levels and width of the occupied spectrum, they could still cause interference to one another given the

²Unlicensed LTE versions include LTE-Licensed Assisted Access (LTE-LAA), LTE in Unlicensed Spectrum (LTE-U), LTE-WLAN Aggregation (LWA) and Multefire. In general, these unlicensed LTE technologies are either license-anchored or non license-anchored. For license-anchored systems (LTE-U, LAA) the primary carrier (anchor) uses the licensed parts of the spectrum and for non license-anchored systems (Multefire) both control and data traffic are sent through the unlicensed spectrum [74].

³While radars are the incumbent users of the 5.150-5.350 GHz and 5.470-5.725 GHz bands, Wi-Fi devices operating in the 5GHz band also share the spectrum when no radar system is transmitting [74]. This can lead to interference from the Wi-Fi devices to the legacy radar systems.

⁴DSRC supports applications based on vehicular communication [78]. The IEEE 802.11p standard, an amendment to the IEEE 802.11 standard, was developed to incorporate the wireless access in vehicular environments (WAVE) vehicular communication system in order to support intelligent transportation systems. This technology operates between 5.85-5.925 GHz band. Since Wi-Fi also operates as a secondary user in the same radar band, they can often times cause interference to the legacy radar technologies.

⁵IEEE 802.15.4 is the basis for Zigbee, ISA100.11a, WirelessHART, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) and cordless phone (including analog and frequency hopping phones). Other technologies include microwave oven, wireless camera, baby monitors, car alarms, video devices amongst others. It defines the low-rate wireless personal area networks (LR-WPANs)[86].

fact that they share the congested unlicensed 2.4 GHz band.

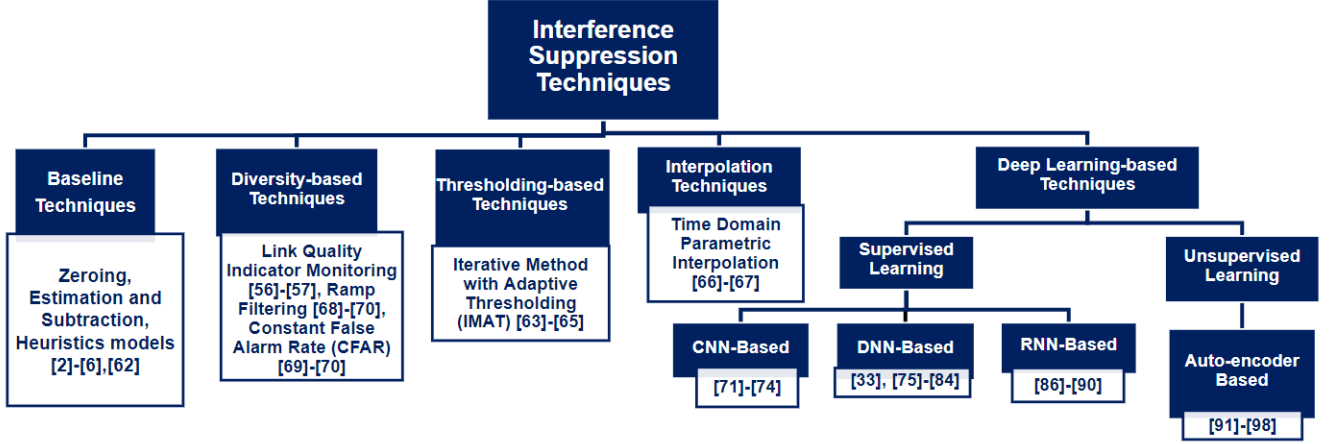


Figure 2.3: Interference Suppression Techniques

2.3.2 Classification of Interference Suppression Techniques

Broadly speaking, interference suppression can be divided into the following categories (shown in Fig. 2.3): baseline techniques, diversity-based techniques, thresholding-based techniques, interpolation techniques and deep learning-based techniques. Furthermore, researchers have investigated polynomial-based cancellation techniques to cancel SI [87, 88, 89]. Polynomial-based methods used in full duplex systems typically result in a huge number of parameters which further make their implementation complex. While this survey focuses on deep learning-based techniques, we briefly describe non-deep learning techniques in this section to provide some background information necessary to understand the comparison in Section 2.7. Fig. 2.3 shows a taxonomy that includes both non-deep learning and deep learning techniques for interference suppression.

Before the adoption of ML or deep learning methods to suppress interference, researchers used various techniques to suppress or excise them. Because many low-power wireless networks are power- and computation-constrained, PHY layer information and time diversity of

the wireless channel have been proposed to suppress heterogeneous technology interference. For example, Hithnawi [90] monitored link quality indicator (LQI) [91] values of corrupted frames together with conventional threshold mechanisms to detect the presence of interference. When interference is detected, the changes in received signal strength indicator (RSSI) is collected. The RSSI changes are captured by sampling the received signal. The RSSI readings is then sent to the detection algorithm. The corrupted packet is reconstructed from subsequent interfered re-transmission and the buffering of the correctly decoded bytes [90].

While we acknowledge that earlier works have investigated interference avoidance schemes, such as transmit power control [92], dynamic frequency switching [93, 94, 95], amongst others, our survey focuses on techniques that suppress interference. These techniques, shown in Fig. 2.3, include zeroing of interference samples [96], IMAT [97, 98, 99], time domain parametric interpolation [100, 101], ramp filtering (RFmin) [102, 103, 104] amongst other approaches. While zeroing out interference components may be the simplest method in suppressing interference, it could reduce SNR or even potentially zero out actual signal [62]. The RFmin technique exploits the diversity of range spectrum over ramps [62]. In IMAT approaches, the interfered samples are first zeroed and the corresponding zeroed time-domain gap is then iteratively interpolated by replacing the gap with the inverse Fourier transform of the actual dominant spectral component [62, 97, 98]. The time domain parametric interpolation approach is similar to the IMAT approach with the difference being the use of a parametric signal model to interpolate the gaps resulting from zeroing out interfered signals [100]. It is important to point out that the list of non-deep learning techniques for interference suppression mentioned in this section is non-exhaustive, we only highlight a few of those techniques since the main focus of this survey is deep learning techniques for interference suppression. Readers who are interested in these other techniques can refer to [59, 60, 61]. Fig. 2.3 shows popular algorithms for each category.

Deep learning techniques are generally divided into supervised and unsupervised learning⁶. For the supervised learning techniques, CNN-based suppression techniques were discussed in [105, 106, 107, 108], deep neural network (DNN) based techniques were discussed in [68, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119], and recurrent neural network (RNN) based techniques were investigated in [120, 121, 122, 123, 124]. For the unsupervised learning techniques, autoencoder-based suppression techniques were discussed in [51, 52, 53, 54, 125, 126, 127, 128].

Furthermore, ML based channel state information (CSI) techniques [129] have also been studied to estimate the required transmit power level necessary to suppress interference. This approach strongly relies on a precise estimation of transmit power needed to suppress interference in a wireless sensor network (WSN) application. Other methods, such as [91, 130] have utilized lightweight feature extraction and lightweight decision trees to detect and identify interference in Wi-Fi, Bluetooth, and microwave ovens.

Because of the plethora of devices contending to share the wireless spectrum, conventional interference suppression approaches may not be scalable to resolve associated interference issues that may result. Since deep learning approaches have been found to improve classification accuracy for many interference classification or identification schemes[3, 4, 5, 85], researchers have started taking their investigation to the next level and their potentials for interference suppression [105, 106, 120, 125]. We discuss them in detail in Section 2.6.

2.3.3 Deep Learning Fundamentals

In the last two decades, we have witnessed tremendous and unparalleled breakthrough in computing hardware technologies. This breakthrough led to affordable and pervasive com-

⁶The major distinction between supervised and unsupervised learning is the use of labeled datasets. The former relies on labeled inputs and outputs while the latter uses ML algorithms to discover the structure of the unlabeled input data

puting resources. Take, for example, graphics cards which have seen significant growth in computing power. Computers are now capable of doing complex operations in such a short amount of time. Deep learning techniques allow computers to build complex concepts out of simpler concepts [131]. Broadly speaking, there are three types of deep learning algorithms: supervised, unsupervised and reinforcement learning. Supervised learning techniques typically learn from labeled data [132, 133] while unsupervised techniques do not require labels to learn from the data [134]. Reinforcement learning [135] on the other hand deals with an intelligent agent that take actions in order to maximize its cumulative reward. Semi-supervised learning [136, 137, 138], a technique that combines supervised (labeled data) and unsupervised (unlabeled data) learning has also received a lot of attention in recent times. While earlier works such as Longi *et al.* [31] utilized a semi-supervised learning technique to detect interference sources from wireless local area networks (WLAN), we note that, to the best of our knowledge, its use in interference suppression has received little to no attention.

Deep learning techniques have enjoyed significant success in solving challenging problems in several application domains, including image classification, medical bio-informatics, natural language translation amongst others. Motivated by these developments, researchers in wireless communication have embraced deep learning to address the most difficult problems in wireless communications and networking, such as interference identification, classification [3, 5] and suppression [125], modulation recognition [139], transmitter classification [29], spectrum sensing [140], traffic classification [141] amongst other challenges. In addition, deep learning techniques can be very useful in managing coexistence issues arising from the deployment of fifth generation (5G) New Radio (NR) and sixth generation (6G) technologies with regards to scalability, generalization and the ability to learn from the data rather than reliance on domain expertise. Therefore, using deep learning to estimate, predict and suppress time-varying interference strengthens interference resilience of the PHY layer.

Earlier applications of deep learning to wireless communications have investigated DL-based end-to-end systems for multiple user, multiple antenna systems that mutually optimize the transmitter and the receiver so as to improve performance compared to traditional non-DL or non-ML approaches [53, 142]. Deep learning techniques have also been used for channel modeling and estimation. For instance, generative adversarial networks (GANs)⁷ were used by [144, 145] to approximate stochastic channel response from the raw data for real time channel modeling. Furthermore, DL-based OFDM receivers have been used to learn the channel between a transmitter and a receiver and also decode the received signal [146]. DL-based belief propagation algorithm that cascades CNN to a belief propagation network was utilized by Liang *et al.* [147] to accurately estimate noise. While these techniques are not interference suppression techniques, they may provide useful information for subsequent interference suppression processing.

The major difference between deep learning techniques and model-based techniques is that deep learning techniques are data-driven and thus differ significantly from model-based techniques that typically require domain expertise (commonly referred to as expert systems). Our investigation shows that many of the deep learning techniques outperform conventional model-based techniques as seen in the discussions in Section 2.6.

2.3.3.1 Convolutional Neural Network

CNNs are supervised deep learning techniques that learn the relationship between input data and their corresponding labels [148]. For instance, in [149], the authors proposed an approach

⁷GANs learn deep representations by training a pair of networks in competition with each other. It consists of a generator and a discriminator network in which the generator aims to trick the discriminator by generating fake samples. The discriminator on the other hand receives both the fake samples and the real samples and tries to tell them apart. Both generator and discriminator are usually implemented with neural networks [143]. Hence, one neural network is pitted against another neural network with each improving its capabilities over time.

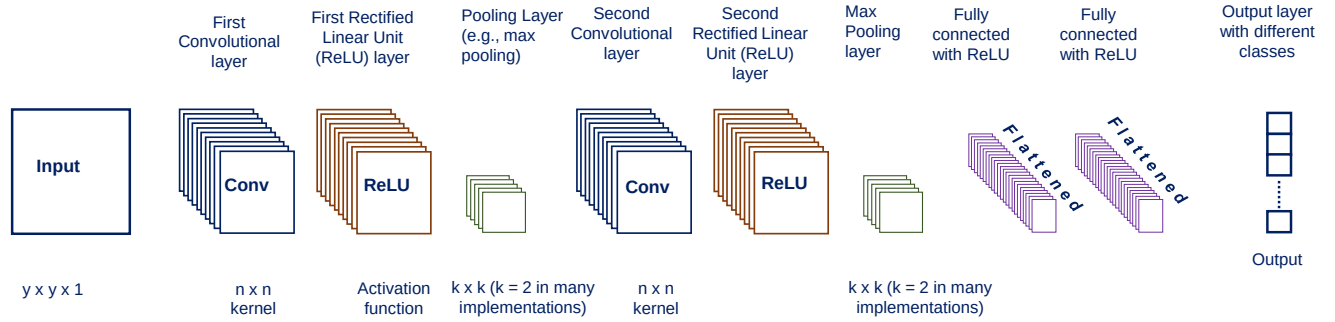


Figure 2.4: A generic CNN architecture with convolution layers, ReLU activation function, pooling layer interlaced with each other and two fully connected layers

to suppress radio frequency interference (RFI) in radio frequency data using the U-Net ⁸. A U-Net is a CNN algorithm that is used to classify clean signal and RFI signatures in 2D time-ordered data acquired from a radio telescope. While this work did focus on the use of a deep learning algorithm to classify interference, the technique used was only validated for cosmology applications that deals with satellite systems, air traffic communications, radio telescope, etc. Just like RFI suppression research, radar interference suppression is also another well-studied research area [105, 106, 120, 125, 150, 151]. Using CNNs to suppress interference is still an emerging area of research. This is because there are many factors to the successful adoption of CNNs to interference suppression. One such factor is the availability of benchmark datasets.

Fig. 2.4 shows a generic CNN architecture with convolution layers, rectified linear unit (ReLU) activation layers and pooling layers interlaced with one another before the output is then sent to two fully connected layers. This generic architecture is similar to the one described by [152]. The input data could be images as seen in the modified National Institute of Standards and Technology (MNIST) ⁹ data with size $28 \times 28 \times 1$ [154]. The input image is then passed through the convolution layer that uses convolution filters or kernels to run over

⁸A U-Net is a U-shaped CNN architecture

⁹The MNIST database is a large database of handwritten digits that is commonly used for training various image processing systems [153]

the input array and compute a dot product in order to extract different features from the input array. The kernel could be slid in increment of one or more cells, this is usually referred to as the stride. In order to introduce non-linearity to the model, activation functions are used to learn complex mappings between the input data and the response variables. Examples of activation functions include ReLU, sigmoid, tanh, leaky ReLU, etc. The ReLU activation function is the most common of them because it is a piece-wise linear function defined as follows

$$ReLU(x) = \max(0, x). \quad (2.1)$$

From equation 2.1, if $x > 0$, the function is positive and if $x < 0$, the output will be zero. The pooling layer decreases the size of the features that were convolved so as to reduce overfitting. The fully connected layer is very useful in capturing non-linear combinations of the high-level features. There could be more than one fully-connected layers. Depending on the application, the last fully-connected layer's output dimension is based on the number of classes. CNNs are capable of learning over both time and frequency domain, hence, their usefulness in modelling devices with output that is not restricted to frequency outputs alone [85]. Furthermore, CNNs are very useful when the data captures non-linearities in the time domain specifically for identification or classification tasks [32].

Prior studies have demonstrated that CNNs can automatically extract device-specific features by analyzing a huge amount of training data to improve the performance. For instance, authors in [32] presented deep learning techniques useful in fingerprinting several IEEE 802.15.4 devices. Further, another work [33] investigated the combination of ML algorithms with wavelet transforms to uniquely identify and classify disparate wireless transmitters. On a different note, DL-based modulation recognition solutions have also attracted

much attention in recent years. For example, [139, 155] leveraged CNN algorithms to recognize modulation schemes based on sequences of constellation points. Several other works [44, 156, 157] studied the application of deep learning to modulation recognition problems.

Some authors investigated interference classification using deep neural networks. For instance, [22] developed a technique for classifying interference at the PHY layer. The interference classified include single tone, chirp, filtered noise and unknown modulated signal. They extracted cyclic spectrum and spectrum information from the received signal. While their overall finding identified which spectrum information feature to extract, they did not consider ways to suppress interference sources.

Grunau *et al.* [5] proposed a wireless interference identification that classified multiple Bluetooth, Wi-Fi and IEEE 802.15.4 interfering signals in the presence of the desired signal using a CNN. In their work, they generated a *multi-label* dataset that consists of time- and frequency-limited sensing snapshots that combines the desired signal with multiple interference signals. While their work classified the interference sources with high accuracy, it did not address the more challenging task of how to suppress the interference from the different technologies. For interference suppression, however, authors of [105, 106] have shown that CNNs can learn to suppress interfering signals. We discuss these techniques in Section 2.6.1.

2.3.3.2 Autoencoders

Autoencoders (AE) have been studied as a technique to suppress interference in applications such as radars and Wi-Fi. AE are neural networks that aim to learn a compressed latent-space representation of the input data, that can be successfully reconstructed into the original data [158]. The latent-space representation can usually encode useful information about the interfering signal that can then be suppressed at the decoder. They are composed of the encoder which compresses the input into a latent-space representation and can be generally

represented by the function: $h = f(x)$ while the decoder reconstructs the input from the latent representation. The decoder can be represented by the function: $x' = g(h)$. For an autoencoder, the main task is to ensure that x' is as close as possible to the original input, x . Fig. 2.5 shows a block diagram of a generic autoencoder with encoder and decoder networks.

The latent space representation, h , has very useful properties since it learns very important features in the input data. At a high level, we can picture the autoencoder as a way of copying the input to the output. However, when the copying process is constrained, such as, limiting h to have a dimension smaller than x , the autoencoder is compelled to learn the most important features of the training data [159]. On the other hand, if the autoencoder has too much capacity, it may end up not learning vital information about the distribution of the input data [159]. The imposed constraint is analogous to intentionally introducing a restriction or bottleneck in the network which results in a compressed useful representation of the original input. The lack of dependence between the input features could make reconstruction difficult. On the other hand, if there is some correlation between the input features, the imposed bottleneck can be leveraged to learn the structure of the input [160]. Without the bottleneck or constraint, the network simply memorizes the input values and passes it along through the network to the output. The loss function is usually either mean square error (MSE) or reconstruction loss which is the cross-entropy between the input and the output. The network is penalized for creating outputs different from input. To prevent the autoencoder from simply memorizing the input, it is tuned in such a way that it is sufficiently sensitive to the input such that it can precisely build a reconstruction while also adequately insensitive enough to not overfit the training data [161]. One potential application of autoencoders for interference suppression is that h can also be pruned to suppress or reject interference.

Some general applications of autoencoders include data denoising[162], dimensionality reduction[163], anomaly detection[164], image in-painting[165], information retrieval[166], etc. In data de-

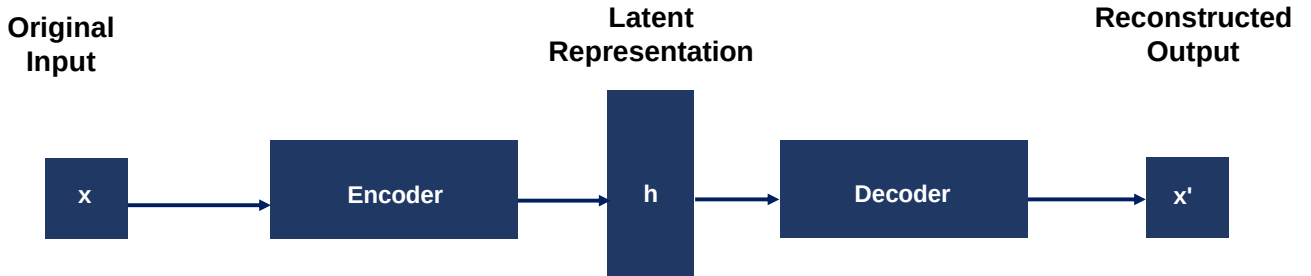


Figure 2.5: A Generic Autoencoder Architecture

noising applications, an AE learns a generalizable encoding and decoding function, thus, corrupting the input data with noise may not significantly impact accurate reconstruction. On the other hand, for dimensionality reduction applications, when compared to principal component analysis (PCA) or other dimensionality reduction techniques, autoencoders are able to easily learn data projections that are more useful [159]. This helps to map from high feature space to low feature space. As we discuss in Section 2.6.3, these unique features of autoencoders made them very crucial to detecting and suppressing interference. In Section 2.4, we discuss publicly available interference datasets.

2.4 Publicly Available Interference Dataset

Unlike in other deep learning application domains where there is a plethora of publicly available benchmark datasets, only a handful of datasets are available for interference suppression research. In addition, it is quite difficult to create a dataset that can be used for training and testing different models to address different problems in various wireless communication/networking applications. This is because the wireless channel is stochastic in nature and there can be different realizations at different points in time. More so, wireless channels are highly dynamic, and it is very likely the training data generated in one scenario could not be applicable to other scenarios. Hence, it is not surprising that there are no publicly available benchmark datasets for the vast majority of wireless research problems. Researchers work-

ing on these problems need to generate their own datasets, which is very resource-intensive and time consuming. However, datasets that capture unique environments are still valuable in training deep learning algorithms despite the fact that they are not representative of all channel environments. As noted earlier, SDRs can be leveraged or collecting data in multiple channel environments.

One of the key requirements for using deep learning to solve any problem is the access to benchmark datasets. Take for instance, in the image processing/computer vision and the medical bioinformatics research areas where deep learning techniques are very pervasive, there are a plethora of benchmark datasets available for use by their corresponding research communities. It is important to also point out that utilizing simulated and synthetic dataset is frowned upon by deep learning experts in fields such as computer vision and image processing. This is because the performance of such algorithms on these dataset types often skew their expectation for real world applications. For example, data collected in a simulated environment with AWGN channel and fading differ significantly from data collected over-the-air (OTA). Furthermore, access to benchmark datasets helps to further promote cutting edge research as well as innovation in the field. For instance, in image processing domain, the MNIST database consists of 70k images for both training and testing [154] while there is also an extended dataset, EMNIST [167] that has 280k images of handwritten digits and characters. Despite the paucity of such benchmark datasets in the interference suppression research, there are still a few authors and useful community resources where interference datasets can be accessed. In this section we provide the readers, who are interested in deep learning-based interference classification or suppression, high level information on these resources. Using these datasets and with appropriate scaling and sample rate conversion, synthetic datasets can be created using real data.

2.4.1 Radar Interference Dataset

Ristea *et al.* [105] automatically generated 48k samples of an automotive radar dataset with only one interference source. While this may seem insufficient, it is indeed a good contribution to the largely unavailable public generated databases for interference suppression. They fixed parameters such as bandwidth, sampling frequency, center frequency and sweep time while other parameters such as SNR, SIR, target amplitude, distance, and phase were selected using a uniform distribution between a lower and upper bound. Details of these values can be found in [105]. Their data samples contain three categories of data such as time domain signal with no interference, time domain signal with interference and a label vector with complex amplitude values in target location. The dataset is available for download in [168]. Radar signals are especially important datasets because of the extensive spectrum sharing interference between radar and communication signals, and the dynamic behavior of radar makes it difficult to simulate.

2.4.2 Radio Frequency Interference Dataset

The radio frequency interference (RFI) dataset was generated by Ujan *et. al* [169]. The dataset was created by combining the SOI with three popular jammers: continuous-wave interference (CWI), multi-continuous-wave interference (MCWI) and chirp interference (CI). In addition, the dataset was generated under different signal to noise ratios (SNR). The desired signal, a real-time video stream, was modulated using GNU Radio and then transmitted via a Universal Software Radio Peripheral (USRP-N210). The authors utilized a combiner to combine the respective jamming signal with the SOI. The receiver used was a MegaBee modem. Details of their dataset generation can be found in [169]. Fig. 3.2(a)

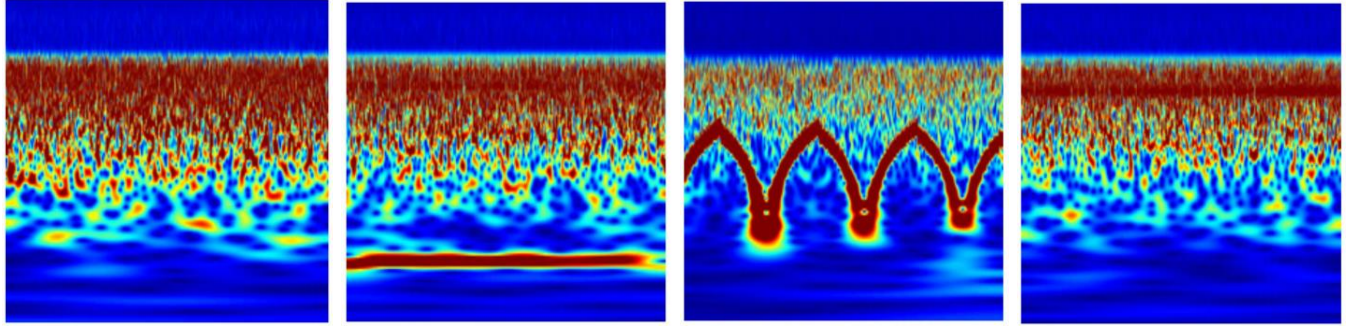


Figure 2.6: A scalogram plot of the RFI dataset from [169]. (Left: The original SOI with no interference; Center Left: The SOI with multi-continuous wave interference; Center Right: The signal of interest with chirp interference; Right: The signal of interest with continuous wave interference). The figure was reproduced from [169] titled: “An Efficient Radio Frequency Interference (RFI) Recognition and Characterization Using End-to-End Transfer Learning”; published as an open access article in the Journal of Applied Sciences.

shows a scalogram plot¹⁰ of the RFI dataset. It is clear from the scalogram plot that distinct features of each class of interference can be easily learned by a deep learning algorithm.

2.4.3 CRAWDAD Dataset

The Community Resource for Archiving Wireless Data at Dartmouth (CRAWDAD) was established to bridge the gap in collecting data captured from production wireless networks so as to understand how real users and devices use networks under real conditions [171]. At the time of writing, there are 125 datasets for different wireless communication applications and more than 14k users from more than 120 countries [171]. The dataset, contributed by Schmidt *et al.* [3], contains traces of IEEE 802.11b/g, IEEE 802.15.4 and Bluetooth packet transmissions with varying SNRs in the baseband [172]. The authors also added frequency offsets in the baseband [5]. In total there are fifteen different classes with ten of

¹⁰The scalogram represents the squared magnitude of continuous wavelet transform (CWT), a type of wavelet transform, which is used for non-stationary and transient signal analysis. The CWT is a non-numerical signal transformation function that continuously varies translation and scale parameter of the wavelets. It essentially provides a detailed or over-complete representation of a signal. Authors in [170] noted that CWT is resistant to noise in the signal.

those classes being IEEE 802.15.1 devices, three being IEEE 802.11 and the remaining two being IEEE 802.15.4 devices. There are two other interference datasets on the CRAWDAD database that are targeted towards the measurement of HTTP requests over 802.11 protocol in a dense wireless classroom [173] and a trace-set that includes received signal strength indicator (RSSI) for frames that were correctly received at receiver nodes when noise was injected [174]. For more on the CRAWDAD datasets, readers are directed to [171].

2.4.4 Wireless Interference Signal Dataset

Authors in [175] collected experimental data in four different environments with one of those environments having no detectable interference, another with controlled interference and the other two environments with uncontrolled interference from real devices. Interference sources for their experiments include Bluetooth and Bluetooth Low Energy (IEEE 802.15.1 and IEEE 802.15.1 BLE respectively), ZigBee (IEEE 802.15.4) and WLAN (IEEE 802.11). In total their dataset consists of more than 110k interference bursts of known origin. One key contribution of their work is the heterogeneous nature of the interference scenarios they considered which makes for a robust dataset that can be used by the wireless community at large.

Table 2.2 shows a high level comparison between the datasets mentioned in this work.

2.5 Deep Learning-Based Interference Classification Techniques

In this section, we discuss common deep learning-based interference classification techniques. Specifically, we discuss deep learning techniques under two categories: supervised and semi-supervised learning. Unsupervised learning techniques are not considered in this work be-

Table 2.2: Comparison Between Interference Datasets

Dataset	Interference type	Signal Types
Radar Interference Dataset [168]	Homogeneous technology interference	3 classes including: • Time domain signal with interference • Time domain without interference • Amplitude information for targets
Radio Frequency Interference Dataset [169]	Heterogeneous technology interference	4 classes including: • Signal of Interest (SOI) • Continuous-wave interference (CWI) • Multi-continuous-wave interference (MCWI) • Chirp interference (CI)
CRAWDAD Dataset [3]	Homogeneous and heterogeneous technology interference	15 classes including • Bluetooth (10) • WLAN (3) • ZigBee (2)
Wireless Signal Interference Dataset [175]	Heterogeneous technology interference	4 classes including: • Bluetooth • Bluetooth Low Energy (BLE) • ZigBee • WLAN

cause they are beyond the scope of this investigation.

2.5.1 Contributions for Interference Classification Using Supervised Learning

Many interference classification tasks have utilized supervised deep learning techniques such as CNNs to classify interference and then subsequently suppress them. CNNs, unlike feature-

based machine learning (ML) techniques do not require that a domain expert handcraft the features, instead the algorithm through the use of convolutional layers and a feature map, learn salient features necessary for classification, and thus, offer a more scalable approach to classifying or identifying interference. Furthermore, CNNs are very useful when the data captures non-linearities in the time domain. In this section, we describe works in the literature that utilized supervised learning techniques.

Authors in [4] utilized CNN to classify automotive radar systems interference signal interference with an accuracy of over 96%. Their work focused on classifying unmodulated continuous wave (CW) signal, slow chirp frequency modulated continuous waveform (FMCW) signal, fast chirp FMCW signal, pulsed CW signal and frequency shift keying signal. Their CNN model had 3 convolutional and fully connected layers respectively, they also used the ReLU activation function in all; the layers. Their data was, however, collected in a simulation environment under the additive white Gaussian noise (AWGN) channel with an SNR of 10 dB, hence the high classification accuracy. It is important to note that most deep learning experts frown on using simulation data since it may hamper the performance of the model on real-world data.

Furthermore, authors in [3] utilized CNN to distinguish between 15 classes of packet transmissions that represent wide-band high rate IEEE 802.11 b/g, narrow-band low rate IEEE 802.15.4-based Wireless-HART and ISA 100.11a, IEEE 802.15.1-related PNO wireless sensor actuator network for factory automation (WSAN-FA) and Bluetooth technologies in the 2.4GHz ISM band. Their approach claim to outperform the neuro-fuzzy signal classifier (NFSC), an expert system used to classify frequency bands to their corresponding known wireless technologies. NFSC, a state-of-the-art model, uses pre-defined features and a rigid decision process for such classification [176]. The work of [3] is quite robust in the sense that they considered 19 different modulation types and symbol rates in a spectrum band that

is shared by proprietary wireless technologies like the IEEE 802.11-based industrial wireless local area network (iWAN) developed by Siemens AG, frequency hopping spread spectrum (FHSS) based Trusted Wireless from Phoenix Contact and the popular IEEE 802.15.1-based WISA from ABB group.

As noted by the authors of paper [3], they made three key assumptions that further helped them achieve a 95% accuracy. They assumed a single-label classification, flat fading channel and additive white Gaussian thermal noise reception distortions. It is our opinion that while all three assumptions are valid and necessary for a realistic wireless transmission, there are cases where either one or none of those assumptions would hold. For instance, the delay spread in flat fading is less than the symbol period whereas the opposite is true for frequency-selective fading which may lead to inter-symbol interference (ISI) in the latter. We believe that this assumption may imply that their classification algorithm may not perform well when a frequency-selective fading is considered. The other assumption made in [3] is the single label assumption for which no concurrent signals were allowed (i.e. only signals of exactly one class is present in each input sensing snapshot). Again, this assumption may not truly represent a typical wireless band. For the last assumption where thermal noise reception distortion was assumed, an additive white Gaussian noise (AWGN) was added using MATLAB Simulink model with an SNR range of -20dB to 20dB.

Pulkkinen *et. al* [85] conducted a performance evaluation of deep learning based interference detection. One of the deep learning algorithms used is the CNN. They considered FFTs of the frequency band as input into the algorithm. According to the authors, “*spectrum sweeps make it easier to identify interference signatures*” [85], and thus, subsequently help the feature mapping process. One key issue that they identified is that systems built to detect interference should be robust in diverse environments even when little to no domain knowledge is known. The implication of this is that such detection systems can generalize well

to the complexity of regular wireless environment since deployment sites and RF environment vary in actual production environment. Similar to many other approaches that utilized deep learning in wireless communication, the authors evaluated their result using confusion matrices. Overall, they achieved a classification accuracy of 73% for an 8-class classification with SNR range from -20 to 20dB in a cross-environment scenario. Despite the seeming issues they identified with using deep learning for interference detection, the overall accuracy of their proposed approach was 73% which is quite below the performance of the CNN algorithm that they used to compare their proposed approach.

Authors in [5] also investigated the use of CNN to identify multi-label wireless interference. They utilized the same dataset used by [3] with each snapshot containing one utilized signal with one or multiple interfering signals. They achieved a classification accuracy of 100% in their STI classification while their CTI performance was about 90%. The technique they utilized to generate the multi-label dataset was particularly interesting. In their work, they generated a synthetic dataset with a vector signal generator (VSG) and used a real-time spectrum analyzer (RSA) for data acquisition. The synthetic dataset consisted of several snapshots of the data as well as the label. Each snapshot for the multi-label dataset requires several technology classes in which one class is the utilized signal while the others are the interfering signals. They kept the signal to interference ratio (SIR) constant. The interfering signal were, however, weighed with the factor $1/N$ (N represents the number of classes) so as to keep the SIR constant. Instead of the regular softmax function as the last layer of the CNN as done by [3] for the single label classification, they used the sigmoid activation function so as to enable an independent calculation for each class [5]. The remaining part of the architecture is similar to the ones used by [3] and [8]. Studies that investigate the use of deep learning for interference identification are very crucial to ensure coexistence of technologies in license-free frequency bands such as the ISM bands and, consequently the

Citizens Broadband Radio Service (CBRS) band which ranges from 3.55 GHz to 3.7 GHz.

2.5.2 Contributions for Interference Classification Using Semi-Supervised Learning

Semi-supervised learning have also been utilized in prior works. For instance, authors in [31] investigated a CNN to detect interference sources from wireless local area networks (WLAN). They collected data for 13 different devices using a spectrum analyzer to sweep the 2.4 GHz spectrum. One key aspect of their work is the use of semi-supervised learning to detect interference. Semi-supervised schemes uses both labeled and unlabeled data to train the classification model. Their CNN model achieved an error rate of 2% when compared to 21% error rate for the baseline method. The baseline method consisted a technique that used the spectral signature feature to describe each device's spectrum use and their angular difference as a similarity metric. The angular difference between all training signatures and the spectral signature of the testing window was calculated and the signature with the smallest angle was picked to successfully detect the classes. Essentially, their technique identified non-Wi-Fi RF devices mainly from spectral data. The major limitation of their work is that only one device could interfere when the main device is transmitting. It is unclear if their work will maintain the same performance when multiple sources of interference is considered similar to [5]. Hence, it is very crucial to develop solutions that can classify signals or devices in the presence of multiple interference sources since this would be directly applicable to modern applications such as IoT, 5G and other applications that require multiple devices to share a frequency band.

2.6 Deep Learning-Based Interference Suppression Techniques

In this section, we survey the most popular DL-based interference suppression techniques shown in Figure 2.3. The most common deep learning techniques discussed in this section include CNN, autoencoders, RNN and deep neural networks (DNNs). While there are other deep learning techniques in general, our research shows that these four techniques are the most common in the interference suppression literature.

2.6.1 Contributions for Interference Suppression Using CNN

In this section, we investigate CNN-based techniques for interference suppression. CNNs, an example of a model-free method, have shown impressive performance in many other domains (e.g. radio fingerprinting [32], image classification [167, 177], etc). CNNs can utilize only raw IQ data or other channel-based information to learn the unique features in interfering signals [34, 178, 179]. Typically, CNN based techniques include layers of convolution, pooling ReLU and batch normalization operations as shown in Fig. 2.7. CNNs, unlike feature-based ML techniques do not require that a domain expert handcraft the features. Instead, the algorithm through the use of convolutional layers and a feature map, learns salient features necessary for suppressing interference. CNNs are capable of learning over both time and frequency domains, hence, they are useful in modeling devices with output that is not restricted to frequency domains alone [85].

While earlier works [3, 4, 5, 85] have investigated interference classification using CNNs, several recent works have shown that CNNs can also go beyond classifying interference to actually suppressing them. For instance, Rock et. al. [106] utilized CNNs to find structured

information in the input data to suppress automotive radar interference. One key application of their CNN model is to de-noise and suppress interference using a dataset that consists of clean data and data with interference. They used metrics such as signal to interference ratio (SIR) and signal to noise ratio (SNR) to scale the interference and noise powers in relation to the actual object signal power. In their CNN model (shown in Fig. 2.7), the first layer consists of a convolution layer and a ReLU activation function while the next four layers consist of batch normalization¹¹, convolution and ReLU activation function operations, the last layer (before the output layer) consists of a batch normalization and convolution operations. Convolutional layers in CNNs often learn unique artefacts in the input data that help in identifying interfering signals in the data. Their results show that the CNN based suppression approach outperforms non-deep learning techniques. For instance their CNN based suppression model outperformed zeroing, ramp filtering and IMAT techniques by more than 40% using the signal to interference plus noise ratio (SINR) metric [106]. Despite these gains, the CNN-based suppression techniques suffered a relatively high error vector magnitude (EVM) performance when compared to the conventional suppression approaches. For radar applications, such as the one considered in the authors' study, a high EVM may result in distortions in object peak values [106]. It is important to point out that the study done by Rock *et al.* [106] utilized a simulation framework. This may imply that some of parts of their results may not generalize to over-the-air (OTA) applications. This is because real world OTA applications may be subject to more channel uncertainties such as fading, noise, mobility and other factors. It is fair to say that one might expect different results for QAM communication signals with information carried in the signal phase.

Similar to [106], Ristea *et al.* [105] utilized a fully convolutional neural network, another CNN approach, to learn the patterns in the input data using a small amount of learnable

¹¹Batch normalization is usually used in deep learning training to reduce the number of epochs and stabilize the learning process by standardizing the inputs to a layer for every mini-batch [180].

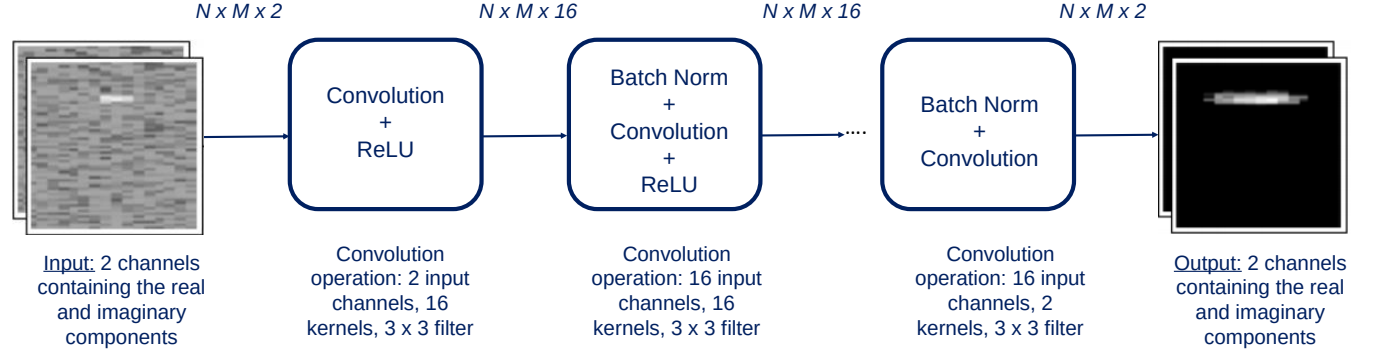


Figure 2.7: CNN Architecture proposed by Rock *et al.* [106] for radar signal denoising (N represents the number of fast-time samples and M represents the number of slow-time samples/ramps). The figure was reproduced from [106] titled: “Complex Signal Denoising and Interference Mitigation for Automotive Radar Using Convolutional Neural Networks”; published at the 22nd IEEE International Conference on Information Fusion (FUSION).

parameters. They proposed two deep learning architectures that outperformed conventional zeroing techniques of interference rejection [96] by taking spectrograms of input signals (with 1024 samples) as input and then arriving at clean signals at the output. The input signals consist of noise and uncorrelated interference which is learned using the fully convolutional neural network (FCN)¹². To obtain the beat signal, they first convert the time domain signal into a spectrogram by using short time Fourier transform (STFT). To get the ground truth, the authors perform an FFT on the time domain labels and consequently trained the network to map the STFT input to the FFT output. Generally, in the wireless communication community, many researchers utilize the I/Q samples to train the deep learning architecture. A good lesson from this work stems from the row dimension of the I/Q samples having a value of 2, the filters used in the convolutional layer may be too small and hence leading to overfitting or further complicating the CNN algorithm. This is because using small sized filters tend to make the model less generalizable, and thus, learn specific features in the training data.

In another work, Rock *et al.* [107] investigated CNN-based quantization techniques to mit-

¹²A fully convolutional network (FCN) differs from a CNN in that it does not contain the last fully connected layer typically used for classification [54].

igate or de-noise interfering radar signals. Specifically, their work focused on memory and computational requirements associated with using deep learning models. The major motivation for this is that high memory requirements could limit the energy efficiency and the time it takes to execute operations when an embedded hardware is used [107]. To account for this, the authors explore quantization capabilities of different CNN architectures. Their results (shown in Fig. 2.8) showed that the CNN-based mitigation models performed better than the three traditional interference mitigation methods considered (zeroing, IMAT and ramp filtering) and are also resilient against different interference levels and patterns. They also considered interference signals from real-sensor measurements that was combined with an object signal from an inner-city measurement campaign, their result showed an SNR improvement of 7dB after CNN-based mitigation [107].

Further, Fuchs *et al.* [108] proposed a complex valued CNN (CVCNN) to suppress mutual interference in automotive radar systems. The main challenge the authors sought to address was the problem of reconstructing the phase of clean signals from the real and imaginary parts of the complex spectra. This is because many object detection algorithms typically rely on only the magnitude spectrum component of the interference mitigated maps, and thus may not include important phase information. While including the phase information adds more complexity to the algorithm, complex-valued analysis can also restrict the degrees of freedom [108]. In [108], the authors found that such a restriction can leverage the physical characteristics of the data to enforce signal transformations thereby reducing the complexity. Their CVCNN model, thus, uses complex-valued analysis for all the operations. This implies that the interconnection between real and imaginary parts in the model architecture are taken into consideration. This also implies that the algorithm does complex convolution, complex batch normalization and complex ReLU activation function. Their results showed that the CVCNN method outperformed the real-valued CNN, zeroing, RFmin and IMAT techniques

in terms of F1-score, EVM and peak phase MSE (PPMSE)¹³.

In practice, a CNN model should be able to generalize well so that it can appropriately classify samples from the test data. To solve this, many researchers use preprocessing techniques such as spectrograms [105] or continuous wavelet transforms (CWTs) [7, 181] to further increase the dimension of the input data along the time-frequency axis. It is very important to preprocess time domain signals to produce a 2D image representation so that CNNs can better learn salient features in the input data. In [105], the authors used a spectrogram which led to an input size of 154×2048 . Their results showed that the FCN algorithm moderately outperformed the conventional suppression method such as zeroing using the area under the receiver operating characteristics curve (AUC) metric. We note that their technique improves the SNR and AUC scores of the actual signal and not just the spectral representation. This goes to show that combining traditional signal preprocessing techniques with deep learning can significantly improve interference suppression performance while reducing overfitting [182].

It is not surprising to see that the research community is slowly adopting the idea of preprocessing input signal dataset since it provides an opportunity to explore the best of both worlds (i.e. signal processing and deep learning applications). More so, preprocessing techniques (such as continuous wavelet transform (CWT)) can be very useful in learning attributes unique to the transceivers, such as non-linearity in the power amplifier, oscillator imperfections, amongst other unique attributes. In over-the-air transmissions, both wireless interference and the stochasticity of the wireless channel can result in the attenuation of unique attributes in the transmitted signals [183]. To magnify these attributes, raw IQ signals can be pre-processed using signal processing transforms similar to CWT such as discrete wavelet transform (DWT), STFT, Fourier transforms amongst others. Since Fourier

¹³PPMSE is the measure of the average squared difference between the angle of the clean range-Doppler signal vector and the predicted range-Doppler signal of all detected object peaks [108].

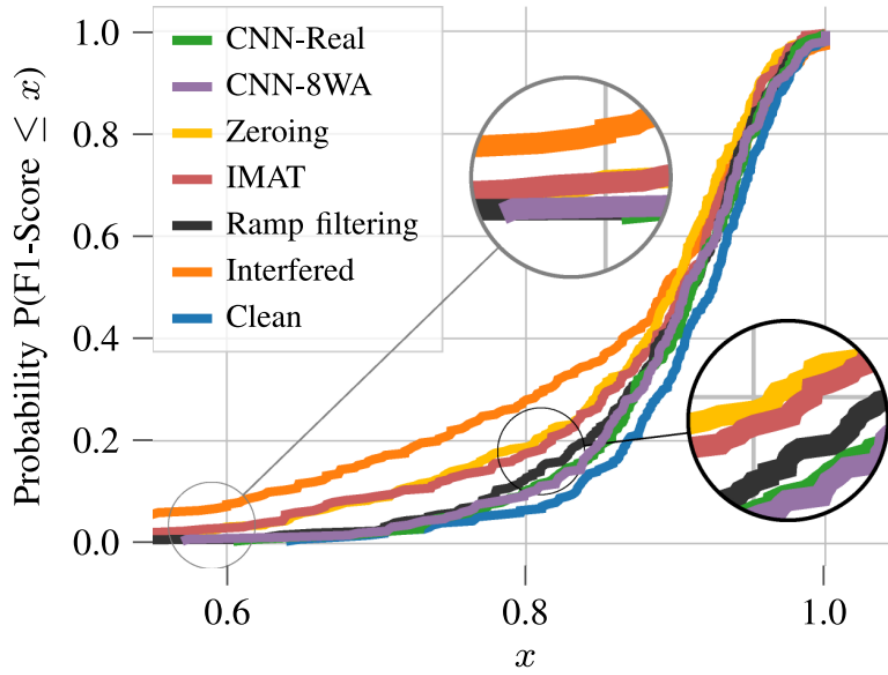


Figure 2.8: A CDF comparison of the sample-wise F1-Score between two CNN models, zeroing, IMAT and Ramp filtering interference mitigation techniques (*where x is a real-valued number*). The figure was reproduced from [107] titled: “*Resource-Efficient Deep Neural Networks for Automotive Radar Interference Mitigation*”; published in IEEE Journal of Selected Topics in Signal Processing, vol. 15.

transforms do not mark the start and the end of the sine wave, the STFT analyzes several smaller section of the signal. In DWT applications, a sparse representation of the signal is provided. CWT is very useful in enabling a clear representation of many frequency components of a signal as a function of time. The CWT introduces spatial correlations that may not be easily observable in the original signals. The unique features of each transmitter is highlighted in a 2D representation, thus improving the performance of the CNN classifier [7]. As a result of this, CWT is a better choice for signal or time series analysis because of its fine grained resolution. Such a resolution is obtained by decomposing the signal into wavelets using basis functions (scaled and shifted versions of the mother wavelet [181, 184]). Finally, a two-dimensional matrix where the rows correspond to the scales and the column corresponding to the length of the signal is then generated. It has been shown that combining

signal preprocessing techniques with CNN can significantly improve performance [181].

2.6.2 Contributions for Interference Suppression Using RNN

Some other works including [120] have used recurrent neural networks (RNN) with gated recurrent unit (GRU) to remove automotive radar interference and to simultaneously reconstruct the transmit signal. Unlike many feed-forward neural networks where the collected samples are independent of each other, RNN leverage the dependencies. RNNs capture the dynamics of sequences via cycles in the network of nodes [185]. Since the received samples are collected sequentially in the time domain, RNNs can indeed be used since previous studies show that they are very useful for sequential data. Similar to the deep learning methods discussed, RNNs performed better than time domain thresholding described in [186] and the threshold-free frequency domain interference cancellation method that attenuates the magnitude response of the recorded chirps described in [102].

From the foregoing, we observe that Mun *et al.* [120] successfully processed sequential data using an RNN model with a gated recurrent unit (GRU). However, they were not able to reconstruct the original signal in the face of interference. To account for the inability of their RNN model to reconstruct the original signal, Mun [121] proposed an RNN with self attention to mitigate OFDM-based and frequency modulated continuous waveform (FMCW) radar interference. To restore the original signal perfectly, the RNN model learned the information on the surrounding signals and then applied an attention mechanism [187] to clearly detect the range and velocity of the target signal using an FFT. Specifically, the authors leveraged the attention blocks to capture the sequential relationship between the time steps since the interference partially occurred in the entire data. Fig. 2.9 shows a 2D FFT plot of the received signal with radar interference and the same signal after the radar interference has been removed. This plot shows the power of RNNs in suppressing

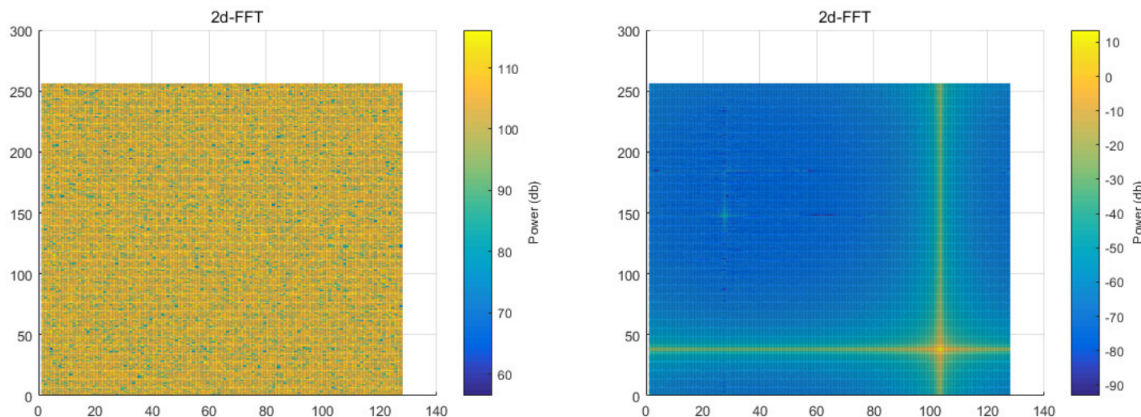


Figure 2.9: A plot showing the 2D FFT result of an RNN-based Interference Mitigation Approach *Left: With radar interference; Right: After radar interference has been removed with RNN*. The figure was reproduced from [121] titled: “Automotive Radar Signal Interference Mitigation Using RNN with Self Attention”; published in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

interference in radar applications. It is important to note that their technique was tested on only simulated data and results of the real world equivalent is not provided.

On a different note, Mao *et al.* [122] investigated an interference suppression technique for global positioning system (GPS) using a derivative-free Kalman Filter-based RNN. An adaptive unscented Kalman Filter-based RNN (UKF-RNN) was used to suppress continuous wave interference (CWI), multi-tone CWI, swept CWI and pulsed CWI. While researchers have shown that RNNs are better at modeling non-linear time series systems than signal processing methods [188], some interfering signals may be non-stationary, hence, a model that captures this phenomenon is even more useful for suppressing non-stationary interference. In general, results in [122] show that their UKF-RNN algorithm increased SNR improvement while also attenuating the mean square prediction error (MSPE) of the received signals in the presence of interference when compared to other conventional methods. With regards to non-stationary interfering signals, the authors’ UKF-RNN algorithm offered SNR improvements between 3.34 dB to 8.30 dB more over other methods [122].

2.6.3 Contributions for Interference Suppression Using Autoencoders

Another important deep learning technique utilized for interference suppression is the deep autoencoder. Autoencoders are very unique in interference suppression because the encoder network can learn unique interfering components of the input signal (with interference) in the latent space representation. Authors in [125] investigated a CNN-based autoencoder technique that was used to suppress automotive radar interference. Similar to [106], their approach involves using a denoising mechanism to suppress interference. They first trained corrupted input signals with their clean non-interfered version. Using deep learning models typically requires access to a huge amount of quality data. Their approach ensured that they collected a large amount of range-Doppler images.

Similarly, Lin *et al.* [54] proposed a method that used denoising autoencoders to learn patterns of interference with *unknown* structures to suppress heterogeneous technology interference. The main idea behind this approach is to allow the algorithm to learn the distribution of the input data and thus consequently learn the distribution of the interference component so that it can be suppressed. It is important to note that in [54], the authors did not assume any form of coordination or synchronization between the transmitter, receiver and interferer. The authors utilized a generative adversarial network (GAN) to generate synthetic training samples and an autoencoder (denoising) to recover a clean Wi-Fi signal. The synthetic training samples were generated as a means of avoiding the steep cost of collecting a huge amount of data in terms of time and resources. Hence, the GAN was used to synthetically create corrupted Wi-Fi signals that the authors used for training their autoencoder-based suppression model. GAN is basically a network that involves a generator and a discriminator that play a zero-sum game where the generator generates data that is as close as possible

to the real data and the discriminator determines whether the generated data is real or fake [143]. Specifically, their experimental set up included a USRP N200 SDR (within interfering range of a microwave oven) that was deployed as a receiver to collect the signal transmitted when the microwave oven is turned on while Wi-Fi signals are being transmitted. One key lesson learned from their work is that denoising autoencoders perform well when the interfering power is larger than the amplitude of the desired signal (Wi-Fi, in their scenario). Another key benefit of this approach is that it does not incur any signalling overhead since the algorithm itself is passive [54].

Further, Chen *et al.* [51] investigated an approach that combined an autoencoder network with an interference detection filter to suppress automotive radar interference. The authors found that using a gated convolution, the encoder network can learn the pattern of the residual signal free of interference. In the decoder network, the interfering portion of the signal can then be recovered. Their results showed that the method improved SINR while also ensuring robustness of the radar measurements in scenarios that have significant interference not represented in the training dataset. Fig. 2.10, reproduced from [51], shows a plot of the SINR (in dB) against the percentage of discarded samples that compares different interference mitigation techniques such as zeroing, auto-regressive (AR) models, autoencoders and the original signals. We see that at 15% (or greater) of discarded samples, the SINR of both AR and zeroing methods decreased while the autoencoder has a more robust performance.

Channel autoencoder-based techniques have also been used in joint optimization of transmitters and receivers to suppress interference [52, 53, 126, 127, 128]. For instance, a channel autoencoder-based interference suppression technique where an encoder and a decoder were jointly optimized in the presence of an interfering signal in a MIMO communication system was investigated by Erpek *et al.* [53]. The authors considered a Gaussian co-channel interference scenario, where the transmitters and the receivers use the same channel. Their

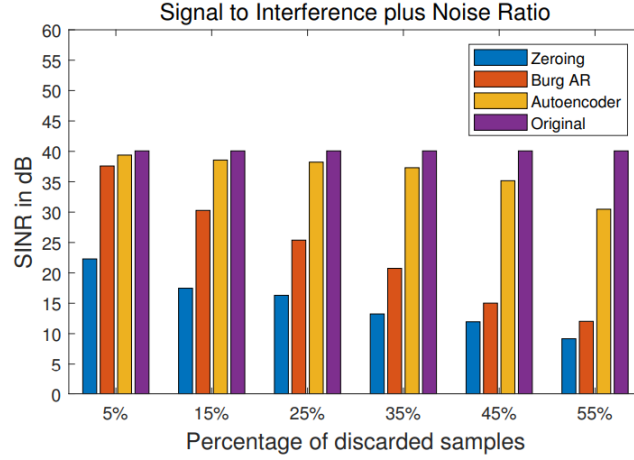


Figure 2.10: Comparison of Autoencoder-based Interference Mitigation technique with Zeroing and Auto-regressive models. The figure was reproduced from [51] titled: “A DNN Autoencoder for Automotive Radar Interference Mitigation”; published in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

technique exploited the structure of the interfering signal and then minimized symbol error rate (SER). Their results show that for a 2×2 system at SINR higher than 16 dB, the autoencoder-based model performs significantly better than the baseline (single user single input single output (SISO) systems using QPSK modulation with no interference and no channel coding). One key benefit of jointly training the transceiver in the autoencoder system is that the receivers learn to suppress interference and perform better in comparison to baseline techniques. Despite the fact that the baseline approach may perform better when channel coding is utilized, the autoencoder based suppression scheme from [53] still significantly outperformed the uncoded conventional communication system because it removes the interference. The authors also showed that the autoencoder performance will continue to improve as the network is trained longer at every SINR level.

2.6.4 Contributions for Interference Suppression Using DNNs

Next, we look at deep learning techniques that utilized deep neural networks (DNN) to suppress interference. Typically, many other deep learning algorithms can be regarded as

subcategories of DNNs. In this section we consider neural networks, without a convolutional structure, that have at least 4 layers with the inclusion of the input and output layers.

DNN-based methods are particularly useful in self-interference (SI) cancellation applications since they can successfully learn a large number of parameters with a much lower computation complexity. For instance, Muranov *et al.* [114] proposed two DNN-based methods: time-invariant & time-varying non-linear distortion ; to mitigate non-linear SI in full duplex relay link. For both methods, they utilized a feed-forward DNN that has two hidden layers and ReLU activation function. For the former, they utilized a DNN to estimate the non-linear distortion that does not vary with time and then utilized a least squares method to estimate the SI for a time-varying channel. For the latter method, they used a transfer learning based DNN to estimate the time-varying non-linear distortion. First, they decompose the SI signal into its linear and non-linear components and then estimated and subtracted the linear component from the SI which is then fed to the DNN. Fig. 2.11 shows a comparison between the investigated approaches: time-invariant non-linear distortion (TID) and time-variant non-linear distortion (TVD) and other methods used for SI cancellation. We note that the DNN-based canceller shown in Fig. 2.11 proposed by [68] is another non-linear interference cancellation technique that was developed to outperform non-deep learning non-linear cancellation techniques that relied on polynomial bases functions. Similar to [114], DNN was used to construct the non-linear component of the digital signal. The major difference between [114] and [68] is that the latter utilized only one hidden layer and is thus less complex than the former. This is not surprising since the goal of [68] was to improve computation complexity in comparison to non-DNN techniques like polynomial-based cancellers [87]. In [117], the authors investigated a hardware implementation of the SI cancellation technique proposed in [68].

In another effort to suppress SI, Guo *et al.* [115] utilized a DNN technique to cancel non-

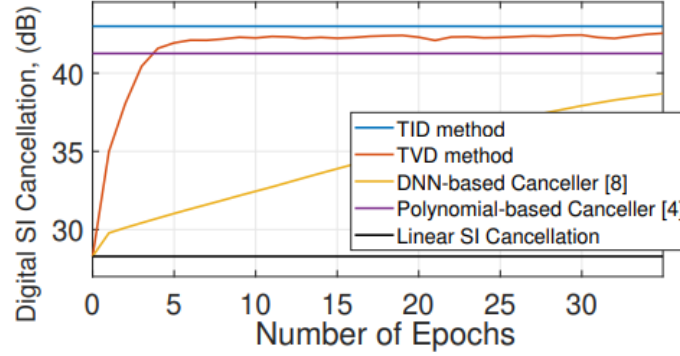


Figure 2.11: Comparison of DNN-based Self Interference Suppression Techniques with non-DNN based techniques. The figure was reproduced from [114] titled: “On Deep Learning Assisted Self-Interference Estimation in a Full-Duplex Relay Link”; published in IEEE Wireless Communications Letters.

linear SI. They collected channel data using an SDR testbed consisting of universal software radio peripheral (USRP) in different scenarios transmitting images, texts and numbers. Their DNN structure consist of two hidden layers each followed by ReLU activation function layers. Simply put, their approach showed that their DNN model is stable, fast and can successfully cancel SI with a high accuracy. Shi *et al.*[189] investigated a feed forward NN to reconstruct SI signal. First, an adaptive linear filter was used to extract the transmitted baseband signal features and then the features were utilized to train the NN. Similar to [68], their feed forward NN consist of one hidden layer and suffers from a similar shortcoming. It is important to note that while DNN techniques outperform non-DNN techniques in SI suppression across different channel environment and modulation systems, we also observed that other DNN-based SI suppression techniques either reduce computation complexity or require fewer hardware resources and may not necessarily outperform polynomial-based SI suppression techniques.

Another work that utilized a DNN for interference suppression is [109]. In this work, Xu *et al.*[109] developed a technique to cancel interference for non-orthogonal multiple access (NoMA) techniques using a DNN. In OFDM based systems, the sub-carriers are orthogonal

to one another, hence the issue of cross interference is prevented. However, OFDM-based techniques usually require that the sub-carriers are spaced and the cyclic prefix (CP) is used to ensure that the receiver accurately demodulates the signal. This can sometimes impact spectral efficiency [190] since some parts of the much needed bandwidth is required to accommodate the overhead of adding the CPs. On the other hand, NoMA algorithms such as the spectrally efficient frequency division multiplexing (SEFDM) waveform utilize non-orthogonal sub-carriers, hence leading to interference from adjacent sub-carriers [191]. In [109], their neural network consisted of four layers including two hidden layers. Compared to some of the CNN algorithms described above that utilized the ReLU activation function, the authors in [109] utilized a sigmoid activation function at each layer. They trained a DNN algorithm offline with $40k$ QAM symbols generated using simulated random data. In evaluating the performance of their DNN, they utilized the bit error rate (BER)¹⁴ metric for comparison. Although theirs is a simulation-based investigation, their key conclusion is that deep neural networks can suppress inter carrier interference (ICI) within a SEFDM signal waveform. Also, they achieved a significant gain in SNR when compared to conventional techniques that utilize hard decision detectors. Another key findings is that the neural network performance depends strongly on the waveform characteristics. For instance, the first neural network considered in [109] is the *no connection-neural network (N-NN)* in which each symbol in one sub-carrier is an independent network and the number of sub-networks is determined by the quantity of sub-carriers in the SEFDM signals. In another neural network proposed—*partial connection-neural network (P-NN)*—two symbols are connected in one sub-net. Other neural network architectures are discussed in [109].

In another DNN-based interference suppression approach, Hager *et al.* [110] investigated a learned digital backpropagation (LDBP) technique to mitigate non-linear interference for

¹⁴Mathematically, BER can be defined as a ratio of errors to the total number of bits. When a signal is strong, the corresponding BER is small while the converse is true when a signal is weak.

a fiber optic communication application. The designed network is based on unrolling the split-step Fourier method (SSFM)¹⁵ that has a similar functionality as a DNN [110]. In the LDBP approach, the SSFM serves as a basis for a complex-valued DNN and then is further used to optimize the network parameters. The network parameters include the weights, attenuation, dispersion, and non-linearity parameters. Their results show that the LDBP achieves a 50% complexity reduction when compared to the DBP. Also, one key benefit to using this approach is that the LDBP method helps in choosing the appropriate hyper-parameter choices (e.g., number of network layers, type of non-linearity, etc.) that usually plagues DNN algorithm adoption. Compared to standard “black-box” deep NNs, this approach leads to clear hyper-parameter choices.

Further, Promsuk *et al.* [111] proposed a multi-layer perceptron (MLP) algorithm to reduce ACI in an IoT network. In their proposed model, they utilized a FFT data and amplitude as input to the MLP model. The input signals were binary phase shift keying (BPSK) and 16QAM modulated signals that were synthetically generated with path loss and small scale fading to mimic a more realistic IoT network environment. To mitigate the ACI, they combined a MLP with a minimum mean square error (MMSE) filter. The output from the model when the input data is the amplitude is the estimated amplitude of the interference signal. The estimated amplitude of the received signal is then evaluated using a system of equation used in [111]. On the other hand, with an FFT input signal, the output is the estimated FFT of the interference signal. An inverse FFT of the signal is then computed. Their results showed that the performance of the MLP model for both types of input data outperforms the method that utilized the MMSE alone in terms of accuracy, precision, recall and F1-scores.

In another effort, Mosavi *et al.* [112] investigated a DNN interference suppression technique

¹⁵SSFM is a numerical method to implement DBP [110].

for mitigating narrow-band interference for global positioning systems (GPS) navigation applications. Their method leverages adaptive notch filters as a cascade filter and a simple DNN structure. The DNN structure can be trained quickly because the adaptive notch filters are used instead of updating weights and threshold typically done in MLP models. This technique also reduces exponential computation incurred while updating weights and thus makes it attractive for hardware implementation [112]. Their simulation results showed that they were able to improve SNR of the intended receiver in the presence of single-tone continuous wave interference (SCWI) and multi-tone continuous wave interference (MCWI) when compared to MLP, RNN and other compound methods.

Finally, Shi *et al.* [113] proposed an approach to suppress interference in a direct sequence spread spectrum (DSSS) system using a combination of a power distributing predominance wavelet packet transform (PDP-WPT) and an extended backpropagation neural network (EBPNN) in a satellite navigation system. The process involves a step that first tracks the interfering signal effectively and then identifies the transmitted signal using the EBPNN based on its structural simplicity. Their results show that the proposed method can eradicate strong interference in the received signal with a good signal correlation. Also, the mean square error (MSE) of the proposed technique is smaller compared with other conventional techniques. Another benefit of the technique is its higher convergence rate when compared to other time-domain interference suppression methods [113].

In summary, deep neural networks can be used to model and extract interference. One major benefit of DNNs over analytical models is that they can easily adjust to real-world imperfections [110]. Next, we compare the benefits and drawbacks of each deep learning techniques. This comparison helps the reader to gain some insights into why deep learning techniques, in general, are being adopted in interference suppression and what challenges may be associated with using them.

2.7 Comparison Between the Approaches

Table 2.3 contains a summary of selected DL-based approaches used by existing works in the interference suppression literature. We look at features such as application/type of interference, deep learning methods used, the type of dataset and the performance metrics used for evaluation. The interference types considered include radar, optical, and a variety of wireless communication standards, amongst others. From our findings, there are more works on CNN for interference rejection or suppression compared to other deep learning techniques.

2.7.1 Application/Interference Type

In works where the type of interference considered is not explicitly stated we provide the application considered (for example [109] with optical communication systems). Other interference types considered include heterogeneous technology interference between different wireless technologies sharing the same frequency band such as those of [54, 192]. Some of the most popular applications where interference suppression was studied is the automotive radar applications such as [105, 106, 107, 108, 125].

2.7.2 Deep Learning Algorithm

The deep learning algorithms range from supervised learning techniques such as vanilla deep neural networks (DNN), CNNs, RNNs to unsupervised learning techniques such as autoencoders that utilize a network of encoders and decoders to learn salient features of interfering signals from input data and then consequently removing them from the signals. It is not uncommon to train the autoencoder encoder or decoder using supervised algorithms such as CNNs as done by [125]. It is important to point out that using deep learning

Table 2.3: Dataset Analysis and Performance Comparison of Selected DL-Based Interference Suppression Techniques

DL Algorithm	Application/ interference type	Dataset type	Dataset Size	Performance Metric
CNN [105]	FMCW radars	Simulation	48k	SNR, area under the receiver operating characteristics curve (AUC), mean absolute error (MAE) in dB
CNN [106]	FMCW radars/Chirp Sequence (CS)	Simulation	2,250 scenarios	SINR; EVM
CNN [107]	FMCW radars/CS	Simulation & OTA + simulated interference	3k range Doppler (RD) maps	F1 Score; RD magnitude spectra; SNR
CNN [108]	FMCW radars/CS	OTA + simulated interference	2.5k range Doppler (RD) maps	F1 Score; EVM; PPMSE
DNN [109]	Optical Communication Systems	Simulation	40k	Bit error rate (BER)
Learned DBP [110]	Non-linear interference in optical communication systems	Simulation	Unknown	Transmit power vs Q-factor
Artificial Neural Network - Multi-layer Perceptron (MLP) [111]	IoT Network	Simulation	Unknown	CDF of BER
DNN [112]	Narrow-band interference for GPS navigation application	Simulation	62.854k test samples	SNR; RMS
DNN - Extended back-propagation NN (EBPNN) [113]	Direct sequence spread spectrum (DSSS) system	Simulation	Unknown	BER; MSE; Correlation value
DNN [114]	Full Duplex Relay Link	Simulation	8k (training data size)	BER, SNR
DNN [115]	In-band Full Duplex Wireless	Simulation & OTA	Unknown	MSE, BER
Feed Forward Neural Network [116]	Full Duplex Wireless Communication Systems	Simulation	Unknown	PSD
Neural Network [117]	Full Duplex Radios	Hardware Experimentation	Unknown	Throughput, Efficiency, Si Cancellation in dB
Feed Forward Neural Network [119]	Direct Sequence CDMA (DS-SS) (CDMA)	Simulation	Unknown	BER, SINR, SNR
Gated Recurrent Units (GRU), recurrent neural network (RNN) [120]	Time domain interference for chirp sequence radars	Simulation	150k	Signal to remaining interference noise ratio (SRINR)
Attention RNN [121]	Triangle FMCW; CS; multiple frequency shift keying (MFSK); OFDM radar interference	Simulation	150k	SINR; 2D- FFT
RNN [122]	Single-tone CWI; Multi-tone CWI (MCWI); Pulsed CWI, Linear FMCW	Simulation	Unknown	SNR Improvement; mean square prediction error (MSPE)
Denoising Autoencoders (DAE) [54]	Heterogeneous technology interference (Wi-Fi, ZigBee, LTE-U)	OTA, synthetic (GAN augmented data)	8.7k	Symbol error rate (SER), signal-to-noise ratio (SNR), interference power (dBm)
Autoencoder-based convolutional neural network [125]	FMCW radars	OTA	8.78k	Error vector magnitude (EVM), signal-to-noise-plus interference ratio (SINR), absolute & phase structural similarity index (SSIM)
Autoencoder [51]	FMCW/CS for automotive radar applications	Simulated training data; real OTA data for validation data	2k	SINR
Linear regression (LR) without regularization [129]	Wireless sensor networks	Simulation	Unknown	Classifier score; packet receive ratio (PRR); battery power savings

techniques usually require a lot of data to be able to perform well. For some of these works [54, 106, 125], however, the quantity of the datasets used is relatively small when compared to the recommendations from sample complexity study in [7].

2.7.3 Dataset Type

We also investigate the dataset types used for these tasks. Some works utilized OTA data collected from real hardware transmitters and receivers in the wild. Some other works also utilized simulated datasets generated from MATLAB, NS3 and other simulation engines. Other researchers utilized synthetic data which sometimes could be a combination of simulated and real devices. Some works [51, 107] also utilized simulation data in training and then tested with actual OTA data. Although there are numerous sound simulation models, the assumptions made by their creators do not always hold for OTA transmissions in the wild. This negatively impacts the acceptability of their results by other experts in the community. Recent work has shown that a combination of OTA and simulated data, as well as augmented data (i.e., synthetically modified OTA data), can reduce the required size of OTA training sets [193].

2.7.4 Performance Metrics

Performance metrics used include bit error rate (BER), symbol error rate (SER), SNR, SINR, EVM, signal-to-remaining-interference-noise ratio (SRINR), interference power, packet receive ratio (PRR), F1 score, PPMSE, mean square prediction error (MSPE) amongst others. Xu *et al.* [109] utilized a plot of the BER with optical SNR (OSNR) to evaluate their interference cancellation technique. A comparison was made between the conventional hard decision technique and a variation of deep neural networks algorithms. For BER, a lot of data would need to be processed before having a single error since practical BERs must be

very low (10^{-1} to 10^{-9}) and 10-100 errors are required for confidence. It is often used in computer simulations to evaluate the performance of digital communication systems [194]. Similar to BER, SER is usually plotted against some measure of signal-to-noise ratio (SNR). SER is the ratio of the number of symbols in error to the total number of symbols transmitted. While SNR is a metric that is used to parameterize the performance of a signal processing system that uses a Gaussian distributed noise. This is a very intuitive way to measure the performance of a digital communication system. For instance, Lin *et al.* [54] plotted the SER with respect to interference power in order to compare the performance of their denoising autoencoder. In understanding how deep learning can effectively suppress an interfering signal, [54] compared corrupted Wi-Fi signal before suppression and the signal after suppression by looking at a plot of the SNR versus the interference power. The resultant signal post suppression can be compared to a clean signal (interference-free) and the neural network output SNR can be very useful in understanding the impact of such a suppression approach. SINR defines the ratio of the power of a signal of interest to the sum of the interference power and the power of the background noise [195]. For radar systems application, the ratio of the power of a single object reflection to the average remaining interference power after cancellation is referred to as signal to remaining interference plus noise ratio (SRINR) [102]. Mun *et al.* [120] utilized this metric to compare the performance of the RNN-based automotive interference suppression technique.

Most of the deep-learning based interference suppression techniques discussed in Section 2.6 have an underlying assumption that training was done offline [54, 109] and thus discussions on training time complexity is not typically included. Unlike the computer vision, natural language and other fields where deep learning has been widely used, many works in the interference suppression literature scarcely discuss both time and computation complexities. Model complexity, on the other hand, is typically measured in terms of the number of param-

eters used for training. The larger the number parameters, the more complex the model is. For instance, Rock *et al.* [106] proposed six models with varying levels of complexity which in turn led to varying numbers of parameters. The key distinction between the complexity in models include kernel sizes (that is, filter sizes such as 1×1 , 3×3 , etc), number of layers, number of kernels per layer, number of pooling layers (used in CNNs), amongst other factors. The authors would like to note that the details of time and model complexities have not been provided in most of the works considered in this survey. In Table 2.4, we provide a summary of model complexity parameters for the works which provided this level of information. Since time and model complexities are very important implementation concepts in deep learning, the authors consider this as a reasonable future work where all the earlier mentioned factors are investigated in a thorough manner.

Table 2.4: Model Complexity Parameters from Selected DL-based Interference Suppression Techniques

DL rithm	Algo-	Input Size	Dataset	Model	Number of Layers	Kernel Size	Number of Parameters
CNN [106]		$1024 \times 128 \times 2$		Model A	4	2 kernels of size 3×3	160
				Model B	8	8 kernels of size 3×3	3,898
				Model C	4	16 kernels of size 3×3	5,298
				Model D	6	16 kernels of size 3×3	10,002
				Model E	8	16 kernels of size 3×3	14,706
				Model F	6	32 kernels of size 3×3	38,434
CNN [126]		2×128			9 (2 conv layers, 2 max pooling layers, 1 flat layer and 4 dense layers)	128 filters with size 2×8	324,330
						64 filters with size 1×16	
CNN [168]		154×2048			15 layers (12 convolution layers with 3 pooling layers)	5×5 filter	N/A
		1024×2048			21 layers (15 convolution layers with 6 pooling layers)		
CNN [192]		$10000 \times 2 \times 1024$			2 layers	First layer: 128 filters of size 1×3	N/A
						Second layer: 64 filters of size 1×1	

We provide some details on other common performance metrics used in deep learning literature which are not typically used as wireless communication metrics, yet may provide utility for evaluating interference suppression.

2.7.4.1 Area under the receiver operating characteristics curve (AUC)

The AUC metric is typically used to ascertain detection or classification performance for a multi-class classification problem at various threshold settings. The receiver operating curve (ROC) is a probability curve while the area under the curve (AUC) defines the degree or measure of separability [196]. In essence, it describes how well a model can distinguish between classes. It is a plot of the true positive rate (TPR) against the false positive rate (FPR). The higher the AUC, the better the model is at distinguishing between classes. Ristea *et al.*[105] used this metric to compare the performance of their suppression technique with other conventional suppression techniques. In their specific application, the closer the value was to 1 the better. When compared to non-deep learning technique such as zeroing, their deep fully convolutional neural network performed better for both the validation and test sets.

2.7.4.2 Mean Absolute Error (MAE)

MAE is another simple performance metric that measures the error between the values of an actual quantity and the predicted quantity. For instance, [105] computed the MAE in dB between the range profile amplitude of targets computed from label signals and the amplitude of target from predicted signals. Similar to SNR and AUC, [105] used MAE to compare between their technique and other suppression techniques. Hence, this metric can be applied at the application level.

2.7.4.3 Error Vector Magnitude (EVM)

The EVM is the root mean square (RMS) average amplitude of the error vector (*a two-dimensional vector in the I/Q plane that measures the distance between the corrupted and non-corrupted signal values*) that is normalized to the ideal signal amplitude reference [197].

Fuchs *et al.*[125] used the EVM to compare the detection in the original image with the predicted image. Rock *et al.*[106] also used the EVM to compare the performance of their scheme with the state-of-the-art. This metric is thus a metric that is applicable to the I/Q data.

2.7.4.4 Structural Similarity (SSIM) Index

The SSIM index is typically used for qualitative visual assessment. Research has shown that they perform better than mean squared error (MSE) [198]. According to [125], it compares image characteristics such as luminance, contrast and structure which are useful when removing interference since images may preserve target peaks resulting from interference. Hence, Fuchs *et al.*[125] utilized this technique to compare the spectrum free of interference with the reconstructed spectrum after interference removal. Hence, if the output can be viewed as a two dimensional array of I/Q or application output, it is applicable.

2.7.5 Benefits and Drawbacks of Deep Learning-based Interference suppression Schemes

In this section, we compare the benefits and the drawbacks of the deep learning-based interference suppression approaches. This is crucial because despite the gains brought about by applying deep learning to interference suppression, there are some drawbacks. This is also important as it helps the reader to understand what the deep learning algorithm solves as well as providing a better understanding of which algorithm may fit their need. Details of the benefits and drawbacks of each of the DL-based suppression techniques has been provided in tabular form in Table 2.5.

2.8 Open Challenges

Despite the successes with using deep learning algorithms in suppressing interference, there are still some challenges that need to be overcome before they can be adopted in the production environment. We note that some of these challenges also hold true for other deep learning-based wireless communication tasks.

2.8.1 Lack of Interpretability of Deep Learning Algorithms

One of the major challenges with using deep learning to classify interference is their lack of interpretability. Interpretability is a concept that deals with the model transparency and our ability to explain the intricacies of the model after the fact [85]. As expected, this is very important in suppressing interference sources since this can be useful in subsequently identifying spectrum offenders or devices misusing the spectrum in an opportunistic spectrum access paradigm such as can be found in the industrial, scientific, and medical (ISM) radio bands. This implies the need for interpretable artificial intelligence (AI). If the deep learning engine is viewed as a black box, it will be difficult for end users to trust the system. As researchers make progress in applying deep learning to interference suppression, privacy concerns may start to grow as we have seen in other deep learning applications [199, 200].

2.8.2 The Stochasticity of the Wireless Channel

Another issue with using deep learning in interference suppression is the issue of the randomness of the wireless channels. Most of the works reviewed in this survey have used data trained in only one type of environment (e.g. an-echoic, AWGN, etc). Meanwhile, in practical deployments, test signals go through a variation in channel environments which can result in the incapability of the trained model to adapt to new test cases. Since, there are numerous

channel environments at any given time, it is important to be able to interpret deep learning models so that they can better learn different RF environments. In other words, any difference between training and testing environment could impact the performance of deep learning algorithm. Developing effective GAN approaches for training data creation could be very valuable. Given that deep learning algorithms require a huge amount of data to train them, the nexus between the quality and quantity of data results in another challenge that impairs the use of deep learning in interference suppression. Training data collection is very laborious and can be more difficult when a supervised learning algorithm that requires labeling the data is considered.

Often times, researchers use data augmentation techniques to further increase the quantity of their data. For instance, in the image processing domain, one way to improve the quantity of training data is to augment the data using both simple and complex techniques. Simple transformation techniques include flipping, cropping or rotating the images to increase the quantity of such data. Since these techniques are not directly transferable to I/Q data used in interference suppression, complex techniques that leverage popular generative models such as generative adversarial networks (GAN) and variational autoencoders (VAE) have been proposed to augment the input data image in such a way as to generate new data that belong to the same distribution as the training data but not exactly in the training data. In most data augmentation for wireless communications tasks (including interference suppression, emitter and device classifications), such augmentation techniques have had very little success since the algorithm often times learn noise and other unwanted artefacts in the signal. While the stochasticity of the RF environment makes it difficult to augment datasets for wireless communication applications, Clark *et al.*[\[193\]](#) has shown that a combination of OTA and simulated data, as well as augmented data (i.e., synthetically modified OTA data), can reduce the required size of OTA training sets.

2.8.3 Need for Time and Model Complexity Studies

The need for a comprehensive discussion on time and model complexities for deep learning algorithms can not be over-emphasized. Usually, the more complex a deep learning model, the more computational resources and processing time it will require. Typically, the training and validation time processed in relation to the accuracy of a deep learning algorithm is described by the time-complexity analysis [201]. Furthermore, the time complexity of a model depends strongly on the computing capabilities of the underlying processor. For instance, graphical processing units (GPU) tend to train data faster than central processing units (CPU) because GPUs excel at processing complicated mathematical data such as those used in deep learning, thus, making them ideal deep learning tools. This is also because the advanced graphics cards are capable of duplicating the same job within a single card many hundreds or thousands of times which therefore results in increased productivity and time savings.

It is very crucial to understand the complexity of a deep learning model since it is essential for understanding the capability and limitation of the model. According to Hu *et al.* [202], deep learning model complexity is divided into expressive capacity and effective model complexity. The expressive capacity is typically a function of the over-parameterized architecture which typically implies that the actual or effective complexity is usually lower than the expressive capacity. Thus, the expressive capacity can be thought of as the upper bound on the quantity of knowledge the model can store while the effective model complexity deals with how much knowledge the model truly has when considering a certain training dataset.

However, in the deep-learning based interference suppression research, many works do not include these studies. Thus, there is a need in the literature for time and model complexity studies in order to enable researchers to understand the underpinnings of the deep learning

algorithms and build effective models based on their training datasets.

2.8.4 Lack of Performance Comparison between Online and Offline training

Other than Muranov *et al.* [114], many of the works discussed in this survey generally assumed that training was done offline. Thus, many of them did not include discussions on the trade-off between online and offline training. For offline training, the train and test datasets are fixed, and the model is developed based on these fixed datasets. On the other hand, online training typically involves sequential data where future data is predicted based on past data and the model parameters are updated on the go. Since most datasets generated in interference suppression are sequential, studies that investigate online learning may be useful in further improving the state-of-the-art which is typically based on offline learning. However, in circumstances where the full dataset is too large to fit in memory, online learning is a viable alternative. One deep learning architecture that combines both offline and online training is the recurrent neural network (RNN). RNNs are trained in an offline manner while they are used in an online way.

2.8.5 Performance Analysis

As we have shown in this survey, many of the researchers utilized different metrics to show the performance of their algorithms. Also, the majority of the DL-based interference suppression techniques utilized I/Q data as input into the algorithm. Compared to rigorous and well established mathematical foundation of signal processing methods developed by earlier pioneers of the wireless communications field (such as Claude Shannon's information theory mathematical derivations, amongst others), the lack of a robust mathematical underpinning for analyzing results from deep learning algorithms may hinder their practical adoption for

Table 2.5: Benefits and Drawbacks of DL-Based Interference Suppression Approaches

DL Algorithm	References	Technical Approach	Benefits	Drawbacks
Convolutional neural network (CNN)	[105, 106, 107, 108]	- Uncorrelated interference is learnt via FCN [105]. - STFT can be used to convert time domain signal into a spectrogram [105]. - CNN is used to denoising and thus suppress interference using both interfered data and clean data. - SIR & SNR can be used to scale interference and noise powers in relation to the object signal power [106].	- The CNN algorithms both outperformed the conventional suppression method such as zeroing, ramp filtering and IMAT techniques [106]. - Very useful for learning when the amount of data is relatively smaller. - Similar to Autoencoders, CNN can be used for denoising [106].	- Could result in very high EVM performance when compared to zeroing, ramp filtering and IMAT). - High EVM could also result in distortions for radar applications [106]. - Usually require high memory to train large datasets.
Deep neural network (DNN)	[109, 110, 111, 112, 113, 114, 115, 116, 117, 118]	- DNNs can be useful in cancelling interference for NoMA systems [109, 191]. - MLPs can be combined with filters to suppress interference [111]. - NoMA algorithms such as the SEFDM waveform utilize non-orthogonal sub-carriers which can result in ACI. [191]. - Performance depends on waveform characteristics - They are usually structurally simpler when compared to other deep learning techniques.	- Useful in mitigating non-linear interference and reducing complexity [110] - DNNs can suppress inter carrier interference (ICI) within certain signal waveform e.g SEFDM used in [109]. - Capable of learning a huge number of parameters in suppressing SI. - They can be used to reduce computation complexity for non-linear interference suppression. - Gains in SNR when compared to techniques that utilize hard decision detectors.	- High computational complexity - Simulation-based studies that utilize DNNs are very susceptible to impractical assumptions. - Could result in smaller MSE compared to conventional techniques [113]. - Varying complexity (such as number of hidden layers, number of parameters, etc) in the DNN model could result in significant performance degradation.
Recurrent neural network (RNN)	[120, 121, 122, 123, 124]	- The dynamics of sequences can be captured with RNNs using a network of nodes [185]. - Interference can be suppressed in time domain for radar systems. - Transmit signal is reconstructed in the presence of various interference signals. - They are typically used when input data is temporally correlated (e.g. time domain signals).	- Low processing time - Useful for applications where frequency changes linearly. - RNNs with attention blocks can capture sequential relationship between time steps. - Useful in reconstructing original signal. - RNNs can take in arbitrary input or output sequences.	- RNNs are vulnerable to exploding gradient problem. - Most of the implementations have only considered simulation data for interference mitigation. - For some applications, RNN-based methods have slow convergence rate and thus unable to track fast time varying interference [119].
Autoencoders (Denoising AE and CNN-based AE)	[51, 52, 53, 54, 125, 126, 127, 128]	- CNN-based AE is used to de-noise interfered range-Doppler (RD) images [125]. - Autoencoder is used to learn the distribution of the interference component so that it can be suppressed. - Signal is reconstructed without any knowledge about symbol structure and channel response. - AE is combined with an interference detection filter to suppress automotive radar interference in [51]	- Using real-world data measurement further improves the performance of AE methods. - DAEs perform well when the interfering power is larger than the amplitude of the desired signal. - Typically, AEs do not incur any signalling overhead since there's no synchronization between the transmitter, receiver and interferer. - Signal can be reconstructed without any knowledge about symbol structure and channel response.	- AE techniques often perform poorly at low SNR - AEs perform poorly with insufficient data - They also incur a huge processing time, hyper-parameter tuning and there is also a need to validate the model.

interference suppression. On the other hand, it has been shown that training and testing data do not usually belong to the same distribution [47], hence, information-theoretic techniques may still be explored to help with performance metrics that would be suitable for interference suppression research. Furthermore, metrics such as BER, while ideal for communications applications, are burdensome when compared to metrics like EVM due to the extreme amounts of data that must be processed for evaluation of low-BER regions.

2.8.6 Issues with Open Set Recognition (OSR) or Out-of-Distribution (OOD) Data

Open set recognition (OSR) is a common phenomenon in which unknown classes not present in the training phase during learning are found in the testing phase [203, 204]. Also known as out-of-distribution (OOD) data [47], this issue can result in a lack of trust for deep learning models since the test set in real world applications is infinite, and even more so in wireless communications. For example, we discussed earlier how the random nature of the wireless channel can result in a drastic degradation in the performance of a deep learning classifier. Typically, in deep learning-based algorithms, researchers make the assumption that the training and testing data are sampled from the same distribution. As a result of this assumption, the ability of the model to generalize well to unseen testing data is often accounted for. However, when the deep learning algorithm is deployed in real-world applications, testing data is not restricted to the same distribution as the training data. Hence, it is feasible and probable for the testing data to differ significantly from the training data.

Some prior works [45, 46] have shown that DL models usually make high confidence predictions for inputs that are either irrelevant or cannot be recognized. Although, this phenomenon was briefly mentioned in [204], we believe that it has not received enough attention

in the DL-based interference suppression research. For instance, in safety- and mission-critical applications, such as autonomous driving, OSR or OOD data must be effectively handled in order to avoid perception errors that can lead to system failures. Further, the ability to detect and consequently suppress interference using these deep learning techniques could be negatively impacted by OOD or OSR. Another reason why this issue may be consequential is that because they do not result in explicit model errors, they may be undetected. Thus, we believe that in order for deep learning to be safely deployed in interference suppression research, there is an important need to distinguish between the known and unknown classes in the testing data.

2.8.7 Challenges with Implementation

Training a deep learning algorithm is usually time-consuming and computation-intensive, thus most researchers train the algorithms offline and then deploy a trained algorithm in their application. For instance, Grimaldi *et al.* [175] observed that an online classifier, distinguishing between Bluetooth and Bluetooth low energy (BLE)¹⁶ technologies is ten times more likely to mis-classify when compared to offline methods which only mis-classified 2% of the time. This occurred because the two technologies have very similar PHY layer with both of them utilizing the Gaussian frequency-shift keying (GFSK)¹⁷ modulation. Furthermore, Schmidt *et al.* [3] ran their multi-layered CNN algorithm using expensive GPU servers which were used to train the algorithm. This expensive hardware could pose a barrier to the adoption of deep learning to practical interference suppression. While offline training works well in the computer vision research where noise and other factors do not disproportionately

¹⁶Bluetooth is a terrestrial, adhoc, short range communication protocol [205]. While BLE was designed for a reduced power consumption and cost with a similar communication range as Bluetooth[206], it is important to note that Bluetooth and BLE are independent of each other.

¹⁷GFSK filters the data pulses with a Gaussian filter instead of changing the frequency at the beginning of each symbol period so as to make the transitions smooth. GFSK is used for low power design requirements such as BLE [207].

impact performance, the same could not be said for interference suppression application. Since the interference characteristics of the received signal used to train an offline algorithm changes with time, researchers may find that their trained algorithm may not be suitable in different scenarios or channel environment or even at another time period. This would imply that retraining needs to be done during deployment. This retraining may happen more frequently than is done in other applications. For computation-constrained applications, this would quickly result in a bottleneck. One possible solution to this is to implement the deep learning algorithm directly on the deployed hardware so that training time can be more manageable in real-time or near-real-time systems. Again, for resource-constrained IoT devices, this may be infeasible.

2.9 Chapter Conclusion

In this survey, we have identified key characteristics of interference in several wireless communication applications. We have reviewed deep learning techniques that have been utilized by researchers to suppress interference. We also provided a comparison of such techniques for interference suppression. Our findings show that deep learning techniques have been shown to significantly improve interference suppression when compared to traditional suppression techniques. Also, deep learning techniques can learn complex non-linear relationship in the input data, thereby making them useful in learning complex non-linear decision boundaries. We also noted that deep learning techniques require access to a large dataset so that the algorithm can be sufficiently trained thereby improving test performance. It is important to emphasize that the application of deep learning to interference suppression will continue to grow. In conducting this survey, the authors found that deep learning techniques such as autoencoders and CNN are of particular interest, given their unique characteristics that can be leveraged in interference suppression. While there are a lot of merits to using deep

learning to suppress interference, we have also found some challenges that need to be addressed to ensure success. One key challenge with using deep learning is the availability of benchmark datasets. The CRAWDAD database is a potent tool that can serve as a reservoir of key datasets that can help in interference suppression. Lastly, we note that there is still a need to further investigate concepts such as explainable artificial intelligence (AI), computational complexity and online techniques so we can better understand and leverage the inner workings of deep learning models.

Chapter 3

Keep It Simple: CNN Model Complexity Studies for Wireless Classification Tasks

The complexity of deep learning models can be influenced by various factors, such as the number of layers, number of filters, size of the filters, and number of nodes in the hidden layer. Researchers in the field of deep learning often aim to improve the performance of their models by hyper-parameter tuning and other optimization techniques. In the literature, there has been a focus on improving classification performance through these methods. Although it is important to optimize deep learning models for performance, there has been limited attention in the wireless communication literature on thoroughly analyzing the factors that influence model complexity. This work aims to fill this gap by studying the relationship between model complexity, dataset size, and classification difficulty in a thorough and empirical manner. To the best of our knowledge, this is one of the first studies to examine this relationship. The work described in this section has been published by the author of this dissertation and can be found in [\[208\]](#).

Our contributions are as follows:

- We thoroughly analyze the complexity of three different CNN architectures (simple,

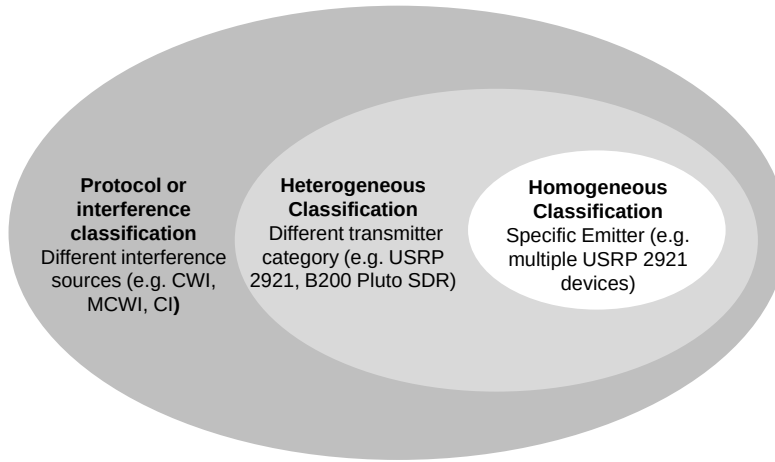


Figure 3.1: Overview of Interference Classification

medium, and complex) in relation to dataset size and classification difficulty.

- We show, empirically, that the performance of a simple CNN model with fewer parameters is comparable to that of a more complex CNN model. This is important because resource-constrained devices, which have limited processing power and storage capabilities, can benefit from using simpler CNN models.

3.1 Overview of Classification Tasks

In this work, our interference classification task is performed at levels of difficulty as shown in Fig. 3.1. At the interference or protocol level, a CNN can be used to classify different protocols or interference sources. At the heterogeneous level, a CNN can be used to classify different transmitter categories. Finally, at the homogeneous level, a CNN can be used to classify specific emitters (homogeneous) categories, such as transmitters from the same model or manufacturer. The outermost layer is the easiest classification while the innermost layer is the most difficult.

3.1.1 Transmitter Categorization

Most transmitters have features that are peculiar to each of them. For instance, when run at high power, the power amplifiers used in many wireless devices sometimes display non-linearities [209][183]. These non-linearities can be used to group the transmitters into different categories. We discuss the details of the transmitters used for our classification tasks.

3.1.1.1 Category A (USRP 2921)

These transmitters utilize more reliable linear power amplifiers and finer filters than other transmitter categories.

3.1.1.2 Category B (USRP B200)

In comparison to those in category A, the components in the transmitters in this category are less reliable. They were designed with low-cost experimentation in mind. They employ a single chain of the *AD9364*, which is frequently utilized to decrease hardware and software complexity.

3.1.1.3 Category C (Adalm Pluto SDR)

This category's transmitters are far less capable than the other two categories'. Considering their small size and low cost, the Pluto SDR is capable of a wide range of useful SDR applications.

3.1.2 Levels of Classification

In this section we discuss the three levels of classification experiments carried out for the model complexity study.

3.1.2.1 Protocol or Interference Classification

The protocol or interference level of classification is easier than the other two discussed in this section. This is because, there are enough distinguishing features at this level. For instance, in the radio frequency interference dataset used in this work, there is a marked difference between the three types of jammers described . For instance, the MCWI, which combines the SoI and a two-tone CW is structurally different from the CI [169].

3.1.2.2 Heterogeneous Classification

One transmitter from each of the three categories listed above in Section 4.1.4.3 is used in this study. While two of the three transmitters (USRP 2921 and USRP B200) were produced by the same company (National Instruments), all three are of distinct models. In level of difficulty, this classification is easier than homogeneous classification (discussed in Section 3.1.2.3 but harder than the protocol or interference classification discussed in Section 3.1.2.1

3.1.2.3 Homogeneous Classification

In this classification task, we seek to distinguish transmitters within categories A, B, or C. The transmitters are identical (same manufacturer and model), making this the most challenging classification problem of all three levels. This is because the classification algorithm must identify slight variations in transmitters which have the same architecture and hardware components. The same OFDM waveform is sent by all of the devices, significantly complicating categorization.

3.2 Implementation Details of the CNN

3.2.1 CNN Parameters for Model Complexity

The CNN algorithm utilized was a modified *Tensorflow* CNN model that was used to categorize handwritten digits from the MNIST dataset. We created three levels of complexity for our CNN models by varying the number of nodes in the hidden layer, the number of filters, and the size of the training dataset. These factors all contribute to the number of parameters in the model, and as a general rule, the complexity of a CNN algorithm increases with the number of parameters. As a CNN architecture becomes more complex, it is generally expected that the performance of the CNN algorithm will improve. However, one potential downside is that the algorithm may begin to overfit to the training data, resulting in a higher training accuracy compared to the test accuracy.

Our network contains three convolutional layers and max pooling layers and a fully connected layer. A 3×3 filter is applied to the input matrix by the convolutional layers. Convolution operations are carried out in the resulting sub-region to produce a single value in the respective output feature map. To integrate non-linearities into the model, the scale-invariant rectified linear unit (ReLU) activation function is applied to the feature map values.

The data collected by the convolutional layer is down-sampled using the pooling layer. We utilized the maximum pooling function of 2×2 . This indicates that the most important features of the signals are kept while others are deleted [210], which facilitates transmitter classification. Table 4.1 lists specifics of the parameters for all three designs.

3.2.2 CNN Architecture for Interference Classification

In this section, we describe the architecture of the CNN model and the training parameters used for the interference classification task. The CNN model is a pre-trained *ResNet18*

Table 3.1: CNN Parameters for Model Complexity

Parameters	Simple	Medium	Complex
Nodes in the hidden layer	0.26k	1.04k	8k
Total number of parameters for the whole network	6.1k	48k	276k
Filter size (No. of Filters)	3×3(16)	3×3(32)	
Batch size	16		
Number of layers	3		
Strides	2		
Number of classes	4 (for homogeneous devices) 3 (for heterogeneous devices)		
Learning rate	0.0002		
Max pooling size	2×2		
Dropout Probability	50%		
Input matrix size	38×100		
Activation function	ReLU		
Optimizer	Adam optimizer with cross entropy loss		
Loss	Cross entropy loss		
Training sizes	16,64,256,1024,4096,8192,16384		
Test sizes	2k		

model described in [211]. The architecture is summarized in Table 3.3. The model consists of many convolutional layers, two fully-connected layers, and one output layer, in that order. Leaky ReLU ($\text{Leaky ReLU}(x) = \max\{\alpha x, x\}$, where $\alpha \in (0, 1)$ is a preset parameter). All convolutional layers and fully linked layers are subjected to an activation function with $\alpha = 0.2$. The resulting (output) layer then has the softmax function applied to it. All convolutional layers are subject to batch normalization [180], however the output layer and fully-connected layers are excluded. Additionally, we use stride = 2 in the convolutional layers rather than 2×2 pooling layers for down-sampling. Such changes enhance performance and lower the variance of the results across various training epochs. $\mathbf{x}$ is normalized as $\mathbf{x}' = \mathbf{x}/x_{\max}$, where $x_{\max}$ is the largest input value allowed in $\mathbf{x}$. We utilize the Adam optimizer with the suggested default values in [212]. The learning rate is 5×10^{-4} , and the batch size is 64.

3.3 Experimental Setup

In this section, we describe the datasets used for the investigation of dataset size, model complexity and level of classification. It is worth noting that we only used publicly available

Table 3.2: Comparison of Training Accuracy and Testing Accuracy for all Model Complexity and Device Category

Homogeneous Classification (USRP 2921)							Homogeneous Classification (USRP B200)						
Simple CNN			Medium CNN		Complex CNN		Simple CNN			Medium CNN		Complex CNN	
Dataset Size	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc
16	1.00	0.407	1.00	0.406	1.00	0.408	1.00	0.502	1.00	0.492	1.00	0.544	1.00
64	0.996	0.545	1.00	0.563	1.00	0.568	0.988	0.567	1.00	0.608	1.00	0.591	1.00
256	1.00	0.660	1.00	0.688	1.00	0.694	0.996	0.623	1.00	0.658	1.00	0.652	1.00
1024	0.971	0.744	1.00	0.784	1.00	0.788	0.955	0.663	1.00	0.709	1.00	0.722	1.00
4096	0.947	0.807	0.99	0.836	1.00	0.829	0.900	0.806	0.995	0.821	1.00	0.826	1.00
8192	0.899	0.838	0.986	0.856	0.99	0.842	0.864	0.830	0.984	0.837	0.99	0.858	1.00
16384	0.88	0.860	0.975	0.862	0.99	0.876	0.856	0.843	0.971	0.860	0.97	0.870	1.00
Homogeneous Classification (Adalm Pluto SDR)							Heterogeneous Classification						
Simple CNN			Medium CNN		Complex CNN		Simple CNN			Medium CNN		Complex CNN	
Dataset Size	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc	Test Acc	Training Acc
16	0.859	0.369	1.00	0.437	1.00	0.407	0.96	0.762	1.00	0.842	1.00	0.816	1.00
64	0.988	0.512	1.00	0.548	1.00	0.450	1.00	0.836	1.00	0.844	1.00	0.841	1.00
256	0.939	0.641	1.00	0.633	1.00	0.579	0.997	0.883	1.00	0.885	1.00	0.895	1.00
1024	0.862	0.700	0.993	0.689	1.00	0.673	0.99	0.913	1.00	0.920	1.00	0.919	1.00
4096	0.820	0.718	0.959	0.699	0.99	0.706	0.97	0.937	0.99	0.954	1.00	0.952	1.00
8192	0.775	0.736	0.901	0.738	0.96	0.718	0.98	0.955	0.986	0.958	1.00	0.968	1.00
16384	0.772	0.747	0.865	0.753	0.85	0.754	0.98	0.97	0.975	0.970	1.00	0.972	1.00

datasets for interference classification, while we used our own testbed to generate datasets for the model complexity studies presented in this work. This is because there are some limitation with using public dataset, for instance, we were not able to control the types of transmitters used to generate the interference. Also, the channel used for transmission cannot be changed since we are using the dataset as is.

3.3.1 Dataset Generated For Model Complexity

The dataset generation process used for the assessment of model complexity is similar to the one used in [7]. The only difference is that more training data was added when compared to the work in [7]. We define the process of creating a baseband waveform, transmitting it over a channel, and receiving it as the transmitter-receiver chain. It is important to note that the details of the transmitter-receiver chain can be found in [7].

3.3.1.1 Baseband Waveform Generation

The hardware devices transmit OFDM packets created using GNU Radio Companion signal processing blocks. This is done by generating a stream of bits (0s and 1s) from a random source. We used an OFDM waveform that receives 10,000 data bits from a random source, which are mapped to the OFDM waveform using a QPSK modulation scheme with an FFT length of 512 and occupying the center 200 subcarriers with a cyclic prefix of 128.

3.3.1.2 RF Transmission

During transmission, GNU Radio and the transmitter hardware are connected through the USRP hardware driver (UHD) hardware support package. All USRP devices are managed and communicated with using a library called UHD. This is done using a GNU Radio block that takes as inputs the sampling rate, RF bandwidth, buffer size, center frequency, physical address of the device, and attenuation. The OFDM waveform is then up-converted to 2.45 GHz as the transmission center frequency and transmitted through the hardware's antenna. The sets of transmitters include four USRP 2921s, four USRP B200s, and four Adalm Pluto software defined radios (SDRs).

3.3.1.3 RF Reception

On the receiver side, the signal is received when the antenna and the transmitter are on the same channel. We assume that the receiver knows the center frequency and bandwidth of the transmitter and corrects for frequency offset at the receiver. The transmitted signal's center frequency and sample rate are stored in the UHD USRP Source block, which is used by a computer running GNU Radio Companion to down-convert the signal to baseband frequency. After being delivered to a low noise amplifier, the signal is separated into in-phase and quadrature components at baseband. It is then low-pass filtered and transferred

to an analog-to-digital converter (ADC). When the ADC process is completed, the digital samples are clocked into an FPGA. After being digitally down-converted using precision frequency tuning, a series of filters are used to decimate the FPGA image. The raw samples are then transmitted to a host computer through the host interface (using an Ethernet gigabit link, in this case) following decimation, made possible by the UHD. The complex samples are collected by the host computer using GNU Radio, and the IQ data is saved to a file and transferred to MATLAB for further processing.

3.3.1.4 Continuous Wavelet Transform (CWT) Signal Preprocessing

The features in the received signal that can be employed in the classification process are highlighted by the CWT. For CWT, a $M \times N$ matrix of coefficients is generated from N samples for a N -length signal, where M represents the number of scales [213]. The scales are determined automatically using the wavelength's energy distribution in frequency and time [213].

The CWT MATLAB function is used to pre-process the signals in order to enhance key transmitter properties in the received signal. After the signal has been transformed, the resultant 2D matrix of size $M \times N$ for each sample is stacked together and sent to the CNN classifier as a three dimensional matrix. The dimension of the output of the CWT process for this project is a 38×100 matrix of coefficient whereas the input to the CWT is 2×128 .

Table 3.3: CNN Parameters for Interference Classification

Parameters	RESNet
Batch size	64
Number of layers	18
Learning rate	0.0001
Maximum pooling dimension	2×2
Activation function	ReLU
Training/test size	80%/20%
Dropout Probability	80%

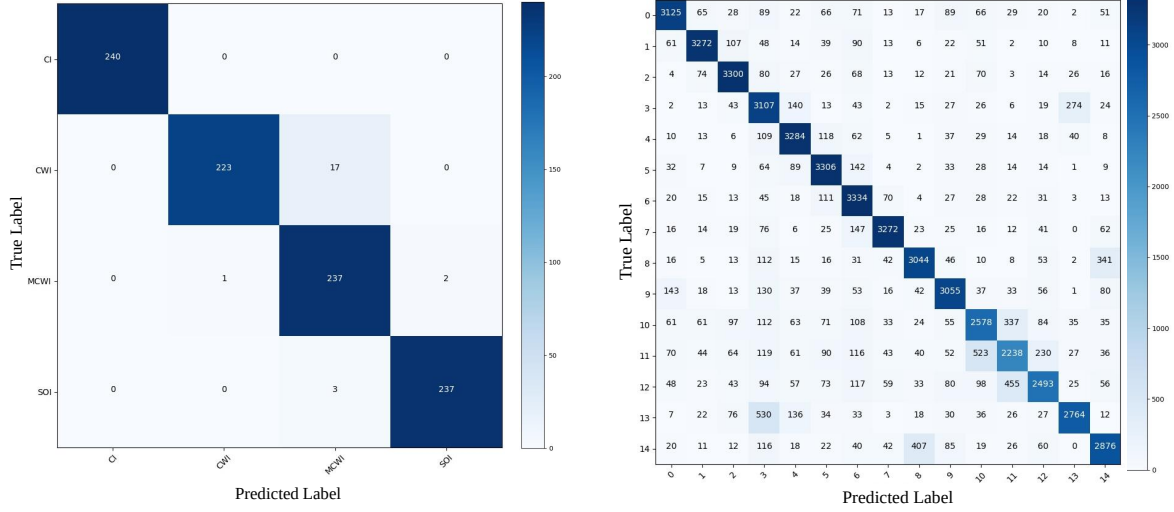
3.3.2 Datasets for Interference Classification

3.3.2.1 Radio Frequency Interference Dataset

In this work, we use publicly available wireless interference datasets to evaluate our approach. The RFI dataset used in this study was created by the authors of [169]. They created the dataset by combining a signal of interest (SOI) with three types of jammers (continuous-wave interference (CWI), multi-continuous-wave interference (MCWI), and chirp interference (CI)) at different signal-to-noise ratios (SNRs).

3.3.2.2 CRAWDAD Dataset

The second dataset used in this work was obtained from the Community Resource for Archiving Wireless Data at Dartmouth (CRAWDAD) website. This dataset, provided by Schmidt et al. [3], includes traces of IEEE 802.11b/g, IEEE 802.15.4, and Bluetooth packet transmissions with different SNRs in the baseband [172], as well as frequency offsets in the baseband [5]. There are a total of 15 distinct classes, 10 of which are IEEE 802.15.1 devices, 3 of which are IEEE 802.11 devices, and the remaining 2 are IEEE 802.15.4 devices.



(a) Confusion matrix for the RFI Dataset. (Classification Accuracy: 97.8%) (b) Confusion Matrix for the CRAWDAD Dataset (Classification Accuracy: 80%)

Figure 3.2: Confusion Matrix for CNN classification of the RFI and CRAWDAD Datasets.

3.4 Results

In this section we discuss the results from the model complexity and interference classification studies. Model complexity studies are important since they help to understand the relationship between classification performance and the number of parameters used by a deep learning algorithm (CNN used in this work). It is important to note that generating our own dataset to study model complexity helps us to vary different conditions in the data generating process which can not be done with publicly available datasets.

3.4.1 Model Complexity

This section examines the performance of the various CNN models across different device categories. When the test accuracy is significantly lower than the training accuracy, it suggests that the algorithm performs well on the training set but poorly on the test set, indicating overfitting. In order for a deep learning algorithm to perform well on new, unseen

data, it is important for the training and test accuracies to be similar. This indicates that the algorithm has learned to generalize well.

Table 3.2 compares the training accuracy to the test accuracy for all the classification tasks. At smaller dataset sizes, all the models tend to overfit significantly. However, as the dataset size increases, the algorithms tend to overcome overfitting issues. For most of the classification categories, the overfitting problem is significantly reduced, with the difference between the training and test accuracy being within 5%. Overfitting often occurs when the CNN is complex, with multiple layers and many nodes in the hidden layer. To prevent overfitting, regularization techniques such as dropout [214] and early stopping [215] can be used.

Figure 3.3 demonstrates that, for most transmitters, the simple CNN performs similarly to the medium and complex CNNs, despite having significantly fewer parameters. In fact, Figure 3.3d shows that the performance of the simple CNN is comparable to that of the complex CNNs for heterogeneous transmitters after a dataset size of 100. While more complex or sophisticated CNN models may offer some benefits, there must be a balance between the network’s generalization ability and its complexity. *These findings suggest that, it is important for the deep learning model to be both simple and robust, this is especially true for resource-constrained applications.*

3.4.2 Interference Classification

Accurate classification of interference sources is crucial for interference suppression or mitigation. Using a pre-trained ResNet18 model, we achieved a 97.8% accuracy on the RFI dataset, as shown in Table 3.3. Figure 3.2(a) shows the confusion matrix for the interference classes, indicating excellent performance on this comparatively easy classification task. For the CRAWDDAD dataset, the classification performance was about 80%, as shown in the confusion matrix in Figure 3.2b, after 25 epochs.

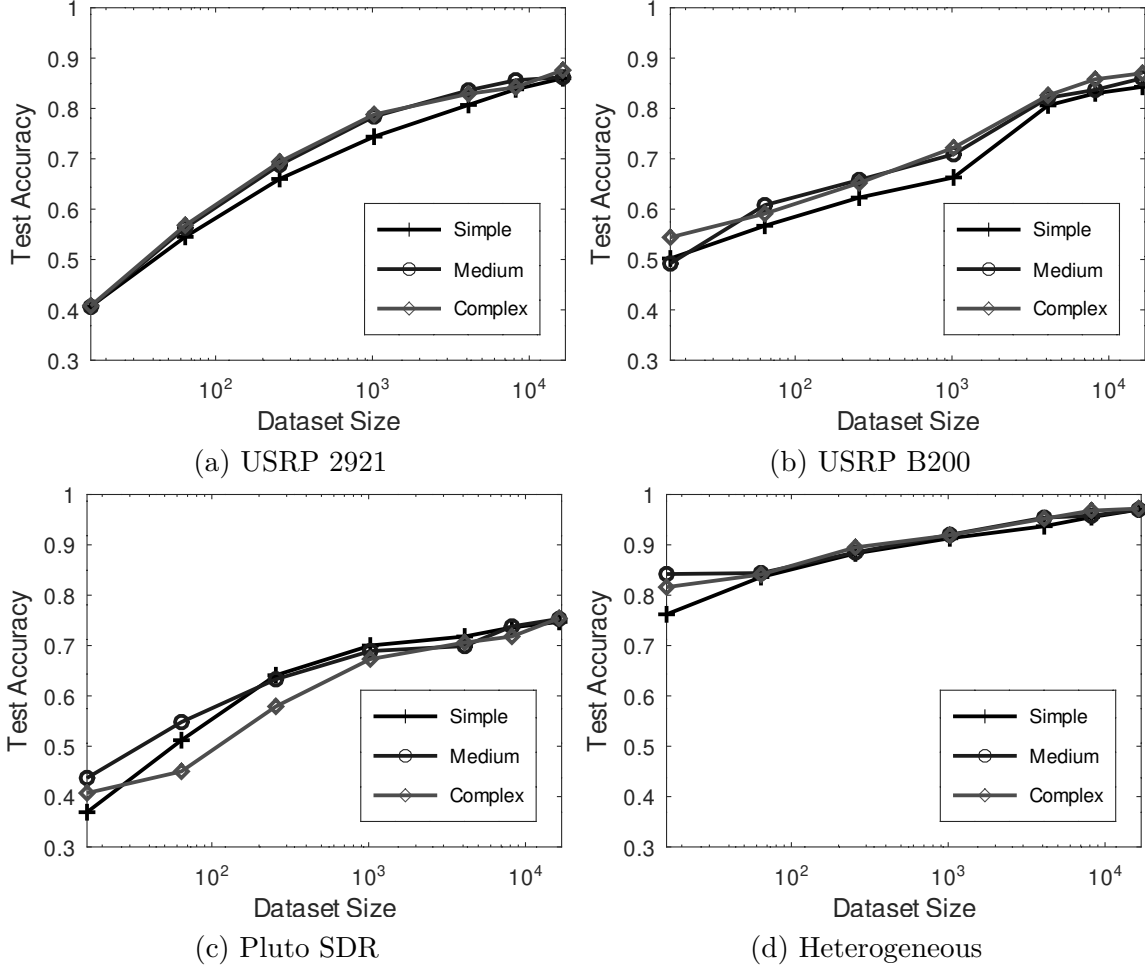


Figure 3.3: Comparison of Test Accuracy for Homogeneous and Heterogeneous Transmitter Classes

As discussed in Section 3.4.1, while more complex CNN models tend to perform better in classification tasks, they are typically prone to overfitting. Overfitting occurs when a deep learning model memorizes the training dataset, leading to high training accuracy but low test or validation accuracy. This can be seen in the results for the Pluto SDR in both Table 3.2 and Figure 3.3c. These results further emphasize the importance of our findings from the model complexity study discussed in Section 3.4.1. As previously mentioned, a less complex model can often perform as well as a more complex one without the risk of overfitting. Therefore, it is important to use models that are neither too complex nor too simple. The

right level of complexity can be determined by using just enough 2D convolution layers and filters to achieve good performance, starting with a simpler model and gradually increasing complexity as needed. This ensures that the deep learning model is neither underfitted nor overfitted.

3.5 Chapter Conclusion

In conclusion, this work has thoroughly examined the use of CNN for interference classification. Our results demonstrate that the CNN model is capable of accurately classifying different interference sources, as shown in the two datasets we used. Our study found that, while medium and complex CNN classifiers performed slightly better than the simple classifier, the difference in performance was not significant. This is an important finding since resource-constrained devices can easily leverage the simpler models. When designing deep learning models, the goal is to create models that can be applied to new data. Simple models are more likely to be able to do this because they are less prone to overfitting, which is a common issue with complex models. Different classification tasks, such as homogeneous and heterogeneous classification, may require different amounts of data and different levels of model complexity. In our study, the training dataset size, number of nodes in the hidden layer has had the greatest impact on CNN model performance. While the size of the dataset is important, we also note that the quality of the training dataset can also significantly impact the performance of CNN classifiers.

Chapter 4

Sample Complexity Studies in Wireless Communications

4.1 Estimating the Required Training Dataset Size for Transmitter Classification Using Deep Learning

In this Chapter, we generate a dataset for training and testing a CNN for transmitter classification, we developed a scheme for the prediction of classification performance and finally, we propose a rule-of-thumb for transmitter classification using CNN. The work described in this section has been published by the author of this dissertation and can be found in [7].

Our main contributions are summarized as follows:

- Using quantitative results, we show the relationship between the training dataset size and performance of transmitter classification using a CNN. Previous works have studied this relationship in non-deep neural networks. Our study focuses mainly on how this relationship holds for deep neural networks, specifically CNNs. We also propose a rule-of-thumb for minimal dataset size for acceptable performance of CNN-based transmitter classification.
- We propose a scheme for predicting the classification performance of a CNN based on

its dataset size via the *inverse law model* and the *Levenberg-Marqardt algorithm*. To verify the prediction accuracy, we used the chi squared goodness-of-fit test, and showed how our predicted values match the empirical results. We also compare our scheme with other schemes proposed in prior works.

The quantitative results presented in this work are relevant to only the specific configuration of the CNN that was used. However, we believe that our findings provide valuable insights on the relationship between the dataset size and classification performance when a CNN is used for transmitter classification and similar problems [32, 33, 35].

Section 4.1.1 provides some technical background to the components of our study. In section 4.1.2, we discuss the dataset generation process. Section 4.1.3 discusses the technicalities of the classification performance prediction scheme. Section 4.1.4 describes the experimental set up consisting of the transmitter component, receiver component and the transmitter categorization scheme. In section 4.1.5, the evaluation of the scheme is presented. We discuss related works in section 4.1.6.

4.1.1 Technical Background

4.1.1.1 Continuous Wavelet Transform

Deep learning algorithms can be used to solve the transmitter classification problem by using the intrinsic attributes of the transmitters, such as a power amplifier's non-linearity, an oscillator's imperfections, etc. To magnify these attributes for classification, we can pre-process the signal using efficient transforms.

Some popular transforms in the signal processing domain include discrete wavelet transform (DWT), short time Fourier transform (STFT), Fourier transforms (FT), continuous wavelet transform (CWT) etc. One deficiency of FT is that it has no mechanism for marking the

beginning and end of the intermittent sine-wave. This deficiency is corrected by STFT via analyzing smaller sections of the signal at a time. For DWT, a scanty depiction of the signal is provided. A pre-processing technique that has proven to work efficiently for IQ samples in transmitter identification is the CWT. [33].

CWT is a tool for time-frequency analysis of signals that enables a clear depiction of many frequency components of a signal as a function of time. CWT clearly separates the physical characteristics of transmitters in a 2D representation, thus improving the classification accuracy of the deep learning algorithm. The CWT is the more ideal candidate for signal or time series analysis due to its more fine grained resolution and is usually chosen in most tasks. In theory, the CWT compares a received signal with several shifted and scaled versions of the mother wavelet. Despite the overlap between the scales, the finer sampling scale usually leads to higher precision during the analysis of the received signal. In other words, the isolation of transients in signals and characterization of oscillatory properties becomes possible.

4.1.1.2 CNN for Transmitter Classification

CNN and other deep learning algorithms have inherent benefits associated with using them in communication or signal processing applications. We briefly discuss these benefits.

Benefits of using CNN in Spectrum Management

CNN extracts features or attributes from raw data by leveraging the hidden layers in neural networks. Unlike traditional machine learning applications, where the data representation is manually hard-coded as a set of features, the representation of the data is constructed automatically in CNN.

As a result of CNN's non-linearity property, it is able to learn complex non-linear associations between input data and their labels (or outputs) for accurate predictions. Several

works [3, 8, 31, 34, 216] including transient classification, estimation of transmitter I/Q imbalance, amongst others have utilized CNN for their tasks as it is capable of learning filters with varying complexity as the layers deepen. Another reason CNN is very useful for such applications is that the many layers of processing units are capable of extracting non-linear and abstract features of wireless signals that are constant for local spectral and temporal variations [34].

4.1.1.3 Curve fitting for Classification Accuracy Prediction

Our approach is similar to what is referred to as sample size determination (SSD) [36]. SSD is a way of selecting the number of observations to include in a statistical sample. For transmitter classification, the dataset size is dependent on so many factors such as difficulty of the classification task, complexity of classification algorithm, etc. The most common approach to predicting the dataset size needed for a target accuracy is the empirical or experiment-based approach [39]. The empirical approach fits experimentally generated classifier's learning curve to empirical models of dataset size versus accuracy.

There are three popular curve fitting models [217] that show how error, E , depends on the size of the training data, n :

4.1.1.3.1 Power Law The distribution of many physical, biological and many artificial phenomena follows a power law. It is given by the general equation: $E = bn^{-c}$, where E , the classification error is 1–accuracy and $n \in [1, \infty)$ is the dataset size. Accuracy is defined as how much the test error of the classification algorithm conforms to its training error. The variable, b , is the learning rate, and the variable, c , is the decay rate. Power laws are useful in modelling stochastic processes and non-linear relationships.

4.1.1.3.2 Inverse Square Root Model This model was motivated by *Bias-Variance decomposition* proposed by Geman et al. [218]. It is given by the equation: $E = a + bn^{-\frac{1}{2}}$. The variable a is a bias term which helps to compensate for model mis-specification. That is, the error approaches a as n tends to infinity. The relationship between the extended power law model and the inverse square root model is that the inverse square root assumes that the datasets are independent, thereby making $c = \frac{1}{2}$.

4.1.1.3.3 Extended Power Law Model This model given by: $E = a + bn^{-c}$. The variable, a , is the minimum achievable error. The extended power law has three parameters to be estimated and this estimation involves non-linear least-squares optimization which can be unstable or fail [217].

4.1.2 Dataset Generation

This section describes the steps in the generation of the dataset used to train and test the CNN algorithm employed. The dataset is generated using our testbed consisting of several hardware devices that serve as transmitters and receiver. We discuss the transmitter-receiver chain which includes: baseband waveform generation, transmission of the signal through the channel, and the signal reception. We also discuss CWT (the preprocessing technique used), and the CNN classification algorithm employed. Fig. 4.1 shows the equipment setup for the training dataset generation. The dataset generation steps described in this section have been used by the author of this dissertation in [7, 47] where the author collaborated with other colleagues.

4.1.2.1 Baseband Waveform Generation

We utilized the popular GNU Radio Companion signal processing blocks to generate OFDM packets that is transmitted through the hardware devices. A random source that generates

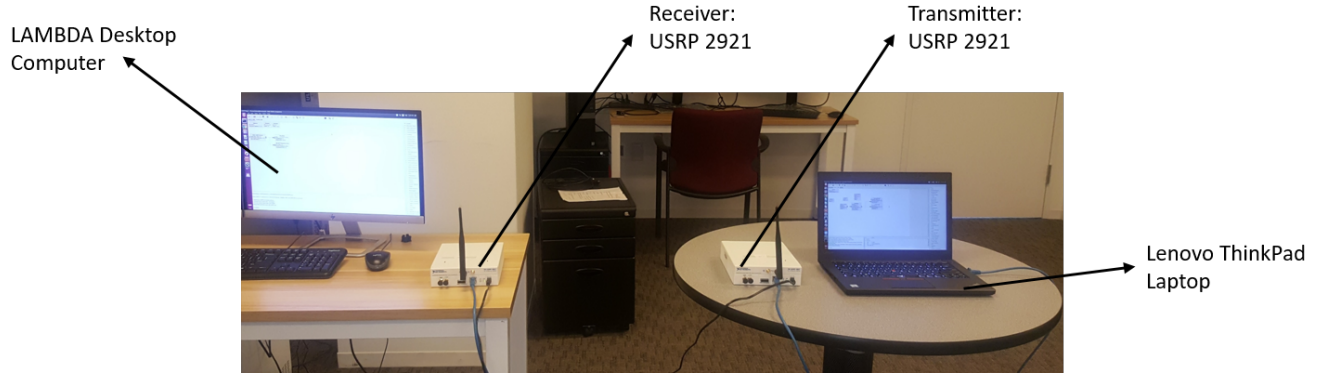


Figure 4.1: Equipment Setup for the Dataset Generation. (*Transmitter: USRP 2921; Receiver: USRP 2921*)

a stream of bits (i.e., 0s and 1s) is created. These bits are then mapped to an OFDM waveform using QPSK modulation with an FFT length of 512 and occupies 200 tones with a cyclic prefix of 128. For transmission, the OFDM waveform is then up-converted to a center frequency of 2.45 GHz.

4.1.2.2 RF Transmission

At the transmission stage, the hardware support package, USRP hardware driver (UHD), links GNU radio with the transmitter hardware. The UHD is a library used to control and communicate with all the USRP devices. It runs on a general purpose processor (GPP). This is achieved by a GNU radio block that takes the sample rate, RF bandwidth, buffer size, center frequency and attenuation as input. The physical address of the hardware device is also specified. The waveform is then transmitted over the air through the antennas on the hardware.

4.1.2.3 RF Reception

On the reception side, the receiver receives the signal through the antenna by latching onto the same channel as the transmitter. We assume that the receiver knows the center frequency and bandwidth of the transmitter. A computer running GNU Radio companion uses the

UHD USRP Source block, which holds the sample rate and the center frequency of the transmitted signal, to down-convert the signal to baseband frequency. The received signal is sent to a low noise amplifier after which the signal is split into the in-phase and quadrature component at baseband. The baseband signal is then low-pass filtered and then sent to the analog-to-digital converter (ADC). After the ADC is done, the digital samples are clocked into an FPGA. The FPGA image is then digitally down-converted via fine frequency tuning and then decimated using several filters. After the decimation is done, the raw samples are streamed to a host computer through the host interface (an Ethernet gigabit link, in this case). The UHD enables this communication between the USRP and the host computer. The host computer then collects the complex samples via GNU Radio. The IQ data is written to a file and sent for further processing using MATLAB.

4.1.2.4 Signal Data Preprocessing using Continuous Wavelet Transform

As discussed in section 4.1.1, CWT helps to magnify the features in the received signal that can be used in the classification process. For CWT, from N samples for an N -length signal, a $M \times N$ matrix of coefficients is constructed (where M is the number of scales) [213]. The wavelet transform scales are typically a power of 2 and are a function of the number of voices per octave, and the range from zero to the largest scale. The scales are automatically determined via the energy spread of the wavelength in both frequency and time [213].

After the received data has been collected, continuous wavelet transform (CWT) is employed to preprocess the signals in such a way to scale them up, thereby highlighting the transmitter attributes in the received signal. The file created in section 4.1.2.3 is sent to MATLAB where the *CWT* function in the wavelet toolbox is used to transform the signal. After the signal has been transformed, the resulting 2D matrix of dimension $M \times N$ for each samples is stacked together, resulting in a three dimensional matrix which is sent to the deep learning

classifier. The matrix generated by the CWT function can also be viewed as a magnitude scalogram, the absolute value of the CWT represented as a function of time and frequency [213]. Since we are interested only in the matrix, we refer interested readers to [213] for more information on the magnitude scalogram.

While certain differences can be seen among certain transmitters in the time domain, CWT does a better job of highlighting a transmitter's attributes due to the spatial correlations introduced by the transform. These correlations are harder to see when only time domain is considered.

4.1.2.5 CNN Architecture

In order to implement our deep learning algorithm, we started with a *Tensorflow* CNN model [219] and then modified it to fit our needs.

The three main components of the CNN architecture are the convolutional, pooling and fully connected layers. Our network has 2 convolutional layers, 2 max pooling layers and 1 fully connected layer. The convolutional layers apply a 5×5 filter to the input matrix. In the resultant sub-region, convolution operations are performed to create a single value in the corresponding output feature map. The rectified linear unit (ReLU) activation function, which is scale-invariant, is applied to the values in the feature map to introduce non-linearities into the model. When compared to the sigmoid activation function, the ReLU suffers less from the vanishing gradient problem and thus results in a better gradient propagation.

The pooling layer is used to down-sample the data extracted by the convolutional layer. This is useful for dimensionality reduction to reduce the processing time which also reduces the number of parameters, and the computations performed in the network. We used the 2×2 maximum pooling function (i.e., the algorithm keeps the maximum value in the sub-region of the feature map). This implies that the maximum value has important features that can

Table 4.1: CNN Parameters

Parameters	Value
Batch Size	200
Learning Rate (α)	1×10^{-4}
Maximum Pooling	2×2
Strides	2
Convolutional layer Filters	32
Kernel Size	5×5
Number of Classes	2, 3 & 4
Dropout Probability	80%
Training & Test Sizes	60% & 40%
Image height and width	71×1000
Number of Parameters	419328
Optimizer	Adam Optimizer
Activation Function	ReLU

be used for transmitter classification. Usually, the convolutional and pooling layers are both referred to as the convolutional layers. The parameters used for training and testing are listed in Table 4.1.

4.1.2.6 Classification Using CNN

The CNN algorithm described in the architecture above takes the CWT matrix as an input and automatically learns features in the data and then goes through the two convolutional layers of convolution and maximum pooling. In the fully connected layer, where all the nodes are connected to every other nodes in the previous pooling layer, the feature classification is done. The dense layer that comes after the last convolutional layer is responsible for grouping extracted features in the convolutional layer. A softmax activation function is used to compute the probability values for each node; it shows how likely each row vector falls into their corresponding target class.

4.1.3 Transmitter Classification Performance Prediction Scheme

We discuss the inverse law model selected, the parameter estimation algorithm utilized, the model fitting process and then we explain how the classification performance is predicted.

In the context of our study, transmitter classification performance prediction is the process whereby the performance of a larger dataset size is predicted using a power law model, parameter estimation algorithm and a model fitting technique. The parameters estimated are variables b and c explained in section 4.1.2. The CNN classification explained in section 4.1.2 provides data (i.e., dataset size and classification accuracy) used to estimate the parameters for fitting the model. Figure 4.2 is a high level diagram of four stages in the performance prediction stage. All four stages occur in a sequential manner and subsequent subsections explain them in more details.

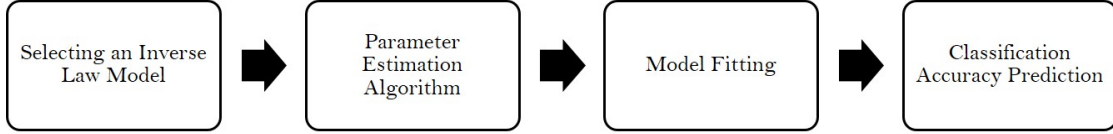


Figure 4.2: Overview of the Transmitter Classification Performance Prediction Scheme

4.1.3.1 Selecting an Inverse Law Model

Our empirical approach relies on findings from studies that demonstrated that classifier learning curves generally follows the power laws [42]. From the three models described in section 4.1.1.3, we chose the power law model (expressed by Equation 4.1) to fit our model. Before selecting this model, we tested all three models. The extended power law performed poorly when compared to the other two models. While the difference in performance between the power law model and inverse square root model is minimal at smaller dataset sizes, we observed that the predicted classification performance for the inverse square root model did not match the performance of the original experimental data. Performance, in this context, is defined as how close the predicted accuracy values are to the accuracy values from the actual experiments in section 4.1.2.6. After running multiple trials of the power law model and the inverse square root model, we found that the power law model achieved the best performance at both smaller and larger dataset sizes.

$$E = bn^{-c} \quad (4.1)$$

To fit the model, the initial results from our experiments were used to set up a system of equations from equation 4.1. From these systems of equations, we then use the parameter estimation algorithm, explained in section 4.1.3.2, to estimate the values of variables b and c .

4.1.3.2 Parameter Estimation Algorithm

After the learning curve has been generated as described in section 4.1.3.1, the parameters needed for fitting the model are estimated. These parameters would vary for different experiments since the experiments are designed to explore different classes of transmitters.

This step is crucial as this is where we employ our preferred parameter estimation algorithm. There are many popular parameter estimation algorithms in the literature. These algorithms solve for the needed parameters and they can also be used to fit a learning curve for the experimental data available. We discuss briefly the Levenberg-Marquardt algorithm used to estimate the parameters used in this work.

Levenberg-Marquardt Algorithm In many linear least squares problems, the Gauss-Newton (GN) method can be used to estimate parameters where a search direction is obtained at each iteration [220]. To ensure that the function decreases at every iteration, the direction of search is considered when a line search is carried out. However, when non-linearity sets in, the Gauss-Newton method encounters problems and hence a non-linear estimation algorithm becomes necessary. The Levenberg-Marquardt (LM) algorithm helps to overcome the problems encountered by the GN method.

A stopping criterion of 1×10^{-4} multiplied by a function tolerance is used to terminate the

Levenberg-Marquardt algorithm. The search direction used is computed as a cross between the GN direction and the steepest descent direction [220]. The algorithm helps in solving non-linear least squares problems. We utilized the MATLAB implementation of *lsqnonlin* function which has the Levenberg-Marquardt algorithm as an option. Given Equation 4.1, the LM algorithm estimates the parameters b and c . In order to estimate the parameters from the system of equations generated from Section 4.1.3.1, we initialized both the parameters, b and c , with the value of zero.

4.1.3.3 Model Fitting

In order to extrapolate or predict dataset sizes for certain performances, there is a need to empirically estimate some parameters that can be used for such predictions. To generate the learning curve, preliminary experiments with a smaller dataset size and their corresponding classification accuracy is conducted. This is a popular methodology in statistics for extrapolation or prediction. The classification accuracy and dataset size from the preliminary experiments in addition to power law models is then used to set up equations to estimate parameters b and c .

The learning curve is a plot showing the test accuracy versus dataset size. In this plot, there are three broad sections. The first section is characterized by a rapid increase in accuracy as the dataset size is increased. The second section shows how the increase in accuracy is less rapid as the dataset size increases. In the third section, the accuracy plateaus as the dataset size increases showing that there is little to no improvement in classification accuracy as the dataset size is increased.

In the medical bioinformatics research, researchers have explored inverse power laws in empirically generated learning curves to predict performance at larger dataset sizes. Mukherjee et al. [38] tested their method on relatively small data set ($n < 500$) and found out that

the differences between predicted and actual classification error was very small (between 1% – 7%).

To achieve our goal, we used an existing algorithm [36] to fit our power law model to preliminary learning curve points. This is done to predict the accuracy of a deep learning classifier at larger training dataset size for which data we may not have the resources to generate.

4.1.3.4 Classification Accuracy Prediction

In this step, we extrapolate the dataset size needed for higher classification accuracy. Since we are constrained by time and resources, the ability to predict the dataset size that we cannot experimentally ascertain becomes imperative. Therefore, interested readers can gain insights from our experiments on how much data would be required for their classification problems.

We employ equation 4.1, and the estimated parameters from section 4.1.3.2 to predict accuracy for larger dataset sizes. The estimated parameter is applied to dataset sizes that have not been considered before. A fitted curve will then be leveraged to extrapolate the performance of the deep learning classifier for larger dataset sizes.

With this step, accuracy for larger dataset sizes is predicted with some level of confidence. For example, Figueroa et al. [36] used an Hessian matrix and second-order derivatives to calculate a 95% confidence interval. Simply put, the fitted curve generated from the model fitting process is what we refer to as the predicted classification accuracy. More of this prediction is discussed in section 4.1.5.2.

4.1.4 Experimental Setup

We describe the setup of the transmitter and receiver used for the classification and prediction. All our experiments were conducted in a lab environment with varying distance between the transmitter and the receiver thereby allowing for multipath and other realistic channel conditions. In addition, we present the transmitter categorization approach utilized.

4.1.4.1 Transmitter Component

We utilized diverse types of transmitters in this section. The sets of transmitters include four USRP 2921, four USRP B200 and four Adalm Pluto SDR. We used the 2.45 GHz band for the radio transmissions, and the frequency offset was corrected. For all the transmitters, GNU radio was used to set up the transmitter block.

An OFDM waveform that received 10,000 samples from a random source with an FFT length of 512 was implemented. The cyclic prefix length of 128 creates the OFDM symbols using QPSK modulation. The generated OFDM signal was sent to the transmitter sink, which was then used to transmit the signal over the air through the antennas.

4.1.4.2 Receiver Component

The receiver only needs to know the center frequency and the bandwidth used by the transmitter. We utilized the National Instrument USRP 2921 to receive the signals over the air. The IQ sample was collected using a GNU Radio script that was running on the receiver. The USRP Source block received the IQ data from the transmitter and a GNU radio graphical sink (GUI Frequency Sink) is used to plot the received signal to confirm an accurate reception of the transmitted signal. Generally, the sink displays multiple signals in frequency by taking a set of floating point streams to plot the power spectral density (PSD) of the

signal. The IQ data is also written to a file that is pre-processed in MATLAB. Data was collected for the various distances between the transmitter and the receiver.

4.1.4.3 Transmitter Categorization based on Hardware Attributes

Many components of hardware devices exhibit some type of attributes unique to each device. For example, the power amplifiers used in many wireless devices tend to exhibit non-linearities when they are operated at high power [209][183]. These non-linearities can be used to group the transmitters into different categories.

There are three categories that cover the type of devices used in our experiments. The difference between these classes is found in the range of values for the third order intercept point (IIP3), frequency accuracy and sample rate. The IIP3 shows the effect of the power amplifier non-linearity. The frequency accuracy, which we associate to oscillator imperfection, is shown in the variation of the frequency offset from device to device. These features have been carefully chosen in the absence of a standard transmitter categorization scheme in the transmitter classification literature. Grouping transmitters that share similar hardware characteristics seem to generalize well in the experiments.

The experiments conducted in this section are designed for different categories under which we have grouped the transmitters used in this work. In table 4.2, we classify them into three broad categories based on three hardware features or attributes unique to each transmitter category.

4.1.4.3.1 Category A (USRP 2900 Series) The transmitters in this class are devices that use tighter filters and more reliable linear power amplifiers. Because of these features, distinction of transmitters in this class becomes more difficult. We have already seen this trend in our results. In other words, the performance is expected to be poorer when compared

Table 4.2: Transmitter Classification Method

Hardware Features	Transmitter Categorization		
	Category A (<i>USRP 2900 Series</i>)	Category B (<i>USRP B200 Series</i>)	Category C (<i>PlutoSDR</i>)
Frequency Accuracy (in ppm)	2.5	2.0	25
ADC/DAC Sample Rate (in Mega Samples/seconds)	400	61.44	20
Third Order Intercept, IIP3 (in dBm)	0 – 15	–20	–18

to other classes.

4.1.4.3.2 Category B (USRP B200 Series) The transmitters in this class are composed of lower quality components compared to class A. This can be seen in their hardware attributes. They are specifically designed for low-cost experimentation. They utilize a single chain of the AD 9364, used when software and hardware complexity is to be reduced. Transmitter classification is projected to be easier compared to devices in class A because the lower quality hardware attributes account for relatively easier classification.

4.1.4.3.3 Category C (Pluto SDR Series) The transmitters in this category are more constrained than those in classes A and B. Despite its miniature size and cost, the Pluto SDR is capable of many practical SDR applications. It is expected that the devices in this class can be more readily distinguished compared to those in classes A and B because their cheaper hardware component becomes very useful for classification purposes.

4.1.4.4 Types of Experiments

4.1.4.4.1 Experiment 1: Categorization of Transmitters in the Same Class In this experiment, devices in the same transmitter category, as identified in table 4.2 are considered. The same OFDM waveform is used for all the transmitters. This is the most difficult classification task because the transmitters are identical (similar manufacturer, model and make), and the classification algorithm has to rely on unique transmitter attributes, such

as non-linearity in their power amplifiers, oscillator imperfection resulting from the crystal oscillators that give rise to an unwanted frequency offset. As seen in table 4.2, the IIP3 is the effect caused by the non-linearity of the power amplifier. We measure oscillator imperfections by looking at the frequency accuracy specified for these devices by their manufacturers.

4.1.4.4.2 Experiment 2: Categorization of Transmitters in Different Classes In this experiment we use a total of three transmitters (i.e., one transmitter from each category in table 4.2). All three transmitters are of different models and make. In addition, they all transmit the same OFDM waveform as experiment 1. Compared to experiment 1, the classification task is easier.

4.1.5 Evaluation of Results

In this section, we discuss our results, conduct the goodness-of-fit test to verify our power law model, compare our prediction with prior works and propose a rule-of-thumb for transmitter classification.

4.1.5.1 Results

For experiments 1 and 2, we generated data for dataset sizes between 3,000 and 7,000. Dataset below 3,000 did not show any interesting results, hence, their omission. The plots in this section show the relationship between the classification accuracy and the training dataset size. Intuitively, the classification accuracy should improve as more data is added. Depending on the type of experiment, the number of dataset needed to achieve optimum accuracy varies. Figure 4.3a compares the classification accuracy for two transmitters across the three transmitter categories. The cyan, blue and black curves represent the classification accuracy obtained from our experiments while the red, green and magenta curves represent the classification accuracy from the prediction model explained in section 4.1.3.1. From

the figures in figure 4.3, the predicted models follows the experimental results with a very minimal deviation for most of the experiments. The plots are synonymous to a qualitative evaluation of the model as they show how the predicted model matches the experimental data. Our quantitative evaluation, described in section 4.1.5.3, is done using the chi square goodness-of-fit test to numerically examine how close the model is to predicting the classification performance for larger dataset sizes.

The classification done in experiment 2 in figure 4.4 has a steeper curve at lower dataset sizes because the features are easily distinguishable for smaller dataset sizes for heterogeneous transmitters. Distinguishing between two heterogeneous transmitters is a relatively easier classification task, as a result of this, we omit results for this category since they do not provide much insight into the effectiveness of our model. However, the results for three heterogeneous transmitters is shown.

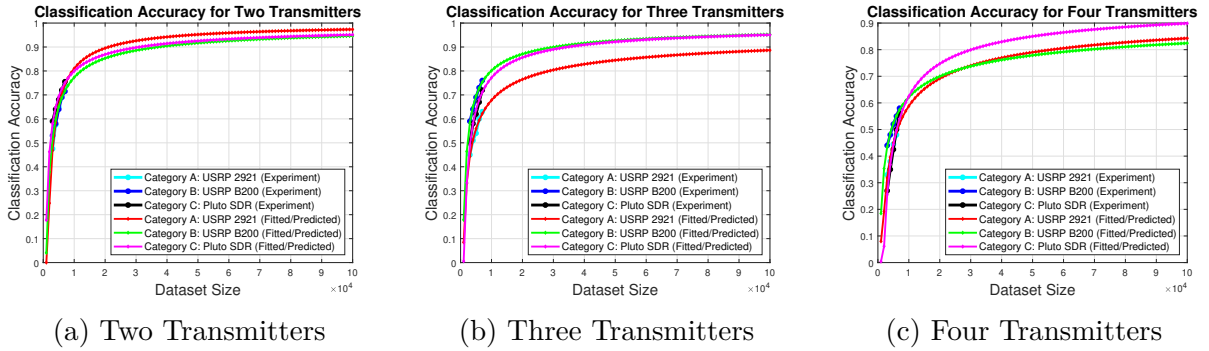


Figure 4.3: Comparison of Classification Accuracy Prediction with their corresponding experimental accuracy for Experiment 1 across all Three Transmitter Categories A, B & C

4.1.5.2 Discussion of Results

In figure 4.3a, we observe that the power law model and the estimation algorithm successfully fit our experimental results with minimal errors. However, the three experiments reach the saturation point at different dataset sizes where additional data points do not significantly improve performance. Take for instance, in figure 4.3b, the model fits the experimental

data, but more data is needed to achieve a better classification accuracy when compared to the two transmitter classification shown in figure 4.3a. In figure 4.3c, the four transmitter configuration requires a higher dataset size to achieve a classification accuracy of 90%. It is important to note that the transmitters classification for four transmitters (shown in figure 4.3c) is harder than that of two and three transmitters. This is not surprising as an increase in the number of transmitters makes the classification algorithm's job harder.

In figure 4.3b, the classification accuracy achieved for category A transmitters (*USRP 2921*) was lower compared to the other two categories B and C transmitters (*USRP B200* and *PlutoSDR* respectively). One reason for this could be that the *USRP 2921* transmitters are better equipped thereby diminishing the manifestation of unique transmitter attributes (e.g., power amplifier non-linearity) that helps in classifying them. For the *plutoSDR* transmitters, the device lends itself easily to leverage these transmitter attributes because they manifest such attributes more than the better equipped transmitters (*USRP B200* and *USRP 2921*). At lower dataset sizes (between 3000 and 20,000) for the predicted model, the difference between categories B and C transmitters is minute and have similar performance at higher dataset sizes as shown in figure 4.3b. In figure 4.3c, when the number of transmitters increased to 4, category C transmitters outperformed the other two categories. This further proves our point that category C transmitters easily manifest important attributes that helps to significantly improve their classification accuracy.

The results presented in this section is aimed at providing a baseline for determining dataset size for similar transmitter classification tasks. We believe that these results will help to bridge the gap in understanding dataset size requirement for using deep learning in spectrum management applications such as transmitter classification.

4.1.5.3 Goodness-of-fit Test for our Power law Model

We do not rely only on the visualization of the learning curves in section 4.1.5.2 which is a typical qualitative test of our fitted model, we also quantitatively test the fitted model to verify that it follows the experimental results. In order to quantitatively appraise our power law model, we measure how far the power law model deviates from similar measurements recorded from our empirical data. To measure the distance between the distribution of the empirical data, and the distribution of the power law model, we use a goodness-of-fit measure. In the interest of space, the goodness-of-fit test is performed for only experiment 1. Our hypothesis investigates the match between the performance of the predicted model and the performance of the actual experiment. The null hypothesis is that the predicted model matches the empirical data if a threshold is exceeded. The alternative hypothesis states otherwise.

A goodness-of-fit test generates a p -value that quantifies the tenability of the hypothesis. Essentially, the test is based on the calculation of the distance between the distribution of the empirical data, and the power law model hypothesized. If p is above a threshold, t (typically set to 0.95), the dissimilarity between the empirical data (from our experiment) and the predicted power law model can be credited to only statistical variations, hence, the null hypothesis is accepted. However, if the value of p is less than t , we can verify that the hypothesis is wrong (i.e., null hypothesis is rejected), and not a good fit to the empirical data. We utilized the χ^2 goodness of fit evaluation. The values in the χ^2 column is calculated by equation 4.2, where m_i represents the predicted accuracy and e_i represents the actual experimental accuracy.

$$\chi^2 = \frac{\sum_i^k (m_i - e_i)^2}{e_i} \quad (4.2)$$

Table 4.3: Goodness-of-fit Test for Experiment 1

Transmitter Category	No. of Transmitters	χ^2 value	p - value (p)
Category A	2	0.0406	$p \geq 0.995$
	3	0.0933	$p \geq 0.995$
	4	0.1921	$p \geq 0.995$
Category B	2	0.0383	$p \geq 0.995$
	3	0.5982	$0.975 \leq p \leq 0.995$
	4	0.319	$0.975 \leq p \leq 0.995$
Category C	2	0.1211	$p \geq 0.995$
	3	0.0387	$p \geq 0.995$
	4	0.023	$p \geq 0.995$

Table 4.3 shows the p -value and the χ^2 value for Experiment 1. The p -value is the level of marginal significance that represents the probability of occurrence of a given event. We see from the table that all the experiments we performed had p - values greater than our threshold of 0.95.

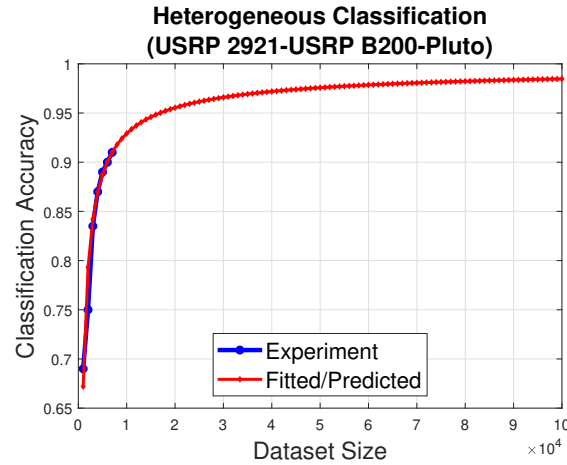


Figure 4.4: Classification Accuracy Prediction for Experiment 2

4.1.5.4 Comparison of Training Dataset Size Predictions with Prior Works and our Proposed Rule-of-Thumb

4.1.5.4.1 Comparison of Prediction As a proof of concept, we compare our prediction accuracy with similar prior works in transmitter classification. We hereby note that the goal

of many existing works in the literature varies from ours. For example, authors in [216] study the adaptation of CNNs to the problem of identifying coexisting wireless devices. Their main goal was to show that the classification performance of their CNN model outperformed other machine learning algorithm. In their experiment, they considered the homogeneous deployments of Wi-Fi 802.11n, Bluetooth and ZigBee device tagged: *Wi-Fi* (W), *ZigBee* (Z), and *Bluetooth* (B) and their corresponding heterogeneous deployments tagged *Wi-Fi-ZigBee* (WZ), *Wi-Fi-Bluetooth* (WB), *Bluetooth-ZigBee* (BZ) and *Wi-Fi-Bluetooth-ZigBee* (WBZ).

In [216], their transmissions were captured within an SNR range of 0 and 30 dB. They generated 15,400 samples from 4.3 million sweeps consisting of 78,400 features. While the result from the different transmitter categories in table 4.2 can be considered in this comparison, we note that our category C transmitters (*PlutoSDR*) has very similar properties with their Wi-Fi transmitter, and hence, enables a *somewhat* direct comparison. We compare the results from their confusion matrix which showed the prediction error per class. If we observe figure 4.3a, we see that when the dataset size is about 16,000, we achieve a similar performance (0.86) to their Wi-Fi to Wi-Fi classification (see Figure 3 of [216]) at 20dB. While it is difficult to directly compare our results with results from other works in the literature, we observe that on the average, the performance of Bitar et al.'s CNN model closely aligns with our predicted model. However, further comparison with other works in the literature is very difficult to do since the testbed setup and channel condition varies across board.

4.1.5.4.2 Proposed Rule-of-Thumb for CNN-Based Transmitter Classification

In the context of transmitter classification, we define rule-of-thumb as conventions or principles that researchers follow when estimating dataset size for classification performance. One of the popular rule-of-thumb is that the dataset size should be at least 50 to 1,000 times the number of classification classes [41] [221]. Some researchers [222] [223] also theorize that

the dataset size should be at least 10 times the number of weights in the network. Clearly, from our results in section 4.1.5.2, these two rules-of-thumb do not hold for transmitter classification in the spectrum management paradigm.

Based on the results for experiment 1 (shown in figures 4.3a-c), we propose a rule-of-thumb that the dataset size should be at least 10,000 to 30,000 times the number of the transmitters to be classified. This is because the transmitters classified in experiment 1 require a larger dataset size than those of experiment 2. Typically, in other domains (such as computer vision, medical bio-informatics, etc), for classification of four classes or less, the training dataset size is usually between 50 to 1000 times the number of classes. However, in the wireless transmitter classification domain, there is a need for more training data since data collection is more unconventional in a wireless environment. For instance, during live capture of IQ data, phenomena such as fading, shadowing, Doppler effects, etc, can significantly impair the quality of the dataset. This implies that the data generation process in radio frequency (RF) domain is more complex which makes the possibility of a representative dataset more difficult.

For the classification done in experiment 2, we propose that the rule-of-thumb be at least 1,000 to 5,000 times the number of the transmitters that needs to be classified. We see that the rule-of-thumb for the transmitters in experiments 1 and 2 varies significantly. For example, in experiment 2, for three transmitters of different classes, the dataset size needed is 6,000 (i.e., 2,000 times 3 transmitters). Following the rule-of-thumb of using between 1,000 and 5,000 times the number of transmitters, 2,000 which is well within this range, results in a classification accuracy of 0.89 for both the experiment and prediction as observed in figure 4.4.

The difference between the rule-of-thumb for experiments 1 and 2 can be attributed to the difficulty of the classification done. For Experiment 2 where the transmitters are hetero-

geneous, smaller dataset size is needed, resulting in a different rule-of-thumb with a lower factor.

4.1.6 Related Works

In domains such as computer vision and medical bio-informatics, there is a plethora of standard dataset, which researchers can utilize for various classification tasks. For example, the MNIST database consists of 60,000 training images and 10,000 testing images [154], an extended dataset called EMNIST [167] contains 240,000 training images and 40,000 testing images of handwritten digits and characters. Researchers have used CNN with the MNIST data to achieve a test accuracy of 99.77% [224]. This standard dataset is a result tedious investigation into the correlation between dataset size and performance.

In the medical bio-informatic field, Figueroa et al. [36] designed and implemented an algorithm that fit inverse power law to a learning curve generated from a smaller dataset. They also predicted their classifier's performance for larger dataset sizes. Alwosheel et al. [40] also empirically established a rule-of-thumb for ANN-based discrete choice analysis using real and synthetic data. They conducted extensive Monte Carlo analyses with varying level of complexity to establish their rules-of-thumb. Similarly, Choe et al. [41] conducted a study to determine the optimum size of training data needed to achieve high classification in the medical image classification systems. This type of investigation is yet to be investigated for transmitter classification tasks.

Merchant et al. [32] utilized deep learning to fingerprint IEEE 802.15.4 devices via developing a framework that trained a CNN algorithm with time-domain complex baseband error signal. In addition, Youssef et al. [33] investigated several machine learning algorithms with wavelets transforms to identify and classify wireless transmitters. None of these works attempted to investigate the correlation explored in this work.

The similarity between our work and authors of [36] and [38] is in the use of the learning curve to empirically fit our experimental data. This is a well known statistical approach to estimating sample size (or training dataset size). Our work departs from theirs in our dataset generation process, parameter estimation algorithm and application domain. In addition, we evaluate our work using the goodness-of-fit test with a high level of confidence. To the best of our knowledge, our work is the first to do this type of analysis on training dataset size requirement for CNN-based transmitter classification.

4.2 Chapter Conclusion

In this work, we have successfully utilized CNN to classify different categories of transmitters with varying difficulty level by utilizing continuous wavelet transform (CWT). We were also able to fit our experimental data to the power law model thereby predicting classification accuracy for larger dataset sizes for which we had no training data. In experiment 1 (which consists of transmitters in the same category), our prediction model is able to predict with at least 97.5% confidence in some experiments and more than 99.5% confidence in other experiments. Our study provides insights into dataset size requirements for the application of CNN to transmitters that have similar hardware features as the transmitters in table II. Intuitively, classification accuracy will improve as more data is provided to the deep learning algorithm, for different transmitter categories, however, rate of improvement varies. From our experiments, the transmitters in category C achieved a higher classification accuracy because they have features that are easily distinguishable in difficult classification scenarios.

Chapter 5

An End-to-End CNN-AE Interference Suppression System for Wireless Communications

5.1 Introduction

Interference suppression is of paramount importance in wireless communication systems to guarantee reliable and efficient communication, particularly in an increasingly congested and interconnected wireless environment. Interference in the context of wireless communication refers to the unintended presence of extraneous signals that disrupt the transmission and reception of the desired signal, impairing overall system performance. Interference, arising from a multitude of factors including coexisting wireless devices, environmental factors, and inherent channel limitations, can severely degrade the quality and reliability of transmitted signals. It leads to communication failures, reduced data rates, increased error rates, and compromises spectral efficiency, all of which undermine the fundamental goal of seamless and uninterrupted wireless communication [48]. By effectively suppressing interference, wireless systems can ensure robust signal reception, higher throughput, extended coverage, reduced latency, and enhanced overall network performance, contributing to improved user experience and enabling the successful deployment of critical applications such as 5G/6G,

IoT, autonomous vehicles, and mission-critical services. In essence, interference suppression serves as a linchpin in the quest for reliable and efficient wireless communication, enabling the realization of the full potential of modern wireless networks [49].

Interference-induced communication failures and degraded quality of service are prevalent and critical issues in numerous real-world scenarios. In urban environments, densely populated areas, and commercial centers, the proliferation of wireless devices and networks often results in co-channel interference [65], causing dropped calls, disrupted data transmissions, and overall deteriorated voice and data quality. In wireless LANs and Wi-Fi networks, neighboring access points and routers can interfere with each other, leading to slower internet speeds, interrupted streaming, and frustrating user experiences [225]. In industrial settings, wireless sensors and automation systems are susceptible to electromagnetic interference from machinery and equipment, risking data corruption, command misinterpretation, and potential safety hazards [226]. In wireless LANs and Wi-Fi networks, neighboring access points and routers can interfere with each other, leading to slower internet speeds, interrupted streaming, and frustrating user experiences [225]. In military and defense applications, interference can jeopardize secure communication and command execution, posing a threat to national security. In addition, high power transmitter spurs can also cause interference. Furthermore, in the context of emerging technologies like autonomous vehicles and remote healthcare, interference can compromise the reliability of critical data transmission, impeding the safe operation of vehicles and the timely delivery of medical information. In summary, interference-induced communication failures and degraded service quality have profound implications across a broad spectrum of real-world scenarios, underscoring the urgency of effective interference mitigation strategies.

Interference can manifest in several ways. Co-channel interference occurs when multiple devices or signals share the same frequency band or channel, causing overlapping transmissions

that interfere with each other [65]. Adjacent-channel interference arises when signals spill over into adjacent frequency bands due to limited spectral isolation, affecting neighboring channels [64].

5.1.1 Interference Suppression in Wireless Communications

Traditional, expert designed, methods have historically been employed to address interference in wireless communication systems. These methods, such as filtering, adaptive algorithms and conventional signal processing approaches typically rely on predefined models, statistical assumptions and heuristic-based approaches to detect and mitigate interference [9, 10, 12]. Such techniques include filtering and other conventional signal processing approaches. While traditional techniques have been effective to some extent, they face significant limitations when dealing with complex interference patterns. These limitations encompass difficulties in adaptation to diverse and dynamic interference sources. Non-deep learning methods may face challenges in effectively modeling the diverse and unpredictable nature of interference sources, thereby impeding robust interference suppression. Furthermore, while non-linear interference rejection methods like successive interference cancellation exhibit efficacy, they suffer from specificity and limited generalizability.

In contrast, deep learning techniques, such as CNNs, recurrent neural networks (RNNs), autoencoders, amongst others, offer the advantage of learning data-driven representations from raw signals, allowing them to adapt to complex interference scenarios, extract high-level features, and generalize well across diverse interference types. This adaptability and ability to perform nonlinear processing make deep learning approaches increasingly attractive for addressing interference in wireless communication, particularly in the face of evolving and dynamic interference challenges. Much of the works that studied the application of deep learning to wireless communications have focused mainly on the classification aspect. In many of

the prior work, the primary focus was on signal classification and recognition, thereby emphasizing interference classification techniques rather than interference suppression techniques. In our earlier work [23], we surveyed many DL-based interference suppression techniques such as CNN, RNN, deep neural networks as well as autoencoders. While autoencoders have been explored in the context of interference suppression for automotive radar [51] and in jointly optimizing transmitter and receiver systems to combat interference [52, 53], we found a lack of prior work evaluating autoencoders for receive-side interference suppression for digital communication signals. In a related work, authors in [54] explored the use of denoising autoencoders for specific interference sources and structures within the Wi-Fi spectrum. In this work, we examine the use of autoencoders to understand the latent space representation of the input data with interference so as to suppress the interference at the decoder. This approach offers a holistic and end-to-end solution for interference suppression, where the entire process, from encoding the data to its reconstruction, is seamlessly integrated into a single neural network framework compared to traditional suppression methodologies and also deep learning approaches in the literature.

5.1.2 Study Objectives

Our study provides several insights. It eliminates the need for manual interference estimation or preprocessing steps, streamlining the signal processing pipeline. Additionally, it allows the autoencoder to adapt and generalize to varying interference patterns, enhancing its robustness in real-world scenarios where interference characteristics may change unpredictably. In addition, our approach eliminates the necessity for explicit interference detection mechanisms, enabling the system to autonomously discern and mitigate the undesired signal disturbances. Our model's adeptness in autonomously suppressing interference signifies a significant advancement, showcasing its potential to streamline and simplify the process of

managing interference within wireless communication systems.

Compared to the prior art, this study leverages rigorous empirical analysis to understand how autoencoders can enhance signal quality by suppressing interference, and ultimately improve the reliability and efficiency of wireless communication. We thus provide empirical evidence of their utility, adaptability, and robustness across various interference scenarios, thereby underscoring the pivotal role of autoencoders in revolutionizing interference management strategies, offering a transformative solution for mitigating the adverse impact of interference in modern wireless networks.

5.1.3 Contributions

Our contributions are summarized as follows:

- We investigate how an autoencoder represents a digital phase-amplitude modulated signal affected by tone and wide-band interference. Emphasizing the latent space formation during encoding, the autoencoder expands these signals into a discernible and high-dimensional space.
- We leverage the learned latent space representation to discern and suppress the interfering tones present within the signal of interest. Through its encoding and decoding processes, the autoencoder effectively disentangles the signal components, enabling the removal of the interference from the original signal. The system efficiently reconstructs the cleaned signal from the latent space representation, thus demonstrating its ability to substantially attenuate or eliminate the interference.
- We investigate the removal of explicit interference detection mechanisms, enabling the system to autonomously discern and mitigate the undesired signal disturbances. The autoencoder's capability to suppress interference in a seamless, self-contained fashion

emphasizes its potential for streamlining and simplifying interference handling in wireless communication systems. In addition, we explore generalization to interference characteristics such as center frequency, amplitude, amongst others.

- Using QPSK as the exemplary SOI, we evaluate the performance of the autoencoder using error vector magnitude (EVM), signal to interference plus noise ratio (SINR) and bit error rate (BER) to quantify the fidelity and efficiency of the autoencoder-driven interference suppression in comparison to traditional techniques, thereby aiding in validating the performance of the CNN-AE.
- We compare the CNN-AE method with two traditional methods. Our results show that CNN-AE outperforms matched filtering and adaptive filtering (least squares filtering, in this case), leveraging its ability to learn intricate signal representations, thus enabling robust noise and interference removal while preserving signal integrity. In addition, our study shows the efficacy of CNN-AE as an advanced and versatile tool for significantly improving the quality, reliability, and robustness of digital phase-amplitude modulated signals in noisy and interference-prone wireless channel environments.

5.1.4 Chapter Overview

In Section 5.2, we discuss other interference suppression related techniques. Section 5.3 discusses the system model which includes a description of the wireless communication system being investigated, the signal of interest and interfering signal models and some assumptions and constraints of the system model. In Section 5.4, we discuss the CNN-AE architecture which describes the training dataset generation, the training and evaluation of the data, machine learning optimization techniques and the evaluation metrics for measuring the autoencoder performance. Section 5.5 shows the performance evaluation and a comprehensive discussion on our observation. Finally, section 5.6 concludes the work.

5.2 Related Work

5.2.1 Interference suppression techniques

Traditional interference suppression techniques, including PHY layer schemes and blind source separation (BSS), rely on predefined interference categories to guide their approach. PHY layer schemes optimize the physical layer design or introduce redundancy to mitigate interference at the receiver [9, 12]. These methods use various techniques such as heuristic models, optimization algorithms, and adaptive filtering. For example, Carrick et al. [12] explored the use of time-varying frequency shift filters (TV-FRESH) to suppress linear frequency modulated radar interference in orthogonal frequency division multiplexing (OFDM) signals. Adaptive filters, such as Least Mean Squares (LMS), adjust their filtering criteria based on interference type [15]. However, PHY layer techniques can be limited when dealing with time-varying interference, leading to reduced effectiveness. Other approaches rely on signal properties like cyclostationary features [12, 17], signal envelope [19], and channel bandwidth [20] to identify and suppress interference. However, these techniques may lack scalability in modern wireless environments with diverse and numerous devices present [23].

In traditional interference suppression methods, precise identification of interference attributes—such as location, duration, and specifics—is a key challenge. These methods face three main constraints: reliance on specialized domain knowledge, limited adaptability to varied interference scenarios beyond set parameters, and complexity arising from precise interference detection. These limitations highlight the need for more adaptable and data-driven approaches, like deep learning techniques, in modern wireless communication for effective interference suppression. Deep learning-based interference suppression surpasses limitations of traditional methods by not requiring real-time interference modeling, enabling practical

deployment. In stochastic RF environments, deep learning’s ability to learn directly from data without prior knowledge addresses the challenge of interference suppression.

The authors of [55, 56], surveyed deep learning techniques for radar, wireless personal area network (WPAN) and wireless local area network (WLAN) applications respectively. In our earlier work [23], we addressed the gap in deep learning-based interference suppression by reviewing various techniques applied across the breadth of wireless signal applications, emphasizing learning from data rather than expert systems. Further, we provided an in-depth technical discussion, comparisons, and guidelines for implementing these algorithms, while also identifying challenges and suggesting future research directions in this domain.

5.2.2 Autoencoder-based approaches for signal processing and interference removal

Autoencoders, versatile within diverse domains, excel in denoising applications by filtering noise to enhance data quality. Additionally, they aid in dimensionality reduction, compressing data efficiently and enabling visualization. Autoencoders’ success in artifact removal underscores their relevance in wireless communication. In the field of medical imaging, studies such as by Chuang et al. [227] have employed autoencoders to remove pervasive EEG artifacts and reconstructing brain signals, ensuring the accuracy of diagnostic images. In the realm of natural language processing, autoencoders have been applied to denoise and reconstruct text data, as seen in the work of Vincent et al. [228], aiding in tasks like sentiment analysis and text generation. Furthermore, in image processing and computer vision, autoencoders have been utilized to eliminate artifacts and enhance the quality of images, as demonstrated in [229] for image dehazing and [230] for medical image denoising. These diverse applications showcase the versatility of autoencoders in artifact removal, underscoring their potential applicability in mitigating interference and improving signal quality within

wireless communication systems.

Lin et. al. [54] proposed a method that used denoising autoencoders and generative adversarial networks (GANs) to learn patterns of interference with *unknown* structures to suppress heterogeneous technology interference. While the use of GANs to generate synthetic training samples helped to overcome data collection challenges, it introduces potential limitations. The effectiveness of the model may depend on the quality and representativeness of the synthetic data, which requires rigorous validation. Chen *et al.* [51] investigated an approach that combined an autoencoder network with an interference detection filter to suppress automotive radar interference. Chen *et al.* found that using a gated convolution, the encoder network can learn the pattern of the residual signal free of interference. In addition, their approach uses a traditional interference detection filter that is first used to detect the interference before suppression using the gated convolution. This is fundamentally different from our approach where the interference is detected automatically and then subsequently suppressed. Furthermore, their work did not investigate the generalization capabilities of their proposed model. Another distinction from our study is that their work did not investigate the impact of different types of interference and their location in the spectrum.

Autoencoder-based methods applied to channel modeling have found utility in jointly optimizing transmitter and receiver systems to combat interference, as evident in prior studies [52, 53]. For instance, Erpek et al. [53] designed a modulation scheme using the autoencoder (encoder on the TX side, decoder on the RX side). While their work also utilizes autoencoders for interference handling in wireless systems, our approach does not integrate the autoencoder into the modulation design but rather utilize the autoencoder on the receiver side while using a fixed modulation scheme.

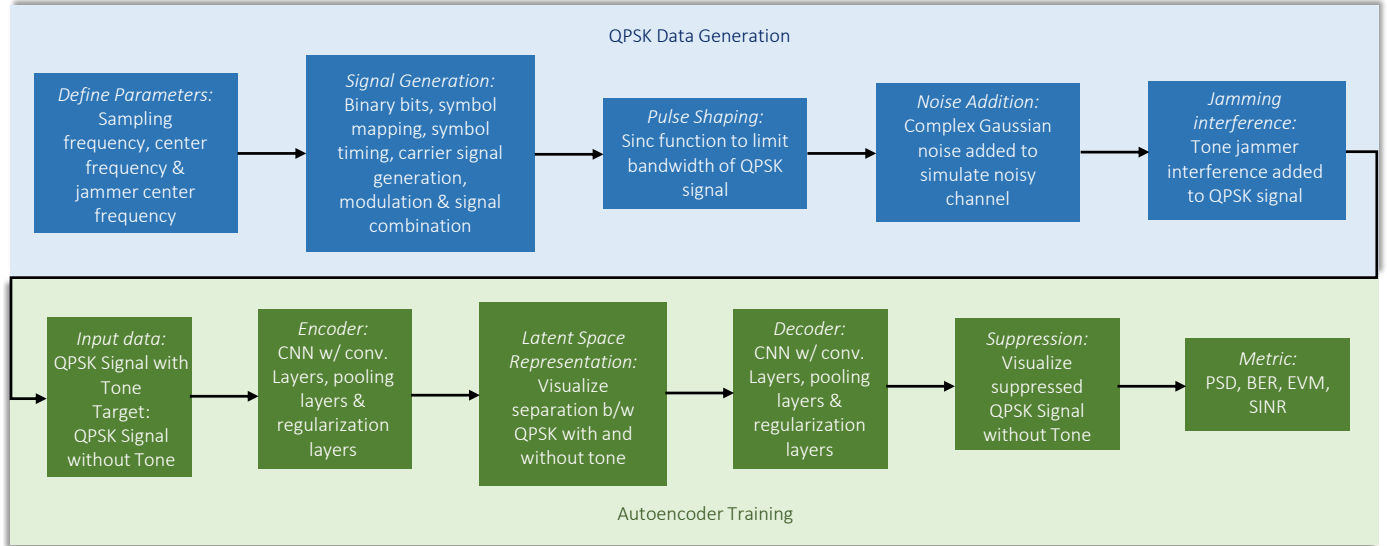


Figure 5.1: Overview of the System Model

5.3 System Model

5.3.1 Description of the wireless communication system

5.3.2 Description of the wireless communication system

We model all signals at complex baseband with two orthogonal carriers, denoted as the in-phase (I) and quadrature (Q) components. The digital modulation maps binary data bits onto complex symbols. Nonlinearities in the communication system, such as amplifiers or mixers, can cause intermodulation products. The out-of-band tone may mix with the digital modulation signal, generating unwanted spectral components within the digital modulation signal's bandwidth. At the receiver, the presence of the interference can introduce distortion and degrade the digital modulation signal's quality. Interference from the tone can result in increased bit error rates (BER), reduced signal-to-noise ratio (SNR), and impaired demodulation performance.

5.3.3 Signal of interest (SOI) and interfering signal models

This system model encompasses the signal generation process and the interference signals. The general system model is described by $r[i] = s[i] + x[i] + n[i]$ where $r[i]$ represents the received signal, $s[i]$ represents the signal of interest, $x[i]$ represents the interference and $n[i]$ represents noise. Note that T_s is the sampling interval and i represents the discrete-time index. In general, our desired signal, $s[i]$, is a QPSK signal and various forms of interference along with noise will be used to test our deep learning interference approach to interference suppression. Noise power, often denoted as σ^2 , represents the average power of the noise present in a signal or a system. The details of the parameters such as sampling frequency, SOI center frequency tone center frequency amongst others are discussed in Section 5.4.

5.3.3.1 Signal Generation

We construct a system model for generating a Quadrature Phase Shift Keying (QPSK) signal while introducing a tone interference. The tone interference was introduced for in-band and out-of-band interference. The model generates random binary bits (0s and 1s) representing the QPSK signal. The binary bits are mapped to complex symbols. Each pair of bits (I and Q) corresponds to one of four possible complex symbols the QPSK constellation $[(+1, +1j), (-1, +1j), (+1, -1j), \text{ and } (-1, -1j)]$. A time vector is created to represent the time axis, and the carrier signal is generated at the specified center frequency. The carrier signal is complex, as it has both in-phase and quadrature components, as given by:

$$s[i] = \sqrt{P_S} \sum_k s_k \cdot \text{sinc}(iT_s - kT) e^{j2\pi F_c(iT_s)} \quad (5.1)$$

where s_k represent the k -th QPSK symbol, P_S denotes the signal power, s_k are the symbols of the signal, $\text{sinc}(iT_s - k)$ is the sinc function in discrete time defined as $\text{sinc}(iT_s - k) =$

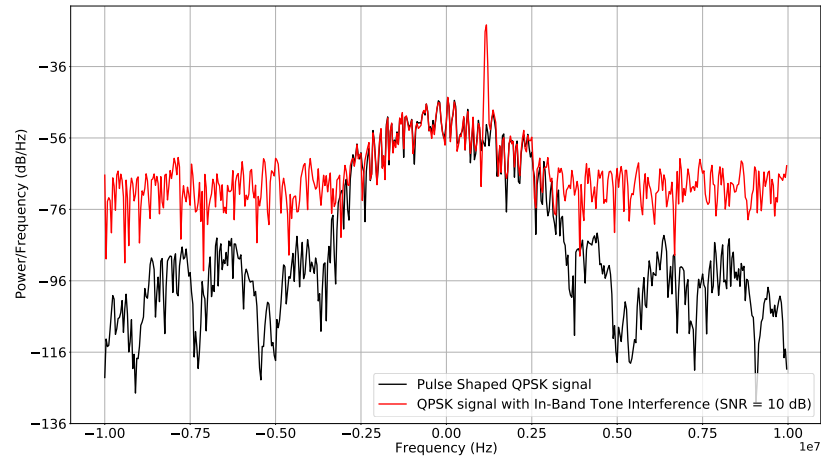
$\frac{\sin(\pi(iT_s-k))}{\pi(iT_s-kT)}$, T is the symbol duration, F_c is the center frequency and i represents the discrete-time index. The relationship between symbol duration (T) and signal bandwidth (B) is inversely proportional, as defined by the formula: $B = \frac{1}{2T}$, where B represents the bandwidth and T denotes the symbol duration. The sinc function is designed to shape the QPSK signal and band limit it. The pulse duration is determined by the symbol rate and is calculated to ensure proper pulse shaping and that it satisfies Nyquist criterion. The pulse is convolved with the QPSK signal to apply pulse shaping, preparing the signal for transmission.

5.3.3.2 Tone Interference

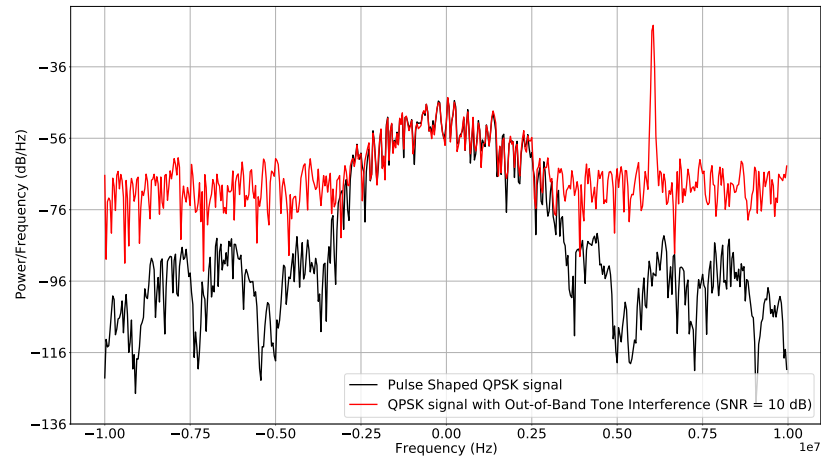
A tone interferer is created by generating a complex tone signal at the specified interferer center frequency. The tone interferer is added to the QPSK signal, introducing interference. The tone signal $x_{tone}[i]$ at center frequency F_{tone} is given as follows where P_X represents the power of the tone signal:

$$x_{tone}[i] = \sqrt{P_X} \cdot e^{j2\pi F_{tone}iT_s} \quad (5.2)$$

The tone interference introduces a frequency offset relative to the center frequency of the QPSK signal. This means that the interfering tone signal operates at a different frequency, often in a frequency band adjacent to or overlapping with the QPSK signal's bandwidth. The extent of this frequency offset plays a critical role in determining the interference's impact. The tone interference can cause phase and amplitude distortion in the QPSK signal. Since the interference signal is continuous and sinusoidal, it can alter the phase and amplitude of the QPSK signal as it combines with it. This effect depends on the interference's frequency and amplitude relative to the QPSK signal. If the frequency offset of the tone interference is close to the QPSK signal's bandwidth, spectral overlap occurs. This overlap can lead to spectral interference, making it challenging for the receiver to separate the desired signal from the interference. The receiver's sensitivity to the QPSK signal is impacted by the presence



(a) In-band Tone



(b) Out-of-band Tone

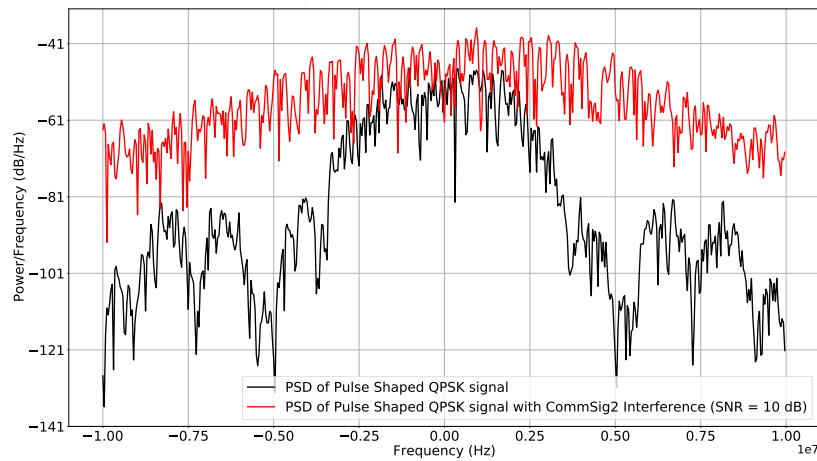
(c) *CommSignal2* Interference

Figure 5.2: Power Spectral Density (PSD) of the Pulse Shaped QPSK signal with different interference types.

of tone interference. In cases of strong interference, the receiver may struggle to detect and correctly demodulate the QPSK signal. Advanced receiver designs and interference mitigation techniques are often necessary to maintain acceptable sensitivity. In this study, we consider in-band and out-of-band tones.

The out-of-band tone signal, has a single frequency component located at a distance of Δf from the carrier frequency F_c of the desired signal. This frequency offset is large enough that it does not overlap with the digital modulation signal’s bandwidth. Figure 5.2(a) shows the comparison between power spectral density (PSD) of the “clean” QPSK signal and the PSD of the in-band tone. Figure 5.2(b) shows the comparison between PSD of the “clean” QPSK signal and the PSD of the out-of-band tone. While tone interference ¹ itself may be trivial and there are simpler ways than deep learning to eliminate it, its presence serves as an illustrative example to elucidate the concept of latent space and aids in comprehending visualization techniques.

5.3.4 Over the Air Recorded Wideband Signal Dataset

In this section, we briefly discuss: *CommSignal2* is a digital communication signal from commercially available wireless device. Since the in-band and out-of-band tones are narrow-band signals and do not present a significant challenge to interference mitigation, we identified *CommSignal2* as a more challenging interference signal that is not just located in a narrow part of the band but is spread across the entire bandwidth of the desired signal. It comprises of over-the-air (OTA) wideband signal dataset from the radio frequency (RF) challenge [231]². *CommSignal2* comprises 150 frames, each containing 43,560 samples in length. The

¹The center frequency for the in-band tone and the out-of-band tone are randomly selected from a continuous list of center frequencies. This is to ensure that the deep learning model does not learn the exact location of the tone.

²The ICASSP 2024 SP grand challenge for data-driven signal separation in radio spectrum was hosted by the Massachusetts Institute of Technology (MIT). The signal separation challenge involves isolating a signal-of-interest (SOI) from a superimposed co-channel interference signal. There are a total of four interference



Figure 5.3: Real and Imaginary Plots of *CommSignal2* Digital Communication Signal

signal generation model and system parameters are undisclosed emphasising the need for data-driven interference suppression models such as CNN-AE [232]. Figure 5.3 shows the real and imaginary parts of a random slice of *CommSignal2*.

Figure 5.2(c) shows the PSD of the pulse shaped QPSK and the *CommSignal2* at an SNR of 10 dB. We observe from the figure that the wideband signal covers a large portion of the frequency spectrum and also overlaps with the in-band portion of the QPSK signal. The wideband signal's presence within the frequency range of the QPSK signal can result in significant degradation of the QPSK signal's performance, affecting demodulation, causing errors in symbol and bit recovery, due to reduced SINR.

5.3.5 Assumptions and constraints of the system model

The model assumes idealized desired QPSK and tone signals. In real-world scenarios, signals may suffer from phase noise, frequency offsets, and timing errors, which can complicate

signal, *CommSignal2* is one of those interference signal.

interference analysis. In addition, the model assumes the wireless channel through which both the QPSK signal and the tone interference propagate is linear and time-invariant. Real-world channels can exhibit various non-linearities and time-varying characteristics that are not considered in this model. Further, the model focuses exclusively on the impact of the tone interference (for in-band and out-of-band) and *CommSignal2* dataset as an interfering signal since it is an OTA dataset collected from the ICASSP RF challenge. Other real-world wireless environments often have multiple sources of interference and noise. It is imperative to acknowledge that these assumptions have been intentionally introduced to formulate a simplified model conducive to analytical and illustrative purposes. It is well understood that practical wireless communication systems manifest a heightened level of complexity.

5.4 Methodology

5.4.1 Training data generation and dataset description

The dataset is synthesized for training an autoencoder to suppress interference to Quadrature Phase Shift Keying (QPSK) signals. The dataset consists of QPSK symbols, each with 128 bits, that are up-sampled by a factor of 4. The signal is sampled at 20 MHz. Each QPSK signal is modulated onto a carrier frequency F_c which is set to zero (baseband simulation). To this clean QPSK signal, we add a single frequency tone that can act as either out-of-band or in-band interference or add a random slice of the *CommSignal2* dataset to serve as *CommSignal2* interference. This combined signal serves as the dataset for training the autoencoder. Details of the step-by-step process of the dataset generation is described in Section 5.35.3.3. The in-band and out-of-band tone interference scenarios operate under specific frequency ranges for the training, validation and test set to cover a wide range of frequencies. We also vary the additive white Gaussian noise (AWGN) from -20 dB to 10

Table 5.1: Parameters of QPSK Generation

Parameters	Value
Number of Bits	128 I bits and Q bits
Upsampling Factor	4
Sampling Frequency	20 MHz
Freq Range (In-band Tone)	Training Set: 300 KHz to 2 MHz
	Validation Set: 1 MHz to 1.3 MHz
	Test Set: 300 KHz to 700 KHz
Freq Range (Out-of-band Tone)	Training Set: 2.5 MHz to 9.5 MHz
	Validation Set: 5 MHz to 7 MHz
	Test Set: 2.5 MHz to 4.5 MHz
SIR	−10 dB

dB at signal-to-interference ratio (SIR) of −10 dB to the entire dataset to simulate a noisy channel. The power levels of each interference type are subsequently adjusted and mixed with the clean dataset to match the desired input SNR for the dataset. These parameters have been summarized in Table 5.1.

5.4.2 Data preprocessing and preparation for training and evaluation

The generated bit wise dataset is preprocessed through several steps before being used for training and evaluation. Each QPSK signal undergoes pulse shaping and is subsequently mixed with a carrier signal. Subsequently, a interference signal is added, simulating real-world interference scenarios. This process yields complex baseband signals that encapsulate the intended QPSK signal along with the added interference effect of desired type.

We also explored the use of Fast Fourier Transforms (FFT) to preprocess the input signal before passing it onto the autoencoder. Using the FFT technique entails transforming the complex baseband signals into the frequency domain and makes feature extraction and analysis more efficient. FFT makes distinction of interference and desired signal more evident. The FFT representation, resembling an image with structured patterns (spectrogram), can

Table 5.2: Parameters of CNN-AE

Parameters	Value
Batch size	128
Learning rate	$1e - 4$ with $(0.5 \times)$ decay until $1e - 8$
Stride	1
Kernel size	15
Train/Test/Validation ratio	70%/15%/15%
Input size	512×2 (in-phase and quadrature)
No. of Parameters	Trainable parameters: 3, 167, 462
	Non-trainable parameters: 4
	Total parameters: 3, 167, 466
Optimizer	Adam
Activation function	LeakyReLU

be fed directly into the CNN layers, utilizing convolutional operations efficiently and potentially improving the model's performance. It is important to note that the raw time domain representation of the QPSK signal with interference had a superior suppression performance when passed through the CNN-AE when compared to the FFT-transformed input. The superior suppression performance observed could be attributed to the autoencoder's capacity to preserve phase information and capture intricate temporal patterns. The time domain representation retains crucial phase details necessary for demodulation, while the FFT process may introduce phase wrapping issues, especially in the presence of abrupt changes or complex interference. The results shown in Figure 5.6 are only shown for completeness sake.

5.4.3 CNN-AE Architecture

The CNN-based autoencoder (also referred to as *CNN-AE* throughout this paper) architecture proposed in this study utilizes 1D convolutional layers for both the encoding and decoding blocks. The encoder consists of several Conv1D layers with LeakyReLU activation and batch normalization. Table 5.2 shows the parameters used for the training of the autoencoder.

Figure 5.4 shows the proposed autoencoder architecture. The input tensor represents the initial layer where the input data, in this case, the time-domain representation of the signals, is fed into the network. The input sensor layer acts as the starting point for the information flow through the autoencoder. The convolutional layers are fundamental to CNNs and play a key role in detecting spatial patterns within the input data. These layers use learnable filters to convolve over the input, capturing local features and enabling the network to recognize hierarchical patterns. Activation functions such as LeakyReLU (Rectified Linear Unit) is applied after convolution. LeakyReLU adds non-linearity and makes it easier to identify intricate relationships within the data [233]. In order to standardize the input and stabilize the training process, batch normalization is employed at the input. Faster convergence is facilitated by its ability to mitigate problems such as internal covariate shift. The convolutional layers' compressed and abstracted representation of the input data is shown in the encoded features aspect of the architecture. In addition to the standard encoder-decoder structure, we have further enhanced the the design of our model by including skip connections at each spatial level between the encoder and decoder which are described in Section 5.4D. The encoded features create a condensed version of the input by capturing important information and eliminating unnecessary details. The most important features that the network has learned are contained in the latent space, which is the encoder's last layer. In the latent space, we can obtain a succinct representation by acting as a bottleneck and reducing the dimensionality of the input data. The justification for why we selected the neural network parameters can be found in the subsequent sections.

5.4.4 Encoding and decoding blocks

A sequence of Conv1D (one-dimensional convolutional) layers in the autoencoder architecture make up the encoding phase. Their purpose is to progressively extract and condense key features from the input data. LeakyReLU activation adds a non-linearity to the process,

Skip connections play an important role in our architecture by bridging the gap between the encoding and decoding layer representations. They allow for the direct transfer of feature maps from the encoder to the decoder, which is critical for preserving fine-grained details that would otherwise be lost during downsampling. These connections improve gradient flow during backpropagation, which improves training efficiency, especially for deeper networks. By combining low-level encoder features with high-level decoder features, our network gains a comprehensive understanding of the input signals, allowing for more effective signal reconstruction.

As depicted in Fig. 5.4, our autoencoder architecture incorporates skip connections that concatenate encoder activations at three distinct spatial resolutions with their corresponding levels in the decoder. Each convolutional block within the decoder follows a specific sequence: it begins with a Conv1D layer processing the output from the preceding layer. This output is then concatenated with the matching spatial activation from the encoder. Subsequently, a Conv1DTranspose layer operates on this concatenated output. The resulting output is activated through a LeakyReLU function and then spatially upsampled before being passed to the subsequent block. The Conv1DTranspose layer in the decoder is meant to reverse the convolution operation of the Conv1D layer in the encoder. This capability is crucial in effectively balancing the activations received from the encoder, ensuring a seamless integration of features within the network's architecture. This design choice ensures the effective transfer of detailed spatial information throughout the network, aiding in the precise reconstruction of the input signal. Thus skip connections allow the model to utilize both local details and global signal structure, making it robust in differentiating the signal of interest from interference, which significantly improves the quality of signal reconstruction and interference suppression.

Thus we ensure that the autoencoder captures and retains important features of the input

signal, minimizing information loss. The autoencoder's ability to accurately reconstruct the original data is critical for its role in interference suppression, as it aims to recover the clean signal from the interference-contaminated input.

5.4.5 Loss function and optimization for the autoencoder

The Mean Squared Error (MSE) loss function, a commonly used metric in regression-based problems, is applied during the autoencoder optimization process. The average squared difference between the estimated output and the true values within the dataset is computed by this function. The model's ability to minimize the difference between predicted and actual values is made easier by the selection of the MSE loss. Additionally, the Adam optimizer is used in the optimization because of its adaptable learning rate properties and efficiency. The Adam optimizer dynamically modifies the learning rate based on previous gradients, incorporating momentum and adaptive learning rates for individual parameters. Its ability to quickly converge during training and adjust to the different gradients found in different regions of the loss landscape makes it possible to fine-tune the model's parameters efficiently as it moves through the optimization process. The autoencoder model is trained more effectively and efficiently when the MSE loss function and the Adam optimizer are used together.

5.4.6 Training and validation procedures for the autoencoder

In the training process of the autoencoder, the model is trained using a dedicated training dataset, while simultaneously evaluating the model's performance on a separate validation set. The CNN autoencoder was trained utilizing a dataset comprising $r[i]$ with interference, with the target signal set as $s[i]$. This training paradigm allowed the autoencoder to learn and discern interference patterns for subsequent suppression, aiming to reconstruct $s[i]$. This

approach serves the purpose of monitoring the loss function on the validation set. To prevent overfitting and ensure generalization, an *early stopping* technique is applied [234], where the training process halts if the validation loss fails to improve over a specified number of successive epochs. An adaptive learning rate strategy is implemented to mitigate instances of stagnation in the loss function. This strategy involves reducing the learning rate upon detecting a plateau in the loss, enabling more refined adjustments to the model's weights and aiding in reaching an optimal convergence point during the training process. This combination of techniques contributes to the model's robustness, preventing overfitting while enhancing its ability to generalize effectively to new data.

5.4.7 Evaluation metrics for measuring autoencoder performance

5.4.7.1 Error Vector Magnitude (EVM)

The performance of the autoencoder is evaluated using Error Vector Magnitude (EVM), which is a measure of how far the reconstructed signal deviates from the ideal or original signal. EVM is calculated as the root mean square of the error vector, normalized by the magnitude of the ideal signal [235]. This metric is presented in decibels (dB) to quantify the performance of the autoencoder in terms of signal fidelity. The lower the EVM, the closer the reconstructed signal is to the ideal signal, indicating better performance of the autoencoder. A high EVM value, on the other hand, indicate a significant difference between the ideal and reconstructed signal. EVM provides a more objective and precise measure. In practical scenarios, achieving a perfect match between denoised and ideal signals may be challenging, and the goal is often to minimize the EVM while balancing other trade-offs in the denoising process. Additionally we can visually observe the reconstruction quality of the autoencoder by visualizing the FFT coefficients or the PSD of the signals. EVM is calculated as follows:

$$EVM(dB) = 20\log_{10}\sqrt{\frac{\frac{1}{N} \sum_{i=1}^N |s[i] - \hat{s}[i]|^2}{\frac{1}{N} \sum_{i=1}^N |s[i]|^2}} \quad (5.3)$$

where $s[i]$ is the clean signal and $\hat{s}[i]$ represents the reconstructed signal value.

5.4.7.2 Signal to Noise plus Interference Ratio (SINR)

The SINR quantifies the ratio between the power of $s[i]$ (the clean QPSK signal) and the power of $x[i]$ (residual interference signal) plus noise power denoted as σ^2 , signifying the effectiveness of the suppression technique. The signal power is typically derived from the reconstructed or output signal after interference suppression. Interference power may be obtained by analyzing the portion of the reconstructed signal that contains the unwanted interference. This also applies for the noise power. We estimate SINR as follows:

$$SINR(dB) = 20\log_{10}\frac{\frac{1}{N} \sum_{i=1}^N |s[i]|^2}{\frac{1}{N} \sum_{i=1}^N |\hat{s}[i] - s[i]|^2} \quad (5.4)$$

An increase in SINR in the context of interference suppression denotes a successful reduction of interference in relation to signal strength. Through the comparison of SINR pre- and post-autoencoder-based suppression, we are able to determine how well the model reduces interference. Greater SINR values, given in dB, highlight the model's capacity to improve signal-to-interference-plus-noise ratio performance and highlight the beneficial effects of interference suppression on overall signal integrity.

5.4.7.3 Bit Error Rate (BER)

When assessing the precision and dependability of digital communication systems using QPSK modulation, the most important metrics is the Bit Error Rate (BER). The QPSK signal is first demodulated by multiplying the received signal by the complex conjugate of the

carrier signal and subsequently down-sampling resultant signal [236]. The received symbols are then translated back to bits for comparison with the originally transmitted bits. The BER is determined for both I bits and Q bits by computing the number of bit errors in each case and dividing it by the total number of transmitted bits. The overall BER is calculated by computing the average of the BER for I bits and the BER for Q bits. The computed BER is an important performance indicator that provides insight into the accuracy of the QPSK communication system. Lower BER values indicate greater accuracy and reliability in digital data transmission.

5.5 Results and Discussion

In this section, we discuss the latent space representation of the input signals, a comparison of our CNN-AE with baseline techniques, PSD comparison between the techniques and a comparison with the three metrics utilized in this study.

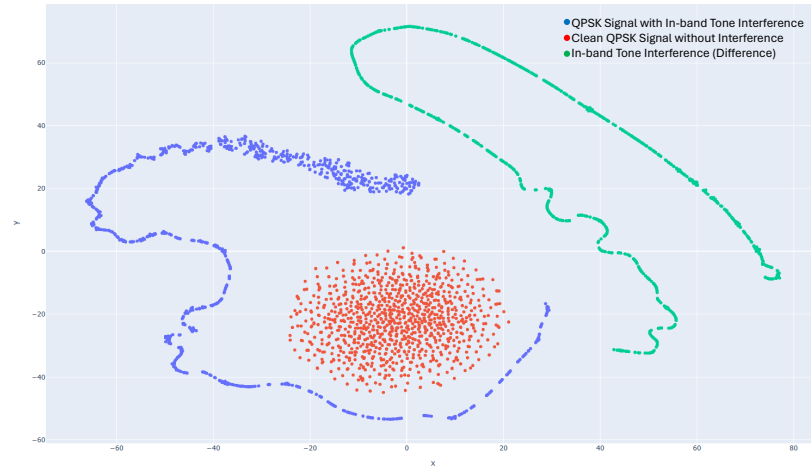
5.5.1 Latent Space Representation Visualization Using TSNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimensionality reduction technique used for visualizing high-dimensional data in lower-dimensional space [50]. t-SNE is especially good at capturing complex, nonlinear relationships in data, which makes it useful for revealing patterns and structures that might be obscured in higher dimensions. The fundamental idea behind t-SNE is to model the similarity between data points in the high-dimensional space and the low-dimensional space. A Student's t-distribution is used to measure pairwise similarities between data points in both spaces, and it is then optimized to minimize the divergence between the two distributions. Put more simply, t-SNE highlights the global structures in the data while preserving the local relationships between nearby points. In the context of the t-SNE 2D plot shown in Figure 5.5, each point represents a

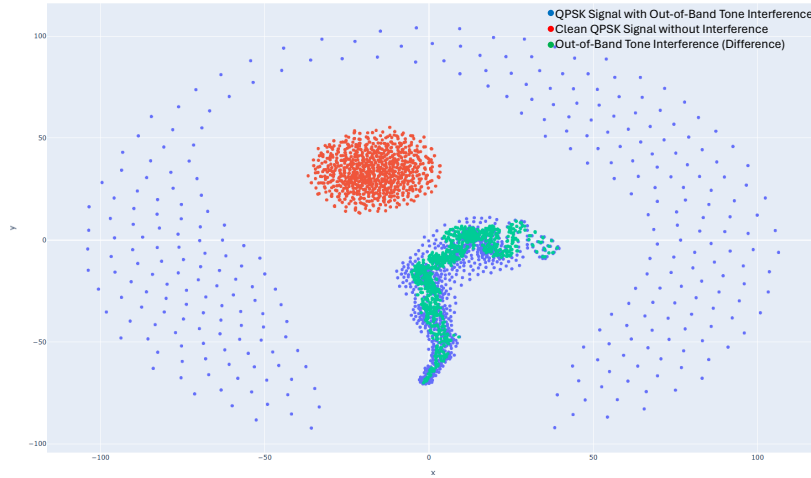
different scenario: clean QPSK, QPSK with in-band tone interference, QPSK with out-of-band tone interference, and QPSK with *CommSignal2* interference. The separation between these points in the 2D space reflects the dissimilarity or similarity of the scenarios based on the features extracted by the CNN-AE. We see that the points belonging to the same scenario (e.g. clean QPSK signal without interference) cluster together and are well-separated from points of other scenario (QPSK with interference) , it indicates that the CNN-AE has successfully captured the distinctive characteristics of all three interference scenarios. The *green* points represent actual interference when the clean QPSK signal is subtracted from the QPSK with interference.

Notably, the TSNE plots helps in verifying that the CNN-AE network is functioning as expected since the encoder is decomposing the QPSK signal with interference into a QPSK signal and the interference component so that the interference component can be properly suppressed at the decoder. Also, the CNN part of the autoencoder is able to learn the difference between the interference component and the desired signal via the convolutional layers. This separability is very crucial in the effective suppression by the CNN-AE.

Furthermore, we used t-SNE to investigate the latent space representation of the Fast Fourier transformed input interference signal. The use of t-SNE enabled the exploration of complex relationships and structures in the frequency domain, providing insights into the underlying patterns of the transformed data. Our findings are particularly promising for visualizing the latent space representations associated with both in-band and out-of-band chirp interference. This visualization technique aided in a thorough understanding of the distinct features and distributions present in the frequency domain, thereby improving the efficacy of our analysis and providing valuable insights into interference patterns and their representation within the latent space. Figure 5.6 shows the latent space representation of the FFT-transformed input QPSK signal with in-band and out-of-band tones. As noted previously, the FFT-transformed



(a) In-band Tone



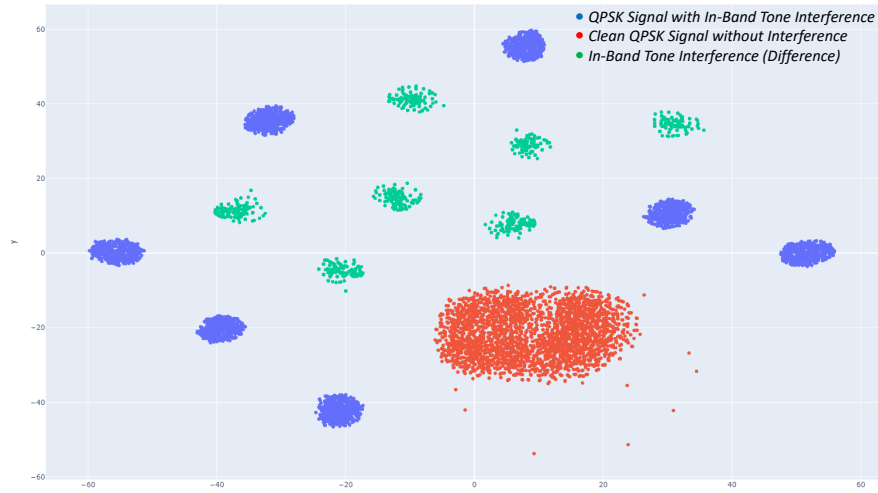
(b) Out-of-band Tone



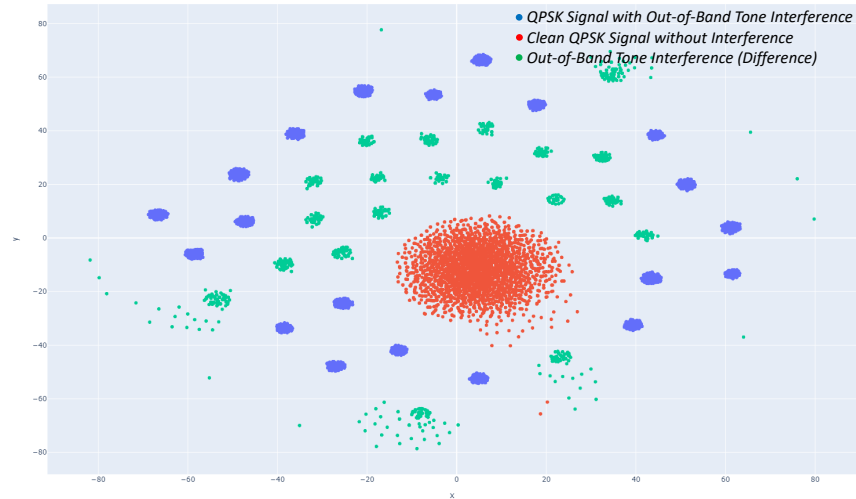
(c) CommSignal2 Interference

Figure 5.5: TSNE Latent Space Representation of the QPSK Signal with the interference types.

input signal is only shown here for completeness sake and was not used in other aspects of the study.



(a) In-band Tone



(b) Out-of-band Tone

Figure 5.6: TSNE Latent Space Representation of the FFT-transformed QPSK Signal with the interference types.

5.5.2 Comparison with Traditional Interference Suppression Techniques

We compare the CNN-AE suppression technique with two conventional suppression techniques known as matched filtering (MF) and adaptive filtering.

5.5.2.1 Matched Filtering

Matched filtering (MF) is a signal processing technique that is commonly used in communication systems to improve the detection of a desired signal in the presence of noise. It works by correlating an incoming signal with a known reference waveform, often referred to as the matched filter, optimized to the expected signal's characteristics. This filter's impulse response is the time-reversed and conjugate complex of the QPSK signal for a signal in white noise. By convolving the incoming signal with this matched filter, it enhances the SNR.

5.5.2.2 Least Squares Filtering

In the second conventional technique, we utilize an adaptive filtering technique, where we leverage a least square filter (LSF). We first estimate the least squares filter by computing the optimal filter coefficients that minimizes the squared difference between the signal with interference and the target signal (desired signal). The estimated interference component is obtained by applying the derived least squares filter to the signal with interference. The least squares filter then outputs the cleaned or filtered signal directly. This adaptive filtering approach facilitates the extraction and removal of both narrow-band and wide-band interference components from the QPSK signal, enhancing its overall quality and integrity.

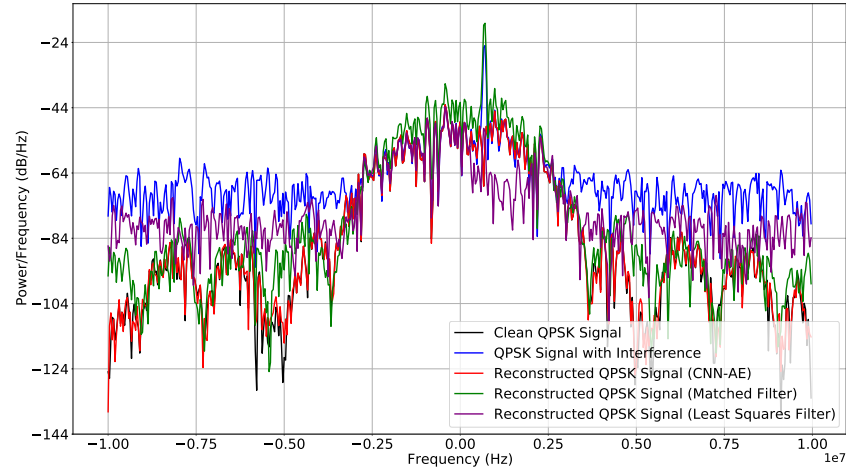
5.5.3 Power Spectral Density

In this section, we present results for in-band tone, out-of-band tone and wideband interference scenarios at SNR of 10 dB. Figure 5.7(a) shows how CNN-AE model reconstructs the interfered signal more closely with the “clean” QPSK reference, thereby providing a qualitative assessment of the efficacy of the autoencoder in suppressing in-band tone interference. Further, it elucidates the spectral characteristics of the desired signal (clean QPSK signal), contrasts it with the PSD plots of the QPSK signal affected by in-band tone interference and then showcases the reconstructed QPSK signal after the application of our CNN-AE. This is pivotal in evaluating the autoencoder’s efficacy in not only mitigating general interference effects but specifically in suppressing disruptive in-band tone interference. The plot provides a nuanced understanding of how our CNN-AE model aligns the suppressed signal’s PSD with $s[i]$, emphasizing its capability to selectively target and attenuate in-band tone interference components. In addition, we also observe how our approach outperforms MF and LSF. The PSD plot of the reconstructed QPSK signal post-CNN-AE suppression (red) demonstrates a substantial reduction in energy around the interference frequency, closely resembling the clean QPSK signal’s PSD. This outcome illustrates the CNN-AE’s ability to accurately identify and suppress the in-band tone interference while preserving the original QPSK signal’s characteristics. In contrast, the PSD plots after applying MF (green) and LSF (purple) depict partial suppression of the interference frequency. It is also important to note that MF is not able to suppress the in-band tone interference since doing this will also impact the desired signal, in essence, MF merely passes the interference through. This would also be observed when other quantitative metrics such as EVM and SINR is evaluated. In the case of LSF, we observe that the in-band tone is successfully suppressed but CNN-AE still outperforms it. MF and LSF inadequately capture the interference characteristics, leading to sub-optimal suppression, thereby exhibiting higher interference residuals compared to

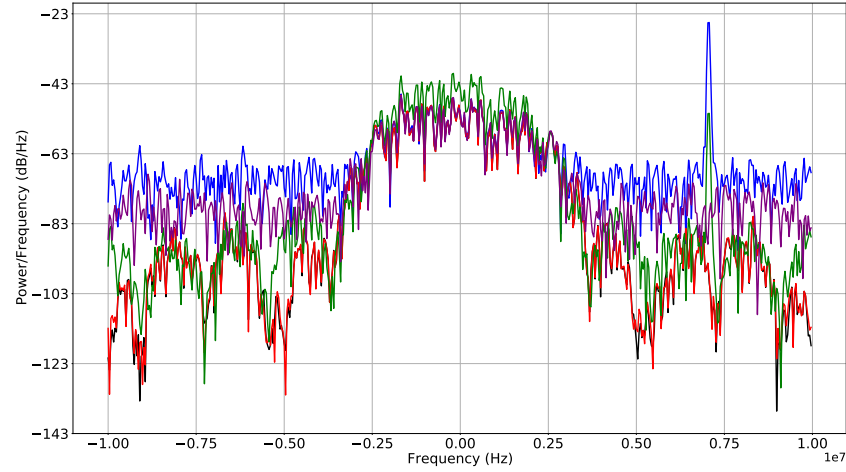
the CNN-AE approach.

Figure 5.7(b) shows the PSD comparison of the desired signal (clean QPSK signal), QPSK with out-of-band tone interference and the reconstructed QPSK with suppressed out-of-band tone interference. The plot illustrates the PSD of a pristine QPSK signal, contrasts it with the spectrum of a QPSK signal affected by out-of-band tone interference, and showcases the outcome after applying our CNN-AE, MF and LSF. This illustrates how CNN-AE is effective at attenuating unwanted out-of-band tone components while maintaining the integrity of the signal. CNN-AE effectively reduces the out-of-band tone interference, resulting in a reconstructed QPSK signal with minimal energy around the interference frequency, resembling the clean QPSK signal's PSD profile. For MF, we observed that the out-of-band tone was suppressed (to an extent) as opposed to its inability to suppress in-band tone. As noted, MF did not completely suppress the out-of-band interference since we still observe that the energy of the tone is not entirely reduced. LSF, on the other hand, suppressed the out-of-band tone appropriately when compared to MF's performance. Just as we observed in the case of in-band tone suppression, CNN-AE outperformed both MF and LSF in a more qualitative manner when their PSDs are compared. This further shows that CNN-AE consistently outperforms MF and LSF for both in-band and out-of-band tones.

Lastly, we compare the PSD performance of the interference suppression of the wideband signal using CNN-AE, MF and LSF techniques. We observe from the PSD diagram shown in Fig 5.7(c) that the suppression of the wideband (OTA) signal (*CommSignal2*) interference using CNN-AE is superior to MF and LSF. CNN-AE outperformed MF and LSF for the wideband signal interference suppression in the signal of interest (SOI) due to its ability to autonomously learn intricate features from the QPSK signal, handle non-linear relationships inherent in wideband interference, adapt and generalize to varying interference sources, and create robust feature representations. MF and LSF both rely on algorithms that might



(a) In-band Tone



(b) Out-of-band Tone

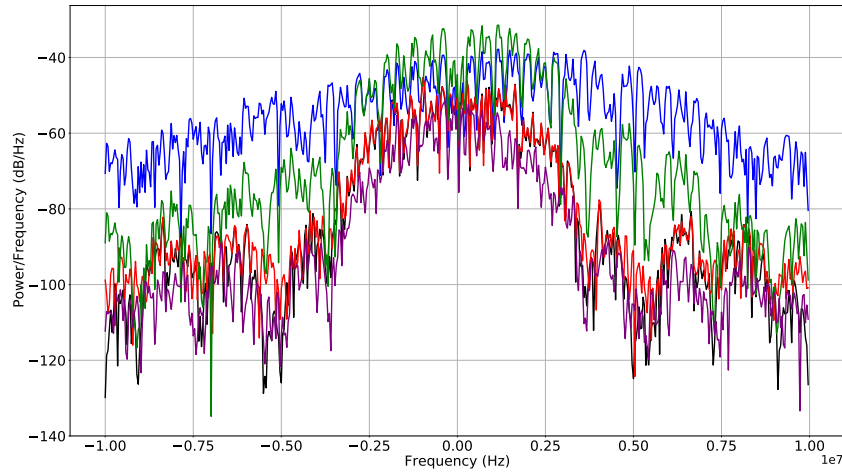
(c) *CommSignal2* Interference

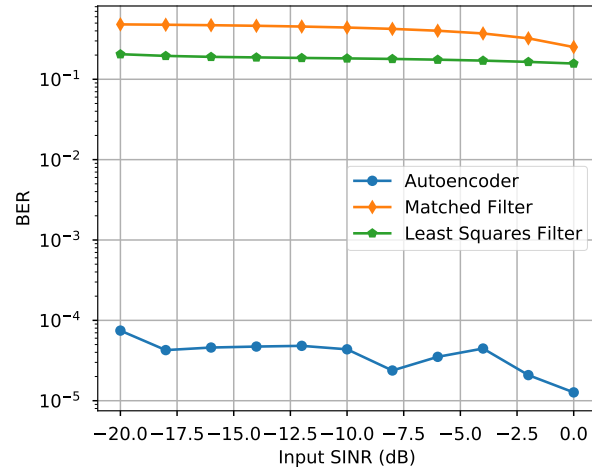
Figure 5.7: PSD Comparison of the Interference Suppression Techniques

not adapt well to the dynamic and varying nature of wideband interference sources like the *CommSignal2* interference. In addition, these techniques are not usually generalizable across different interference scenarios. While CNN-AE outperforms both traditional techniques, LSF outperforms MF since we see that the PSD for the least squares filtering is closer to the PSD of the CNN-AE approach. Finally, Figure 5.7 shows that CNN-AE outperforms MF and LSF for all three interference scenarios.

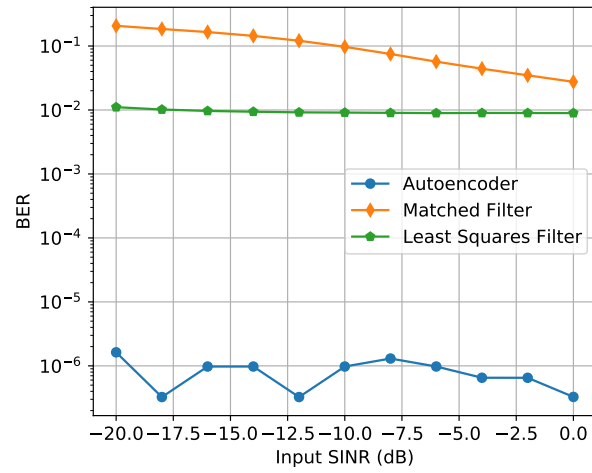
5.5.4 BER vs Input SINR & Input SNR

In this section, we show two different approaches to validate the performance of the CNN-AE model. The plot described in this section is valuable for comparing interference suppression performance as it showcases how well a communication system handles interference and noise. First, we fix the SNR at 15 dB and then vary the interference power, and then we fix the SIR and vary the SNR. This would help us to show the sensitivity of the model to the variation of noise and interference. Figure 5.8 shows the plot of input SINR versus the BER when the SNR is fixed at 15 dB.

The BER vs input SINR plots help to identify the point where error rates begin to degrade with increasing interference levels. As we observe from Figure 5.8(a) for the in-band tone interference, CNN-AE shows a lower BER across varying input SINR levels, thereby outperforming MF and LSF for in-band tone suppression. We also observe that MF has the poorest performance of all three techniques. This is not surprising since we know from the PSD plots that MF will pass the in-band tone interference through since it is not able to suppress it. LSF, on the other hand has less bit errors when compared to MF, but its performance as the input SINR increases is only marginal. CNN-AE outperforms both traditional techniques since the BER is between $1e-4$ and $1e-5$ and can potentially improve when input SINR is above 0 dB.



(a) In-band Tone



(b) Out-of-band Tone

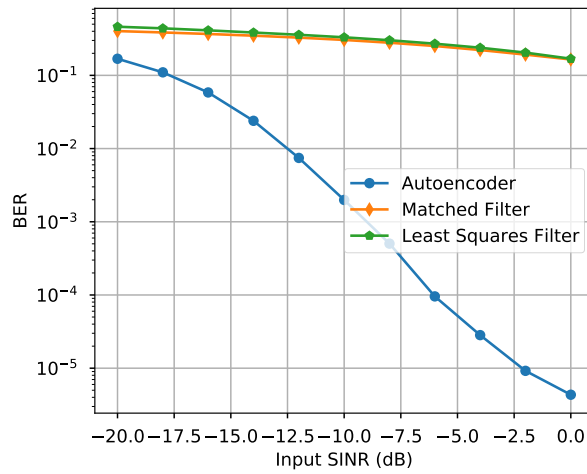
(c) *CommSignal2* Interference

Figure 5.8: BER vs Input SINR Comparison of the Interference Suppression Techniques

In Figure 5.8(b) for the out-of-band tone interference, the performance of MF and LSF for out-of-band tone suppression is considerably poorer than our CNN-AE approach. It is important to note that LSF's performance below 0 dB input SINR is superior to MF at the same input SINR range. While LSF outperforms MF, CNN-AE outperforms LSF in a more significant manner. It is also important to point out that CNN-AE has BER lower than $1e-6$ which is considerably lower than the BER performance in the in-band tone suppression shown in 5.8(a). Another salient point is that all three techniques show a lower BER for out-of-band tone when compared to the in-band tone suppression. This demonstrates that out-of-band tones are easier to suppress than in-band tones.

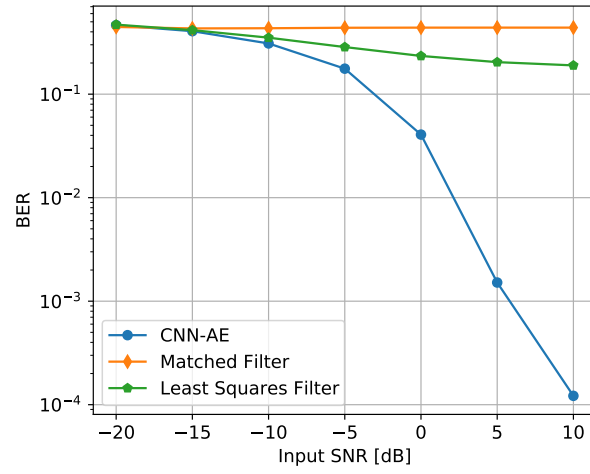
Finally, we observe from Figure 5.8(c), we observe that while MF has a slight advantage over LSF, CNN-AE outperformed both of them in suppressing the wideband interference. MF are specifically designed to maximize the SINR by aligning the filter response with the expected signal, minimizing the effect of noise and interference. MF's performance compared to LSF for wideband interference suppression can be attributed to its specific alignment with the signal, which might offer a slight advantage in extremely low signal-to-interference situations. It is important to note that the performance of the autoencoder improves progressively as the input SINR increases. Our observation show that the CNN-AE consistently maintains lower BERs, signifying its superiority in handling interference across a wide range of input SINR levels when compared to MF and LSF.

Next, we show plots of BER against SNR for various interference suppression techniques, this makes it easier to quantitatively compare their effectiveness in maintaining signal integrity while the SIR is fixed at -10 dB and SNR is varied from -20 dB to 10 dB. The plots help to identify the point where error rates begin to degrade with increasing noise levels. As we observe from Figure 5.9(a) at very low SNR (-20 dB to -15 dB), all three suppression techniques have a comparable performance for the in-band tone interference. However, CNN-

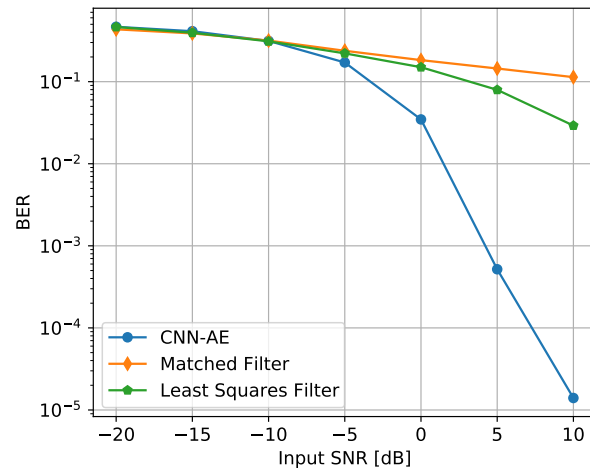
AE shows a lower BER as the input SNR increases, thereby outperforming MF and LSF for in-band tone suppression. We also observe that MF has the poorest performance of all three techniques. This is not surprising since we know from the PSD plots that MF will pass the in-band tone interference through since it is not able to suppress it. LSF, on the other hand has less bit errors when compared to MF at relatively higher input SNR, but its performance as the input SINR increases is only marginal. We continue to observe a decrease in BER as the input SNR increases for the CNN-AE suppression.

In Figure 5.9(b) for the out-of-band tone interference, we observe that all three suppression techniques have a comparable performance between -20 dB to about -8 dB and then we start to see a boost in performance for CNN-AE as input SNR increases from -9 dB and above. It is important to note that LSF's performance at input SNR of -5 dB and above is superior to MF at the same input SINR range. This shows that while all the techniques have a similar performance at lower SNR, CNN-AE's performance rapidly improves as the input SNR improves. It is also important to point out that at input SNR of 10 dB, CNN-AE has BER of about $1e - 5$ which is considerably lower than the BER performance in the in-band tone suppression shown in 5.9(a). Another salient point is that all three techniques show a lower BER for out-of-band tone when compared to the in-band tone suppression. This demonstrates that out-of-band tones are easier to suppress than in-band tones.

Finally, we observe from Figure 5.9(c), we observe that while MF has a slight advantage over LSF, CNN-AE outperformed both of them in suppressing the wideband interference at input SNR of -15 dB or greater. MF's performance compared to LSF for wideband interference suppression can be attributed to its specific alignment with the signal, which might offer a slight advantage in extremely low SNR situations. It is important to note that the performance of the autoencoder improves progressively as the input SNR increases. Our experiment show that the CNN-AE maintains lower BERs, signifying its superiority in



(a) In-band Tone



(b) Out-of-band Tone

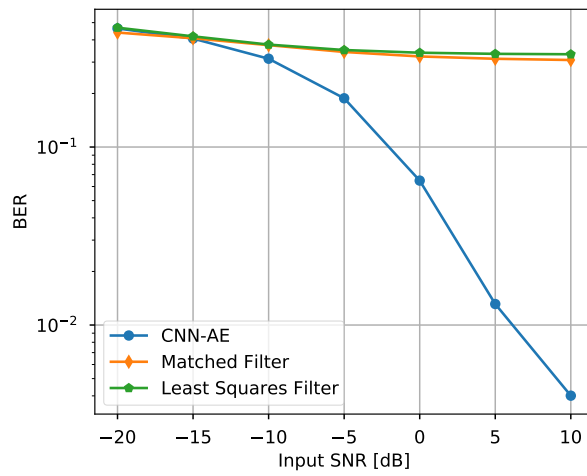
(c) *CommSignal2* Interference

Figure 5.9: BER vs Input SNR Comparison of the Interference Suppression Techniques

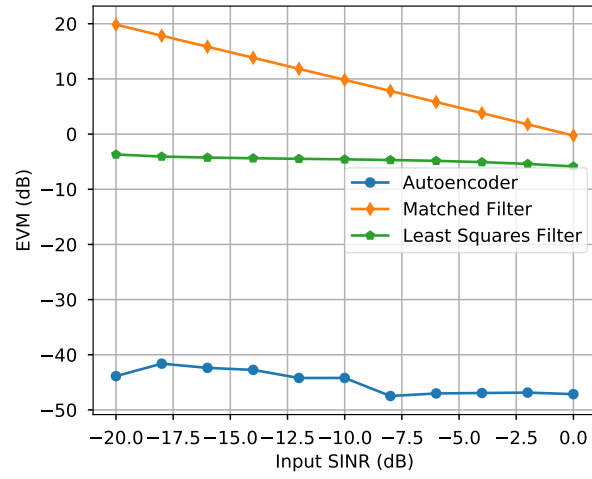
handling interference when compared to MF and LSF.

5.5.5 EVM vs Input SINR & Input SNR

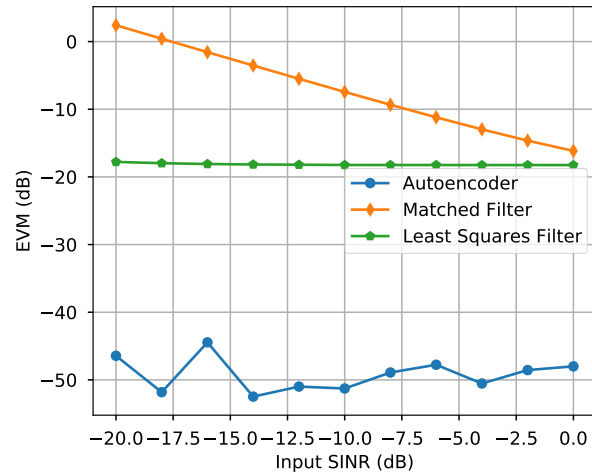
Incorporating EVM, SINR and SNR into the assessment provides a rigorous and standardized framework for evaluating the quantitative performance of the autoencoder in the different noise and interference scenarios. Again, we show the plot of the EVM versus the input SINR and the input SNR. This is crucial in understanding the sensitivity of the CNN-AE model to noise and interference. First, we show the EVM versus input SINR in Figure 5.10 and then also show the EVM versus input SNR in Figure 5.11.

The comparison of EVM and SINR values before and after suppression allows for a clear understanding of the model's ability to restore signal quality and mitigate interference effects for all the suppression techniques utilized. In Figure 5.10, we show EVM values at different input SINR values (from -20 dB to 0 dB). We observe from Figure 5.10(a) for the in-band tone interference, the performance of LSF is superior to that of MF. CNN-AE shows a superior performance compared to MF and LSF. CNN-AE's neural network architecture allows it to capture and reconstruct signal components affected by interference more effectively than the simplistic nature of the traditional filters. This adaptability and complexity in modeling interference sources contribute to the significant performance gap observed in the EVM vs input SINR plots. The EVM for MF is, however, poorer than MF and CNN-AE. It is important to note that the EVM for MF remained above 0 dB while that of LSF falls below 0 dB for the in-band tone suppression. This shows that LSF's performance is better than MF at very low input SINR.

For the out-of-band tone suppression shown in Figure 5.10(b), we observe that while CNN-AE outperforms both MF and LSF. Since we observed from the PSDs plot in 5.7(b), we know that MF is able to suppress out-of-band interference but not as good as LSF. While



(a) In-band Tone



(b) Out-of-band Tone

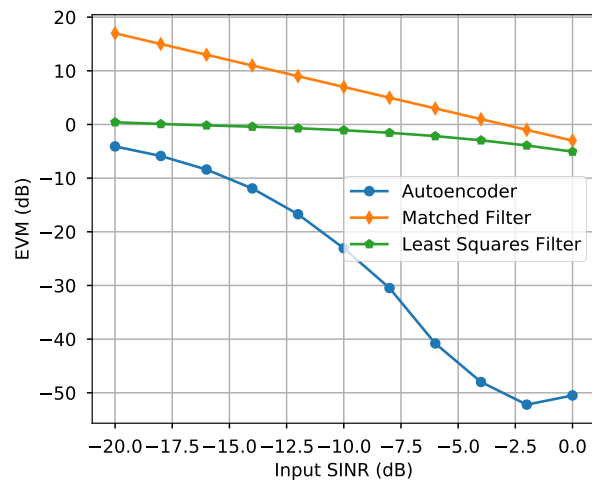
(c) *CommSignal2* Interference

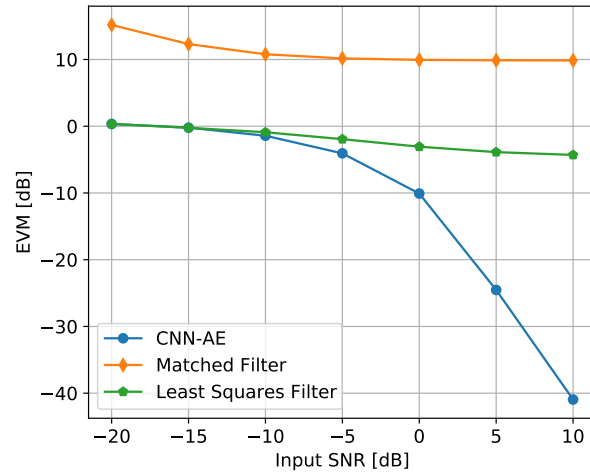
Figure 5.10: EVM vs Input SINR Comparison of the Interference Suppression Techniques

this is remarkable, CNN-AE's performance is significantly better than both techniques.

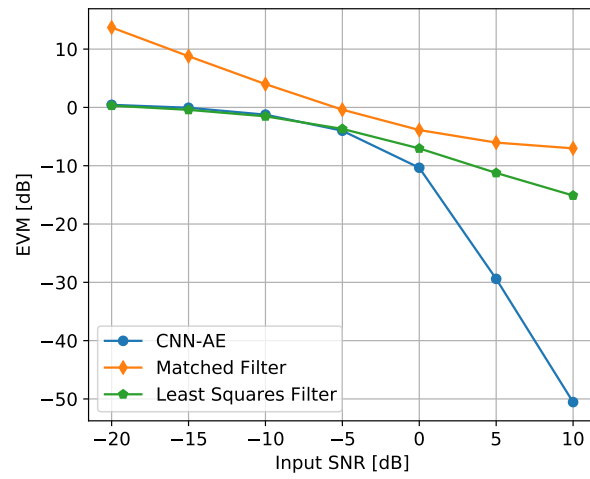
For the wideband interference suppression shown in Figure 5.10(c), we observe that LSF again outperformed MF and they start to coalesce around the 0 dB input SINR. Since least squares filters are adaptive in nature, they are able to handle complex interference patterns such as wideband interference where this patterns do not match appropriately with the assumptions relied on for matched filtering. This implies that at lower input SINR, LSF may perform better than MF when both traditional techniques are utilized. CNN-AE outperforms both techniques as seen on the plots. We see a progressive improvement in EVM values as the input SINR is increased.

Next, we look at the comparison of EVM and SNR values before and after suppression. This helps us in understanding the model's ability to restore signal quality and mitigate interference effects at different noise levels for all the suppression techniques utilized. This quantitative analysis is essential for researchers and practitioners to make informed decisions about the suitability and effectiveness of CNN-AE in real-world wireless communication applications. Also, it helps in understanding the desired trade-offs between suppression performance and reconstruction accuracy.

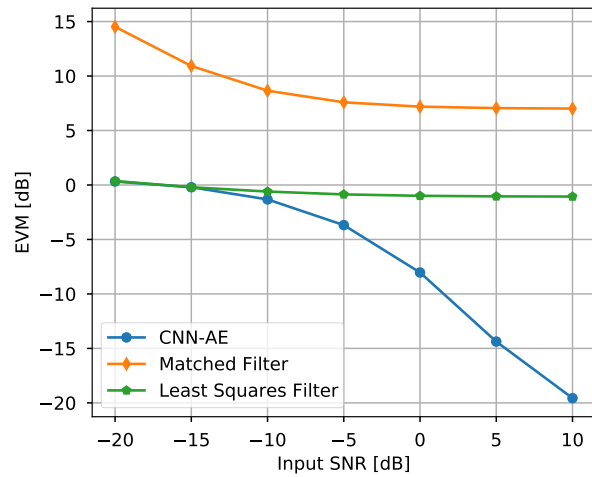
In Figure 5.11, we show EVM values at different input SNR values (from -20 dB to 10 dB). We observe from Figure 5.11(a) for the in-band tone interference, the performance of LSF is superior to that of MF. CNN-AE shows a superior performance at -10 dB or greater compared to LSF. The EVM for MF is, however, poorer than LSF and CNN-AE. It is important to note that the EVM for MF remained above 0 dB while that of LSF falls below 0 dB for the in-band tone suppression. This shows that LSF's performance is better than MF at very low input SNR. CNN-AE's neural network architecture allows it to capture and reconstruct signal components affected by interference more effectively than the simplistic nature of the traditional filters. This adaptability and complexity in modeling interference



(a) In-band Tone



(b) Out-of-band Tone



(c) CommSignal2 Interference

Figure 5.11: EVM vs Input SNR Comparison of the Interference Suppression Techniques

sources contribute to the performance gap observed in Figure 5.11(a).

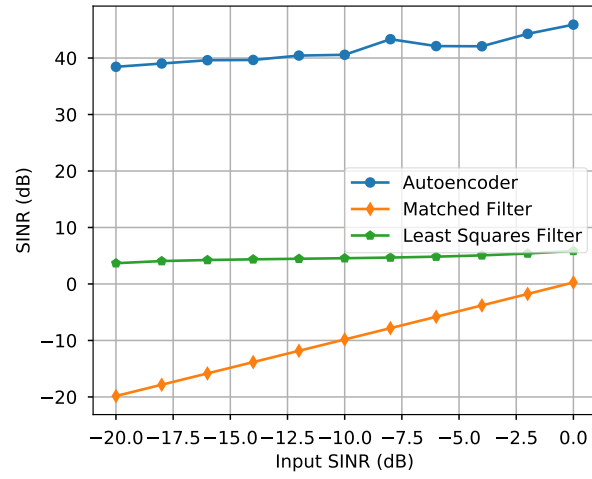
For the out-of-band tone suppression shown in Figure 5.11(b), we observe that CNN-AE outperforms both MF and LSF at input SNR ≥ -5 dB. Since we observed from the PSDs plot in 5.7(b), we know that MF is able to suppress out-of-band interference but not as good as CNN-AE and LSF. While this is remarkable, CNN-AE's performance is better than both techniques at input SNR ≥ -5 dB. This goes to show how well LSF is able to suppress out-of-band tone interference even at lower SNRs. We do observe the significant improvement in performance of CNN-AE as the input SNR increases.

For the wideband interference suppression shown in Figure 5.11(c), we observe that LSF again outperformed MF in a significant manner. Since least squares filters are adaptive in nature, they are able to handle complex interference patterns such as wideband interference where this patterns do not match appropriately with the assumptions relied on for matched filtering. This implies that at lower input SNR, LSF will perform better than MF when both traditional techniques are utilized. CNN-AE, on the other hand, outperforms LSF at -13 dB or greater as seen in Figure 5.11(c). It is important to note that CNN-AE outperforms both technique as input SNR ≥ 13 dB. We also note that MF performs poorly since it consistently shows an EVM value greater than 5 dB. This shows the weakness of MF in wideband interference scenarios.

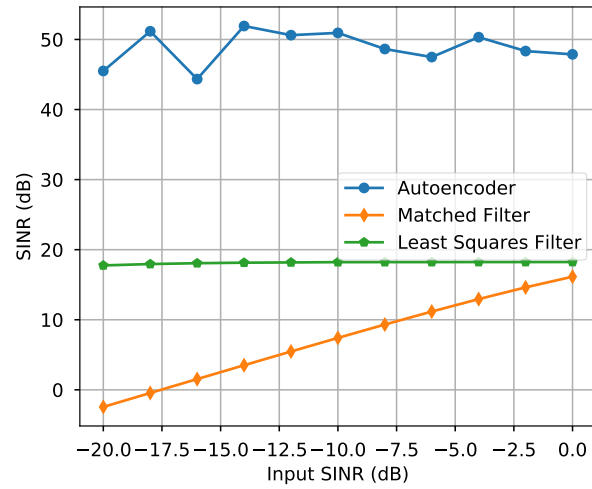
5.5.6 SINR vs Input SINR & Input SNR

In this section, we compare the SINR versus the input SINR and input SNR. This is to show the CNN-AE's sensitivity to noise and interference. We first show plots for SINR versus input SINR in Figure 5.12, while Figure 5.13 shows the plot for SINR versus input SNR.

Figure 5.12 compares the SINR of the received signal against the SINR of the input signal,



(a) In-band Tone



(b) Out-of-band Tone

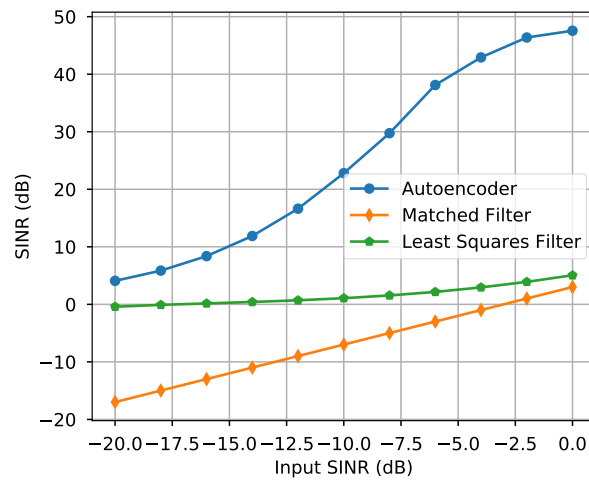
(c) *CommSignal2* Interference

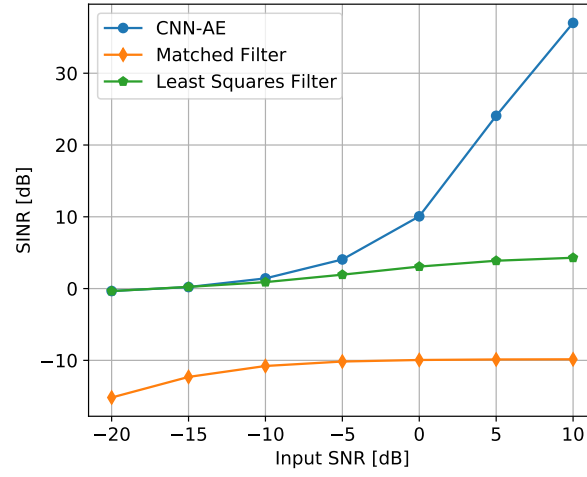
Figure 5.12: SINR vs Input SINR Comparison of the Interference Suppression Techniques

offering insights into how different suppression methods affect the incoming signal quality. It provides a direct comparison of the signal quality improvement achieved by each technique for different input SINR values. This comparison enables the identification of the range of input SINR where a suppression technique effectively enhances the signal quality and indicates an improvement in link budget. We observe from Figure 5.12(a) for in-band tone interference and Figure 5.12(b) for out-of-band tone interference that our CNN-AE approach outperformed both MF and LSF. This is because the autoencoder learns a compressed and meaningful representation of the signal, allowing it to separate the desired signal from unwanted interference more efficiently than methods that might struggle to capture the diverse and nuanced nature of interference. In addition, CNN-AE adapts its parameters based on the dataset provided during training, making it more adaptive and data-driven compared to MF and LSF. Another important thing to note is that for the in-band tone in Figure 5.12(a), LSF outperforms MF while MF merely passed the tone through since it could not suppress it given that the interference is within the band of the signal of interest. For the out-of-band tone suppression, we observe that LSF maintains a higher SINR for input SINR values below 0 dB. It has already been established that at 0 dB, MF and LSF perform comparably while CNN-AE outperforms both techniques across all input SINR ranges.

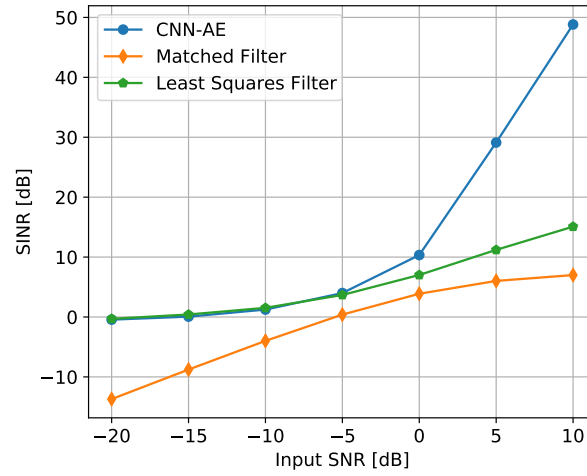
In Figure 5.13c, we observe that the improvement in SINR for CNN-AE spans the entire input SINR range. This is because the *CommSignal2* interference spans a broad range of frequencies, making it more challenging to effectively suppress across the entire bandwidth. In addition, the gradual improvement may indicate CNN-AE's effort to accommodate the diverse interference patterns present in the dataset. Again, we observe that LSF performs relatively better than MF at input $\text{SINR} \leq 0$ dB.

Next, we discuss Figure 5.13 which compares the SINR of the received signal against the SNR of the input signal, offering insights into how different suppression methods affect the

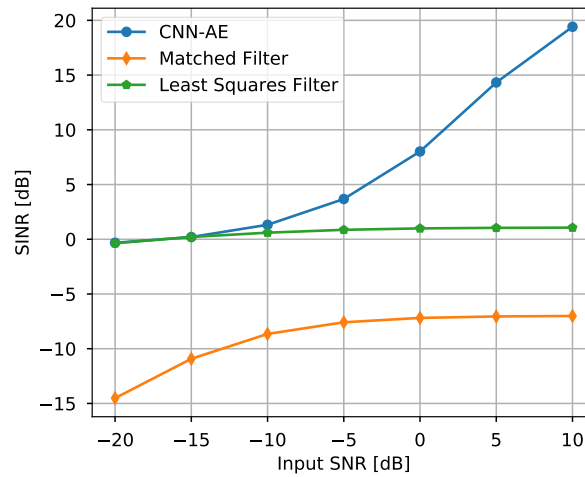
incoming signal quality as the noise level changes. This comparison enables the identification of the range of input SNR where a suppression technique effectively enhances the signal quality and indicates an improvement in link budget. We observe from Figure 5.13(a) for in-band tone interference that MF performs poorly compared to CNN-AE and LSF as SINR is at ≤ -10 dB. We also observe that between input SNR range of -20 dB and -10 dB, both CNN-AE and LSF have similar performance as the noise impacts CNN-AE's performance. CNN-AE's performance at input SNR ≥ -10 dB is superior to LSF and continues to improve significantly as input SNR increases. For out-of-band tone interference shown in Figure 5.13(b), we again observe comparable performance between CNN-AE and LSF between -20 dB and -5 dB range and a sharp improvement in performance of CNN-AE at input SNR ≥ -5 dB. This improvement in performance by CNN-AE can be attributed to its ability to adapt its parameters based on the dataset provided during training, making it more adaptive and data-driven compared to MF and LSF. While MF's performance is poorer compared to CNN-AE and LSF, we still observe a progressive gain in performance as input SNR improves. In Figure 5.13(c), we observe that CNN-AE outperforms LSF at ≥ -13 dB. MF, on the other hand performs poorly at all input SNR values since it is unable to suppress the interference at all noise level. Also, another key point to note is that LSF's performance remains below 2 dB across the entire input SNR range. This is because the *CommSignal2* interference spans a broad range of frequencies, making it more challenging to effectively suppress across the entire bandwidth. CNN-AE is able to suppress interference at all noise level for the wideband interference since it is able to accommodate the diverse interference patterns present in the dataset.



(a) In-band Tone



(b) Out-of-band Tone



(c) *CommSignal2* Interference

Figure 5.13: SINR vs SNR Comparison of the Interference Suppression Techniques

5.5.7 Key Insights

5.5.7.1 Changing Input SINR

Our study reveals that the CNN-AE consistently outperforms MF and LSF across all interference types as input SINR is varied from -20 dB to 0 dB. Specifically, CNN-AE achieves a significantly lower BER compared to MF and LSF, indicating superior accuracy in bit recovery and interference suppression. Moreover, CNN-AE exhibits notably lower EVM values, indicating better signal quality and phase/amplitude recovery accuracy compared to other methods, thus preserving signal integrity. Additionally, CNN-AE consistently yields higher SINR values across various input levels, demonstrating its superior ability to suppress interference while maintaining signal quality. PSD plots illustrate that CNN-AE effectively mitigates interference while closely preserving original QPSK signal characteristics, resulting in reduced interference residuals in the reconstructed signal. Overall, CNN-AE not only retains the desired signal but also learns representations capable of effectively suppressing noise beyond the capabilities of a matched filter.

5.5.7.2 Changing Input SNR

While we demonstrate that CNN-AE outperforms LSF and MF for all metrics (BER, EVM, SINR), it is important to point out that at lower SNR, CNN-AE and LSF have comparable performance. This is because at lower SNR levels, the performance gap between CNN-AE and LSF may narrow, as the noise becomes dominant and the ability to extract meaningful features from the input data is reduced. In such cases, all the suppression techniques (CNN-AE, LSF and MF) may struggle to distinguish between the signal and interference components, leading to comparable performance between the two methods. However, in scenarios with higher SNR, where the SNR is favorable, CNN-AE can effectively leverage these

learned features to suppress interference and recover the signal. Furthermore, the end-to-end learning approach allows the model to directly learn the input-output mapping, without the need for handcrafted feature extraction or intermediate steps. As a result, CNN-AE can better exploit the available information in the input data to suppress interference and recover the signal, particularly when SNR is higher.

5.5.8 Practical considerations and limitations

While CNN-AE shows great potential, there are certain practical constraints and limits that are important to point out with regards to using autoencoders in interference suppression applications in general. First, implementing autoencoder-based suppression techniques typically require substantial computational resources. The training phase can be computationally demanding, affecting real-time application [237] since it involves significant data processing, model optimization, and neural network changes. Our goal is to train the model offline and deploy such trained model in the wild. Another consideration is that CNN-AE's performance is strongly impacted by the quantity, quality, and diversity of the training dataset [238]. In some applications, it might be difficult to get a complete and representative dataset that covers a wide range of interference events, particularly in real-world settings. The most practical implementation would be to learn in the field.

Further, while autoencoders are quite good at learning from training data, in some scenarios, they may not be able to adjust to completely new or unknown forms of interference that are not included in the training dataset. This problem is usually referred to as out-of-distribution data [47]. Their capacity to generalize may be a drawback when handling situations with unprecedented scenarios. Our CNN-AE, however, is able to automatically determine the location of the interference and then subsequently suppress it. Also, we trained our model on test set that were not used for training. Further, there may be some latency

and real-time constraints in wireless communication applications. For instance, the time-consuming nature of processing in the autoencoder architecture may conflict with wireless communication systems' demanding latency requirements.

Another practical consideration is the issue of model interpretability. Interpreting the autoencoder model's inner workings and decision-making processes may be challenging [239], making it difficult to fully understand and explain the rationale behind its interference suppression decisions. The autoencoder model's susceptibility to overfitting and difficulties in stabilizing the learning process may have an impact on its reliability in dealing with dynamic and complex interference environments.

Ultimately, resolving these issues will play a critical role in maturing autoencoder-based solutions and broadening their applicability within the wireless communication landscape. The issues identified here are worth considering as future research directions that would help the research community to better understand how to effectively integrate deep learning in wireless communication research.

5.6 Chapter Conclusion

In the field of wireless communication systems, the implementation of autoencoder-based interference suppression algorithms offers a good improvement over the traditional interference suppression techniques where domain knowledge and the precise location of the interference is required before they can be successfully suppressed. In this extensive research, the effectiveness and potential of CNN-AE in reducing undesired signal interference—particularly in digital phase-amplitude modulation signals—were examined. Overall, across multiple performance metrics, including PSD, BER, EVM, and SINR, the CNN-AE outperforms matched filtering and least squares filtering at certain input SNR ranges. This showcases superior

interference suppression capabilities while maintaining signal integrity and preserving data quality in the presence of interference. These findings highlight CNN-AE's potential as a robust and versatile interference suppression tool for wireless communication systems, offering superior interference mitigation capabilities and promising advancements in signal quality improvement. Furthermore, examining the latent space representations of the autoencoder helps in understanding how deep learning techniques visualize the separation between the different interference signals such that they can be efficiently suppressed. Finally, it is important to note that CNN-AE worked well with wideband interference on a wideband communications signal for which classical linear time-invariant approaches did not yield significant gains.

Chapter 6

Conclusion

The research presented in this dissertation serves as a comprehensive analysis and exploration into the realm of deep learning-based classification and interference suppression in wireless communications.

In Chapter 2, the research provides a thorough review of diverse deep learning techniques dedicated to suppressing interference. This chapter stands as a pioneering effort, offering insights into the various applications and characteristics of interference addressed by these techniques. It delves into the comparison of deep learning algorithms, outlining guidelines for their successful implementation in interference suppression, thereby filling a significant gap in the literature. At its core, this chapter serves as a milestone in the literature by offering unprecedented insights into the multitude of applications and diverse interference characteristics that these cutting-edge techniques effectively address. It meticulously explores the nuances of various interference types confronted by these sophisticated deep learning methodologies, thereby broadening the understanding of their applicability across diverse scenarios and environments. By delineating their strengths, weaknesses, and unique attributes, it provides invaluable guidelines and recommendations for the judicious implementation of deep learning algorithms in real-world interference mitigation scenarios. This analytical depth and clarity fill a significant void in the existing literature, offering researchers, engineers, and practitioners practical insights for leveraging deep learning techniques effectively in suppressing interference across wireless communication systems.

Chapter 3 introduces a novel CNN architecture aimed at enhancing interference resilience and adapting the physical layer to changing interference situations. An investigation into the model complexity's impact on classification accuracy for homogeneous and heterogeneous devices highlights the critical balance between accuracy and computational efficiency, particularly essential in resource-constrained environments like IoT applications. A significant aspect of this chapter involves a detailed investigation into the relationship between model complexity and its subsequent influence on classification accuracy. This analysis, conducted across both homogeneous and heterogeneous devices, underscores the delicate equilibrium between achieving high accuracy and preserving computational efficiency. The nuanced findings of this investigation hold profound implications, particularly in resource-constrained environments such as those prevalent in Internet of Things (IoT) applications. In such scenarios, where computational power and resources are limited, striking a precise balance between model sophistication and efficiency is pivotal to ensuring optimal system performance. This offers pragmatic insights that provide practical guidelines for the development and deployment of CNN models tailored to specific interference classification tasks, allowing researchers and practitioners to make informed decisions regarding model complexity.

Building upon this, Chapter 4 investigates the intricate relationship between model performance and sample size, offering a scheme to predict classification performance. This analysis delivers crucial insights into how sample complexity affects CNN-based transmitter classification, laying the foundation for practical guidelines in selecting models for specific application scenarios. It further pioneers a robust scheme that accurately predicts classification performance, essentially serving as a guiding principle for practitioners and researchers in determining the optimal model performance concerning various sample sizes. By establishing this predictive framework, the research makes substantial strides towards addressing the

challenges posed by resource-constrained environments, especially pertinent in Internet of Things (IoT) applications characterized by limitations in computational power and storage capacities. In addition, the chapter contributes to the field by proposing practical and invaluable rules-of-thumb tailored specifically for transmitter classification using CNNs. These guidelines encapsulate a nuanced understanding of the complexities associated with classification tasks, providing actionable insights into selecting appropriate models aligned with the classification difficulty of the given dataset.

The study culminates in Chapter 5 with the development of an end-to-end CNN-AE system for interference suppression. Notably, this system operates without manual intervention, simplifying the communication pipeline. Furthermore, the study delves into latent space behavior, generalizability, training methodologies, and robust evaluation methods, showcasing the potential of this approach in overcoming interference issues in wireless communication. Beyond the development of the autoencoder system, this pivotal chapter delves deeply into several critical facets essential to comprehending the efficacy and potential of autoencoder interference suppression techniques. Foremost among these explorations is the detailed investigation into the behavior of the latent space—a pivotal element within the autoencoder framework. This study of the latent space serves as a window into understanding how the system captures, processes, and represents interference-induced variations while preserving the integrity of the desired clean signal. Such insights are invaluable for optimizing and fine-tuning the system's performance. In addition, the CNN-AE offers a significant improvement in BER after suppressing both narrowband and wideband interference. Furthermore, this chapter also ventures into the realm of generalizability, scrutinizing how effectively this system adapts and extends its interference suppression capabilities across diverse scenarios and wireless channel conditions. The exploration of training methodologies and the rigorous evaluation techniques employed in this study further underscore the robustness and reliability

of the CNN-AE approach.

Overall, this dissertation significantly contributes to the field by providing a comprehensive survey, innovative methodologies, and practical insights into the application of deep learning in classification and interference suppression. The research is instrumental in advancing the understanding and practical implementation of interference avoidance and suppression techniques, vital for ensuring the coexistence and quality of communication across diverse wireless technologies.

Bibliography

- [1] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, “A general model of wireless interference,” in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pp. 171–182, 2007.
- [2] J. Bruno, “Interference reduction in wireless networks,” *Computing Research Topics, Computing Sciences Department, Villanova University*, 2007.
- [3] M. Schmidt, D. Block, and U. Meier, “Wireless interference identification with convolutional neural networks,” in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pp. 180–185, IEEE, 2017.
- [4] J. Kim, S. Lee, Y.-H. Kim, and S.-C. Kim, “Classification of interference signal for automotive radar systems with convolutional neural network,” *IEEE Access*, vol. 8, pp. 176717–176727, 2020.
- [5] S. Grunau, D. Block, and U. Meier, “Multi-label wireless interference identification with convolutional neural networks,” *arXiv preprint arXiv:1804.04395*, 2018.
- [6] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, “Oracle: Optimized radio classification through convolutional neural networks,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 370–378, IEEE, 2019.
- [7] T. Oyedare and J.-M. J. Park, “Estimating the required training dataset size for transmitter classification using deep learning,” in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–10, IEEE, 2019.

- [8] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*, pp. 213–226, Springer, 2016.
- [9] J. A. Mahal, A. Khawar, A. Abdelhadi, and T. C. Clancy, "Spectral coexistence of MIMO radar and MIMO cellular system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 2, pp. 655–668, 2017.
- [10] A. Khawar, A. Abdel-Hadi, and T. C. Clancy, "Spectrum sharing between s-band radar and LTE cellular system: A spatial approach," in *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pp. 7–14, IEEE, 2014.
- [11] M. Carrick, J. H. Reed, and F. Harris, "An optimal filter for signals with time-varying cyclostationary statistics," in *2017 22nd International Conference on Digital Signal Processing (DSP)*, pp. 1–5, IEEE, 2017.
- [12] M. Carrick, J. H. Reed, and C. M. Spooner, "Mitigating linear-frequency-modulated pulsed radar interference to OFDM," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 3, pp. 1146–1159, 2019.
- [13] A. Batra, J. R. Zeidler, and A. Beex, "Mitigation of narrowband interference using adaptive equalizers," in *2006 14th European Signal Processing Conference*, pp. 1–5, IEEE, 2006.
- [14] M. Kesteven, G. Hobbs, R. Clement, B. Dawson, R. Manchester, and T. Uppal, "Adaptive filters revisited: Radio frequency interference mitigation in pulsar observations," *Radio science*, vol. 40, no. 5, 2005.
- [15] J. R. Mohammed, "Interference mitigation in the wireless communication systems

- using adaptive filters,” in *IOP Conference Series: Materials Science and Engineering*, vol. 1152, p. 012001, IOP Publishing, 2021.
- [16] J. D. Fite, S. P. Bruzzone, and B. G. Agee, “Blind separation of voice modulated single-side-band using the multi-target variable modulus algorithm,” in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 5, pp. 2726–2729, IEEE, 1996.
- [17] K. Kim, I. A. Akbar, K. K. Bae, J.-S. Um, C. M. Spooner, and J. H. Reed, “Cyclostationary approaches to signal detection and classification in cognitive radio,” in *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 212–215, IEEE, 2007.
- [18] B. G. Agee, *The property-restoral approach to blind adaptive signal extraction*. PhD thesis, University of California, Davis, 1989.
- [19] B. G. Agee, J. Reed, K. Cohen, and T. C. Hsia, “Simulation performance of a blind adaptive array for a realistic mobile channel,” in *IEEE 43rd Vehicular Technology Conference*, pp. 97–100, IEEE, 1993.
- [20] J. Palicot and C. Roland, “A new concept for wireless reconfigurable receivers,” *IEEE Communications Magazine*, vol. 41, no. 7, pp. 124–132, 2003.
- [21] J. H. Reed, N. Yuen, and T. C. Hsia, “An optimal receiver using a time-dependent adaptive filter,” *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 187–190, 1995.
- [22] J. Yu, M. Alhassoun, and R. M. Buehrer, “Interference classification using deep neural networks,” *arXiv preprint arXiv:2002.00533*, 2020.

- [23] T. Oyedare, V. K. Shah, D. J. Jakubisin, and J. H. Reed, “Interference suppression using deep learning: Current approaches and open challenges,” *IEEE Access*, 2022.
- [24] Y. Zhao, W.-C. Wong, H. K. Garg, T. Feng, Z. Zhang, and L. Tang, “Indoor position recognition and interference classification with a nested lstm network,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.
- [25] H. F. Atlam, R. J. Walters, and G. B. Wills, “Fog computing and the internet of things: A review,” *big data and cognitive computing*, vol. 2, no. 2, p. 10, 2018.
- [26] F. Pereira, R. Correia, P. Pinho, S. I. Lopes, and N. B. Carvalho, “Challenges in resource-constrained iot devices: Energy and communication as critical success factors for future iot deployment,” *Sensors*, vol. 20, no. 22, p. 6420, 2020.
- [27] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang, “Deep learning for health informatics,” *IEEE journal of biomedical and health informatics*, vol. 21, no. 1, pp. 4–21, 2016.
- [28] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem, “Classification using deep learning neural networks for brain tumors,” *Future Computing and Informatics Journal*, vol. 3, no. 1, pp. 68–71, 2018.
- [29] C. Morin, L. S. Cardoso, J. Hoydis, J.-M. Gorce, and T. Vial, “Transmitter classification with supervised deep learning,” in *International Conference on Cognitive Radio Oriented Wireless Networks*, pp. 73–86, Springer, 2019.
- [30] M. Zheleva, R. Chandra, A. Chowdhery, A. Kapoor, and P. Garnett, “Txminer: Identifying transmitters in real-world spectrum measurements,” in *Dynamic Spectrum Access Networks (DySPAN), 2015 IEEE International Symposium on*, pp. 94–105, IEEE, 2015.

- [31] K. Longi, T. Pulkkinen, and A. Klami, “Semi-supervised convolutional neural networks for identifying Wi-Fi interference sources,” in *Asian Conference on Machine Learning*, pp. 391–406, PMLR, 2017.
- [32] K. Merchant, S. Revay, G. Stantchev, and B. Noursain, “Deep learning for RF device fingerprinting in cognitive communication networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 160–167, 2018.
- [33] K. Youssef, L.-S. Bouchard, K. Haigh, H. Krovi, J. Silovsky, and C. Vander Valk, “Machine learning approach to RF transmitter identification,” *arXiv preprint arXiv:1711.01559*, 2017.
- [34] M. Kulin, T. Kazaz, I. Moerman, and E. De Poorter, “End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications,” *IEEE Access*, vol. 6, pp. 18484–18501, 2018.
- [35] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, “Deep learning convolutional neural networks for radio identification,” *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, 2018.
- [36] R. L. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. H. Ngo, “Predicting sample size required for classification performance,” *BMC medical informatics and decision making*, vol. 12, no. 1, p. 8, 2012.
- [37] N. Boonyanunta and P. Zeephongsekul, “Predicting the relationship between the size of training sample and the predictive power of classifiers,” in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 529–535, Springer, 2004.
- [38] S. Mukherjee, P. Tamayo, S. Rogers, R. Rifkin, A. Engle, C. Campbell, T. R. Golub,

- and J. P. Mesirov, “Estimating dataset size requirements for classifying dna microarray data,” *Journal of computational biology*, vol. 10, no. 2, pp. 119–142, 2003.
- [39] A. Clauset, C. R. Shalizi, and M. E. Newman, “Power-law distributions in empirical data,” *SIAM review*, vol. 51, no. 4, pp. 661–703, 2009.
- [40] A. Alwosheel, S. van Cranenburgh, and C. G. Chorus, “Is your dataset big enough? sample size requirements when using artificial neural networks for discrete choice analysis,” *Journal of choice modelling*, vol. 28, pp. 167–182, 2018.
- [41] J. Cho, K. Lee, E. Shin, G. Choy, and S. Do, “How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?,” *arXiv preprint arXiv:1511.06348*, 2015.
- [42] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. Patwary, M. Ali, Y. Yang, and Y. Zhou, “Deep learning scaling is predictable, empirically,” *arXiv preprint arXiv:1712.00409*, 2017.
- [43] K. Patil, K. Skouby, A. Chandra, and R. Prasad, “Spectrum occupancy statistics in the context of cognitive radio,” in *2011 The 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 1–5, IEEE, 2011.
- [44] W. Xie, S. Hu, C. Yu, P. Zhu, X. Peng, and J. Ouyang, “Deep learning in digital modulation recognition using high order cumulants,” *IEEE Access*, vol. 7, pp. 63760–63766, 2019.
- [45] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, 2015.

- [46] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1765–1773, 2017.
- [47] J. Liu, T. Oyedare, and J.-M. Park, “Detecting out-of-distribution data in wireless communications applications of deep learning,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, pp. 2476–2487, 2022.
- [48] M. U. A. Siddiqui, H. Abumarshoud, L. Bariah, S. Muhaidat, M. A. Imran, and L. Mohjazi, “Ullc in beyond 5g and 6g networks: An interference management perspective,” *IEEE Access*, 2023.
- [49] S. H. A. Kazmi, F. Qamar, R. Hassan, and K. Nisar, “Routing-based interference mitigation in sdn enabled beyond 5g communication networks: A comprehensive survey,” *IEEE Access*, 2023.
- [50] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [51] S. Chen, J. Taghia, T. Fei, U. Kühnau, N. Pohl, and R. Martin, “A DNN autoencoder for automotive radar interference mitigation,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4065–4069, IEEE, 2021.
- [52] 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.
- [53] T. Erpek, T. J. O’Shea, and T. C. Clancy, “Learning a physical layer scheme for the

- MIMO interference channel,” in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, 2018.
- [54] C.-L. Lin, K. C.-J. Lin, C.-C. Lee, and Y. Tsao, “Cross-technology interference mitigation using fully convolutional denoising autoencoders,” in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2020.
- [55] Z. Geng, H. Yan, J. Zhang, and D. Zhu, “Deep-learning for radar: A survey,” *IEEE Access*, 2021.
- [56] D. Chen, Y. Zhuang, J. Huai, X. Sun, X. Yang, M. A. Javed, J. Brown, Z. Sheng, and J. Thompson, “Coexistence and interference mitigation for WPANs and WLANs from traditional approaches to deep learning: A review,” *IEEE Sensors Journal*, 2021.
- [57] N. Azmi, L. Kamarudin, M. Mahmuddin, A. Zakaria, A. Shakaff, S. Khatun, K. Kamarudin, and M. Morshed, “Interference issues and mitigation method in wsn 2.4 GHz ISM band: A survey,” in *2014 2nd International Conference on Electronic Design (ICED)*, pp. 403–408, IEEE, 2014.
- [58] K. R. Chowdhury and I. F. Akyildiz, “Interferer classification, channel selection and transmission adaptation for wireless sensor networks,” in *2009 IEEE International Conference on Communications*, pp. 1–5, IEEE, 2009.
- [59] M. Kunert, F. Bodereau, M. Goppelt, C. Fischer, A. John, T. Wixforth, A. Ossowska, T. Schipper, *et al.*, “D1. 5-study on the state-of-the-art interference mitigation techniques,” *Technical report, European Commission: MOre Safety for All by Radar*, 2010.
- [60] D. Lu, Q. Feng, and R. Wu, “Survey on interference mitigation via adaptive array processing in GPS,” *Piers Online*, vol. 2, no. 4, pp. 357–362, 2006.

- [61] T. T. Le and S. Moh, “Interference mitigation schemes for wireless body area sensor networks: A comparative survey,” *Sensors*, vol. 15, no. 6, pp. 13805–13838, 2015.
- [62] M. Toth, P. Meissner, A. Melzer, and K. Witrisal, “Performance comparison of mutual automotive radar interference mitigation algorithms,” in *2019 IEEE Radar Conference (RadarConf)*, pp. 1–6, IEEE, 2019.
- [63] A. Nabil, Y. T. Hou, R. Zhu, W. Lou, and S. F. Midkiff, “Recent advances in interference management for wireless networks,” *IEEE Network*, vol. 29, no. 5, pp. 83–89, 2015.
- [64] S. Kajita, T. Amano, H. Yamaguchi, T. Higashino, and M. Takai, “Wi-Fi channel selection based on urban interference measurement,” in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pp. 143–150, 2016.
- [65] X. Yang and A. P. Petropulu, “Co-channel interference modeling and analysis in a poisson field of interferers in wireless communications,” *IEEE Transactions on signal processing*, vol. 51, no. 1, pp. 64–76, 2003.
- [66] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, “Practical, real-time, full duplex wireless,” in *Proceedings of the 17th annual international conference on Mobile computing and networking*, pp. 301–312, 2011.
- [67] D. Bharadia, E. McMillin, and S. Katti, “Full duplex radios,” in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pp. 375–386, 2013.
- [68] A. Balatsoukas-Stimming, “Non-linear digital self-interference cancellation for in-band full-duplex radios using neural networks,” in *2018 IEEE 19th International Workshop*

- on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2018.
- [69] S. Farid, Y. Zia, A. Farhad, and F. B. Hussain, “Homogeneous interference mitigation techniques for wireless body area network under coexistence: A survey,” in *2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 124–129, IEEE, 2016.
- [70] L. B. Milstein, “Interference rejection techniques in spread spectrum communications,” *Proceedings of the IEEE*, vol. 76, no. 6, pp. 657–671, 1988.
- [71] R. Vijayan and H. V. Poor, “Nonlinear techniques for interference suppression in spread-spectrum systems,” *IEEE Transactions on Communications*, vol. 38, no. 7, pp. 1060–1065, 1990.
- [72] Z. Hu, R. Susitaival, Z. Chen, I.-K. Fu, P. Dayal, and S. K. Baghel, “Interference avoidance for in-device coexistence in 3GPP LTE-advanced: Challenges and solutions,” *IEEE Communications Magazine*, vol. 50, no. 11, pp. 60–67, 2012.
- [73] S. Bayhan, A. Zubow, and A. Wolisz, “Coexistence gaps in space: Cross-technology interference-nulling for improving LTE-U/Wi-Fi coexistence,” *arXiv preprint arXiv:1710.07927*, 2017.
- [74] G. Naik, J. Liu, and J.-M. J. Park, “Coexistence of wireless technologies in the 5 GHz bands: A survey of existing solutions and a roadmap for future research,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1777–1798, 2018.
- [75] B. Chen, J. Chen, Y. Gao, and J. Zhang, “Coexistence of LTE-LAA and Wi-Fi on 5 GHz with corresponding deployment scenarios: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 7–32, 2016.

- [76] M. Tercero, K. W. Sung, and J. Zander, “Impact of aggregate interference on meteorological radar from secondary users,” in *2011 IEEE Wireless Communications and Networking Conference*, pp. 2167–2172, IEEE, 2011.
- [77] E. Saltikoff, J. Y. Cho, P. Tristant, A. Huuskonen, L. Allmon, R. Cook, E. Becker, and P. Joe, “The threat to weather radars by wireless technology,” *Bulletin of the American Meteorological Society*, vol. 97, no. 7, pp. 1159–1167, 2016.
- [78] J. B. Kenney, “Dedicated short-range communications (DSRC) standards in the united states,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [79] G. Naik, J. Liu, and J.-M. J. Park, “Coexistence of dedicated short range communications (DSRC) and Wi-Fi: Implications to Wi-Fi performance,” in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.
- [80] J. Lansford, J. B. Kenney, and P. Ecclesine, “Coexistence of unlicensed devices with DSRC systems in the 5.9 GHz its band,” in *2013 IEEE Vehicular Networking Conference*, pp. 9–16, IEEE, 2013.
- [81] A. Hithnawi, H. Shafagh, and S. Duquennoy, “Understanding the impact of cross technology interference on IEEE 802.15. 4,” in *Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pp. 49–56, 2014.
- [82] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy, “Crosszig: combating cross-technology interference in low-power wireless networks,” in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12, IEEE, 2016.
- [83] W. Wang, D. He, W. Jia, X. Chen, T. Gu, H. Liu, X. Sun, G. Chen, and

- F. Wu, “PRComm: Anti-interference cross-technology communication based on pseudo-random sequence,” in *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021)*, pp. 163–175, 2021.
- [84] W. Liu, Y. Xia, M. Xu, J. Xie, R. Luo, and D. Huang, “Distributed and accurate packet reception rate estimation under cross-technology interference,” in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2020.
- [85] T. Pulkkinen, J. K. Nurminen, and P. Nurmi, “Understanding Wi-Fi cross-technology interference detection in the real world,” in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 954–964, IEEE, 2020.
- [86] D. Gascón, “Security in 802.15. 4 and ZigBee networks,” *Apr*, vol. 28, pp. 1–5, 2009.
- [87] D. Korpi, L. Anttila, and M. Valkama, “Nonlinear self-interference cancellation in mimo full-duplex transceivers under crosstalk,” *EURASIP journal on wireless Communications and Networking*, vol. 2017, no. 1, pp. 1–15, 2017.
- [88] L. Anttila, D. Korpi, E. Antonio-Rodríguez, R. Wichman, and M. Valkama, “Modeling and efficient cancellation of nonlinear self-interference in MIMO full-duplex transceivers,” in *2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 777–783, IEEE, 2014.
- [89] M. A. Islam and B. Smida, “A comprehensive self-interference model for single-antenna full-duplex communication systems,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2019.
- [90] A. Hithnawi, “Exploiting physical layer information to mitigate cross-technology in-

- interference effects on low-power wireless networks,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, pp. 1–2, 2013.
- [91] F. Hermans, L.-Å. Larzon, O. Rensfelt, and P. Gunningberg, “A lightweight approach to online detection and classification of interference in 802.15.4-based sensor networks,” *Acm Sigbed Review*, vol. 9, no. 3, pp. 11–20, 2012.
- [92] F. Vanheel, I. Moerman, and J. Verhaevert, “Spectral interference study of Wi-Fi on wireless sensor networks,” in *8th UGent PhD symposium*, pp. 108–108, 2007.
- [93] S.-U. Yoon, R. Murawski, E. Ekici, S. Park, and Z. H. Mir, “Adaptive channel hopping for interference robust wireless sensor networks,” in *2010 IEEE International Conference on Communications*, pp. 1–5, IEEE, 2010.
- [94] J. Yun, B. Lee, J. Li, and K. Han, “A channel switching scheme for avoiding interference of between IEEE 802.15. 4 and other networks,” in *2008 International Multi-symposiums on Computer and Computational Sciences*, pp. 136–139, IEEE, 2008.
- [95] M. Hassan, L. Kamarudin, and A. Zakaria, “Mitigating interference in a heterogeneous wireless network using channel selection,” in *Special Issue for International Conference of Advanced Materials Engineering and Technology*, 2013.
- [96] C. Fischer, *Studies on the interference behavior of automotive radar sensors*. Cuvillier Verlag, 2016.
- [97] J. Bechter, F. Roos, M. Rahman, and C. Waldschmidt, “Automotive radar interference mitigation using a sparse sampling approach,” in *2017 European Radar Conference (EURAD)*, pp. 90–93, IEEE, 2017.
- [98] F. Marvasti, M. Azghani, P. Imani, P. Pakrouh, S. J. Heydari, A. Golmohammadi, A. Kazerouni, and M. Khalili, “Sparse signal processing using iterative method with

- adaptive thresholding (IMAT),” in *2012 19th International Conference on Telecommunications (ICT)*, pp. 1–6, IEEE, 2012.
- [99] S. Alland, W. Stark, M. Ali, and M. Hegde, “Interference in automotive radar systems: Characteristics, mitigation techniques, and current and future research,” *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 45–59, 2019.
- [100] B.-E. Tullsson, “Topics in FMCW radar disturbance suppression,” in *Radar 97 (Conf. Publ. No. 449)*, pp. 1–5, IET, 1997.
- [101] Y. Chan, J. Lavoie, and J. Plant, “A parameter estimation approach to estimation of frequencies of sinusoids,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 214–219, 1981.
- [102] M. Wagner, F. Sulejmani, A. Melzer, P. Meissner, and M. Huemer, “Threshold-free interference cancellation method for automotive FMCW radar systems,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, 2018.
- [103] J. Wang, “Cfar-based interference mitigation for fmcw automotive radar systems,” *arXiv preprint arXiv:2101.01257*, 2021.
- [104] M. A. Richards, *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005.
- [105] N.-C. Ristea, A. Anghel, and R. T. Ionescu, “Fully convolutional neural networks for automotive radar interference mitigation,” *arXiv preprint arXiv:2007.11102*, 2020.
- [106] J. Rock, M. Toth, E. Messner, P. Meissner, and F. Pernkopf, “Complex signal denoising and interference mitigation for automotive radar using convolutional neural networks,” in *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2019.

- [107] J. Rock, W. Roth, M. Toth, P. Meissner, and F. Pernkopf, "Resource-efficient deep neural networks for automotive radar interference mitigation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 927–940, 2021.
- [108] A. Fuchs, J. Rock, M. Toth, P. Meissner, and F. Pernkopf, "Complex-valued convolutional neural networks for enhanced radar signal denoising and interference mitigation," in *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6, IEEE, 2021.
- [109] T. Xu, T. Xu, and I. Darwazeh, "Deep learning for interference cancellation in non-orthogonal signal based optical communication systems," in *2018 Progress in Electromagnetics Research Symposium (PIERS-Toyama)*, pp. 241–248, IEEE, 2018.
- [110] C. Häger and H. D. Pfister, "Nonlinear interference mitigation via deep neural networks," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1–3, IEEE, 2018.
- [111] N. Promsuk, "Development of interference mitigation techniques based artificial neural network for IoT network," in *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 374–377, IEEE, 2021.
- [112] M. Mosavi and F. Shafiee, "Narrowband interference suppression for GPS navigation using neural networks," *GPS solutions*, vol. 20, no. 3, pp. 341–351, 2016.
- [113] Y. C. Shi and Y. B. Liu, "Interference suppression for satellite navigation based on neural network and wavelet packets transform," in *Advanced Materials Research*, vol. 457, pp. 1111–1117, Trans Tech Publ, 2012.
- [114] K. Muranov, M. A. Islam, B. Smida, and N. Devroye, "On deep learning assisted self-

- interference estimation in a full-duplex relay link,” *IEEE Wireless Communications Letters*, 2021.
- [115] H. Guo, S. Wu, H. Wang, and M. Daneshmand, “DSIC: Deep learning based self-interference cancellation for in-band full duplex wireless,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.
- [116] C. Shi, Y. Hao, Y. Liu, and S. Shao, “Digital self-interference cancellation for full duplex wireless communication based on neural networks,” in *2019 4th International Conference on Communication and Information Systems (ICCIS)*, pp. 53–57, IEEE, 2019.
- [117] Y. Kurzo, A. Burg, and A. Balatsoukas-Stimming, “Design and implementation of a neural network aided self-interference cancellation scheme for full-duplex radios,” in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 589–593, IEEE, 2018.
- [118] W. Zhang, J. Yin, D. Wu, G. Guo, and Z. Lai, “A self-interference cancellation method based on deep learning for beyond 5g full-duplex system,” in *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pp. 1–5, IEEE, 2018.
- [119] Z. Yang, T. Zhao, Y. Zhao, and J. Yu, “Iterative narrowband interference suppression for ds-cdma systems using feed-forward neural network,” in *2010 IEEE 71st Vehicular Technology Conference*, pp. 1–5, IEEE, 2010.
- [120] J. Mun, H. Kim, and J. Lee, “A deep learning approach for automotive radar interference mitigation,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, IEEE, 2018.

- [121] J. Mun, S. Ha, and J. Lee, “Automotive radar signal interference mitigation using RNN with self attention,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3802–3806, IEEE, 2020.
- [122] M. Wei-Lung, “GPS interference mitigation using derivative-free kalman filter-based RNN,” *Radioengineering*, vol. 25, no. 3, p. 519, 2016.
- [123] P.-R. Chang and J.-T. Hu, “Narrow-band interference suppression in spread-spectrum CDMA communications using pipelined recurrent neural networks,” *IEEE Transactions on Vehicular Technology*, vol. 48, no. 2, pp. 467–477, 1999.
- [124] D.-J. Xu, P.-J. Zhao, F. Shen, and H. Zhao, “Narrowband interference suppression using rkf-based recurrent neural network in spread spectrum system,” in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–5, IEEE, 2008.
- [125] J. Fuchs, A. Dubey, M. Lübke, R. Weigel, and F. Lurz, “Automotive radar interference mitigation using a convolutional autoencoder,” in *2020 IEEE International Radar Conference (RADAR)*, pp. 315–320, IEEE, 2020.
- [126] 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.
- [127] D. Wu, M. Nekovee, and Y. Wang, “Deep learning-based autoencoder for m-User wireless interference channel physical layer design,” *IEEE Access*, vol. 8, pp. 174679–174691, 2020.
- [128] D. Wu, M. Nekovee, and Y. Wang, “An adaptive deep learning algorithm based autoencoder for interference channels,” *arXiv preprint arXiv:1902.06841*, 2019.

- [129] P. Lynggaard, “Using machine learning for adaptive interference suppression in wireless sensor networks,” *IEEE Sensors Journal*, vol. 18, no. 21, pp. 8820–8826, 2018.
- [130] A. Suzain, R. A. Rashid, M. Sarijari, A. S. Abdullah, and O. A. Aziz, “Machine learning based lightweight interference mitigation scheme for wireless sensor network,” *Telkomnika*, vol. 18, no. 4, pp. 1762–1770, 2020.
- [131] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [132] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168, 2006.
- [133] T. Hastie, R. Tibshirani, and J. Friedman, “Overview of supervised learning,” in *The elements of statistical learning*, pp. 9–41, Springer, 2009.
- [134] Y. Bengio, “Deep learning of representations for unsupervised and transfer learning,” in *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36, JMLR Workshop and Conference Proceedings, 2012.
- [135] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [136] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.
- [137] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews],” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

- [138] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” *arXiv preprint arXiv:1905.02249*, 2019.
- [139] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.
- [140] W. Lee, M. Kim, and D.-H. Cho, “Deep sensing: Cooperative spectrum sensing based on convolutional neural networks,” *arXiv preprint arXiv:1705.08164*, 2017.
- [141] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, “Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.
- [142] T. J. O’Shea, T. Erpek, and T. C. Clancy, “Physical layer deep learning of encodings for the MIMO fading channel,” in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 76–80, IEEE, 2017.
- [143] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [144] 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)*, pp. 529–532, IEEE, 2018.
- [145] 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)*, pp. 1–5, IEEE, 2018.
- [146] 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.
- [147] F. Liang, C. Shen, and F. Wu, “An iterative BP-CNN architecture for channel decoding,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 144–159, 2018.
- [148] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, IEEE, 2017.
- [149] J. Akeret, C. Chang, A. Lucchi, and A. Refregier, “Radio frequency interference mitigation using deep convolutional neural networks,” *Astronomy and Computing*, vol. 18, pp. 35–39, 2017.
- [150] W. Fan, F. Zhou, M. Tao, X. Bai, P. Rong, S. Yang, and T. Tian, “Interference mitigation for synthetic aperture radar based on deep residual network,” *Remote Sensing*, vol. 11, no. 14, p. 1654, 2019.
- [151] J. Rock, W. Roth, M. Toth, P. Meissner, and F. Pernkopf, “Quantized neural networks for radar interference mitigation,” *arXiv preprint arXiv:2011.12706*, 2020.
- [152] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.
- [153] Y. LeCun, “The MNIST database of handwritten digits,” <http://yann.lecun.com/exdb/mnist/>, 1998.

- [154] E. Kussul and T. Baidyk, “Improved method of handwritten digit recognition tested on MNIST database,” *Image and Vision Computing*, vol. 22, no. 12, pp. 971–981, 2004.
- [155] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, “Modulation classification based on signal constellation diagrams and deep learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 718–727, 2018.
- [156] Y. Wang, M. Liu, J. Yang, and G. Gui, “Data-driven deep learning for automatic modulation recognition in cognitive radios,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, 2019.
- [157] K. Karra, S. Kuzdeba, and J. Petersen, “Modulation recognition using hierarchical deep neural networks,” in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–3, IEEE, 2017.
- [158] C. Doersch, “Tutorial on variational autoencoders,” *arXiv preprint arXiv:1606.05908*, 2016.
- [159] N. Hubens, “Deep inside: Autoencoders,” Feb 2018.
- [160] F. Morawski, M. S. Bejger, E. Cuoco, and L. Petre, “Anomaly detection in gravitational waves data using convolutional autoencoders,” *Machine Learning: Science and Technology*, 2021.
- [161] J. Jordan, “Introduction to autoencoders,” Mar 2018.
- [162] L. Meng, S. Ding, and Y. Xue, “Research on denoising sparse autoencoder,” *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 5, pp. 1719–1729, 2017.

- [163] Y. Wang, H. Yao, and S. Zhao, “Auto-encoder based dimensionality reduction,” *Neurocomputing*, vol. 184, pp. 232–242, 2016.
- [164] C. Zhou and R. C. Paffenroth, “Anomaly detection with robust deep autoencoders,” in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 665–674, 2017.
- [165] O. Shcherbakov and V. Batishcheva, “Image inpainting based on stacked autoencoders,” in *Journal of Physics: Conference Series*, vol. 536, p. 012020, IOP Publishing, 2014.
- [166] F. Feng, X. Wang, and R. Li, “Cross-modal retrieval with correspondence autoencoder,” in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 7–16, 2014.
- [167] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, “EMNIST: Extending MNIST to handwritten letters,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, IEEE, 2017.
- [168] “Automotive radar interference mitigation data sets (ARIM).” <https://github.com/ristea/arim>. Accessed: 2021-12-08.
- [169] S. Ujan, N. Navidi, and R. Jr Landry, “An efficient radio frequency interference (RFI) recognition and characterization using end-to-end transfer learning,” *Applied Sciences*, vol. 10, no. 19, 2020.
- [170] J. Slavič, I. Simonovski, and M. Boltežar, “Damping identification using a continuous wavelet transform: application to real data,” *Journal of Sound and Vibration*, vol. 262, no. 2, pp. 291–307, 2003.

- [171] “CRAWDAD a community resource for archiving wireless data at dartmouth.” <https://crawdad.org/>.
- [172] M. Schmidt, D. Block, and U. Meier, “CRAWDAD dataset owl/interference (v. 2019-02-12).” Downloaded from <https://crawdad.org/owl/interference/20190212>, Feb. 2019.
- [173] M. Zhanikeev, “CRAWDAD dataset kyutech/interference (v. 2015-06-16).” Downloaded from <https://crawdad.org/kyutech/interference/20150616>, June 2015.
- [174] S. K. Kaul, I. Seskar, and M. Gruteser, “CRAWDAD dataset rutgers/noise (v. 2007-04-20).” Downloaded from <https://crawdad.org/rutgers/noise/20070420/RSSI>, Apr. 2007. traceset: RSSI.
- [175] S. Grimaldi, A. Mahmood, and M. Gidlund, “Real-time interference identification via supervised learning: Embedding coexistence awareness in IoT devices,” *IEEE Access*, vol. 7, pp. 835–850, 2018.
- [176] K. Ahmad, G. Shrestha, U. Meier, and H. Kwasnicka, “Neuro-fuzzy signal classifier (nfsc) for standard wireless technologies,” in *2010 7th International Symposium on Wireless Communication Systems*, pp. 616–620, IEEE, 2010.
- [177] D. Ciregan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3642–3649, IEEE, 2012.
- [178] Y. Xu, X. Qin, X. Xu, and J. Chen, “Open-set interference signal recognition using boundary samples: A hybrid approach,” in *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 269–274, IEEE, 2020.

- [179] Y. Kim, S. An, and J. So, “Identifying signal source using channel state information in wireless LANs,” in *2018 International Conference on Information Networking (ICOIN)*, pp. 616–621, IEEE, 2018.
- [180] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning*, pp. 448–456, 2015.
- [181] K. Youssef, L. Bouchard, K. Haigh, J. Silovsky, B. Thapa, and C. Vander Valk, “Machine learning approach to RF transmitter identification,” *IEEE Journal of Radio Frequency Identification*, vol. 2, no. 4, pp. 197–205, 2018.
- [182] M. Williamson, W. Headley, W. Clark, J. McCollum, T. Krauss, L. Lusk, D. Jenkins, T. Villemez, M. Moore, D. Jakubisin, *et al.*, “Multi-antenna pre-processing for improved RFML in congested spectral environments,” in *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 288–295, IEEE, 2021.
- [183] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, “Identifying wireless users via transmitter imperfections,” *IEEE Journal on selected areas in communications*, vol. 29, no. 7, pp. 1469–1479, 2011.
- [184] R. Polikar *et al.*, “The wavelet tutorial,” 1996.
- [185] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” *arXiv preprint arXiv:1506.00019*, 2015.
- [186] Y. Watanabe and K. Natsume, “Interference determination method and FMCW radar using the same,” Mar. 6 2007. US Patent 7,187,321.

- [187] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- [188] J. T. Connor, R. D. Martin, and L. E. Atlas, “Recurrent neural networks and robust time series prediction,” *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.
- [189] Y. Shi, K. Davaslioglu, Y. E. Sagduyu, W. C. Headley, M. Fowler, and G. Green, “Deep learning for RF signal classification in unknown and dynamic spectrum environments,” in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–10, IEEE, 2019.
- [190] B. Farhang-Boroujeny, “OFDM versus filter bank multicarrier,” *IEEE signal processing magazine*, vol. 28, no. 3, pp. 92–112, 2011.
- [191] H. Ghannam and I. Darwazeh, “SEFDM over satellite systems with advanced interference cancellation,” *IET Communications*, vol. 12, no. 1, pp. 59–66, 2018.
- [192] W. Zheng, J. Yao, and K. Wu, “Mitigating cross-technology interference in heterogeneous wireless networks based on deep learning,” in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 230–237, IEEE, 2020.
- [193] W. H. Clark IV, S. Hauser, W. C. Headley, and A. J. Michaels, “Training data augmentation for deep learning radio frequency systems,” *The Journal of Defense Modeling and Simulation*, vol. 18, no. 3, pp. 217–237, 2021.
- [194] M. Jeruchim, “Techniques for estimating the bit error rate in the simulation of digital communication systems,” *IEEE Journal on selected areas in communications*, vol. 2, no. 1, pp. 153–170, 1984.

- [195] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, “Stochastic geometry and random graphs for the analysis and design of wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 7, pp. 1029–1046, 2009.
- [196] S. Narkhede, “Understanding auc-roc curve,” *Towards Data Science*, vol. 26, pp. 220–227, 2018.
- [197] R. Schmogrow, B. Nebendahl, M. Winter, A. Josten, D. Hillerkuss, S. Koenig, J. Meyer, M. Dreschmann, M. Huebner, C. Koos, *et al.*, “Error vector magnitude as a performance measure for advanced modulation formats,” *IEEE Photonics Technology Letters*, vol. 24, no. 1, pp. 61–63, 2011.
- [198] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [199] F. Mireshghallah, M. Taram, P. Vepakomma, A. Singh, R. Raskar, and H. Esmaeilzadeh, “Privacy in deep learning: A survey,” *arXiv preprint arXiv:2004.12254*, 2020.
- [200] X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, and A. V. Vasilakos, “Privacy and security issues in deep learning: a survey,” *IEEE Access*, vol. 9, pp. 4566–4593, 2020.
- [201] R. LEE and I.-Y. CHEN, “The time complexity analysis of neural network model configurations,” in *2020 International Conference on Mathematics and Computers in Science and Engineering (MACISE)*, pp. 178–183, IEEE, 2020.
- [202] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, “Model complexity of deep learning: A survey,” *Knowledge and Information Systems*, vol. 63, no. 10, pp. 2585–2619, 2021.

- [203] A. Draganov, C. Brown, E. Mattei, C. Dalton, and J. Ranjit, “Open set recognition through unsupervised and class-distance learning,” in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, pp. 7–12, 2020.
- [204] E. Mattei, C. Dalton, A. Draganov, B. Marin, M. Tinston, G. Harrison, B. Smarrelli, and M. Harlacher, “Feature learning for enhanced security in the internet of things,” in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1–5, IEEE, 2019.
- [205] J. M. Tjensvold, “Comparison of the IEEE 802.11, 802.15. 1, 802.15. 4 and 802.15. 6 wireless standards,” in *IEEE: September*, vol. 18, 2007.
- [206] C. Gomez, J. Oller, and J. Paradells, “Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology,” *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.
- [207] D.-C. Chang and T.-H. Shiu, “Digital GFSK carrier synchronization,” in *APCCAS 2006-2006 IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1523–1526, IEEE, 2006.
- [208] T. Oyedare, V. K. Shah, D. J. Jakubisin, and J. H. Reed, “Keep it simple: Cnn model complexity studies for interference classification tasks,” in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2023.
- [209] F. Edalat, *Effect of power amplifier nonlinearity on system performance metric, bit-error-rate (BER)*. PhD thesis, Massachusetts Institute of Technology, 2003.
- [210] W. Wang, Y. Ning, H. Rangwala, and N. Ramakrishnan, “A multiple instance learning framework for identifying key sentences and detecting events,” in *Proceedings of the*

- 25th ACM International on Conference on Information and Knowledge Management*, pp. 509–518, 2016.
- [211] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [212] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [213] “CWT-based time-frequency analysis.” <https://www.mathworks.com/help/wavelet/examples/cwt-based-time-frequency-analysis.html>.
- [214] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, “Disturblabel: Regularizing CNN on the loss layer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4753–4762, 2016.
- [215] X. Ying, “An overview of overfitting and its solutions,” in *Journal of physics: Conference series*, vol. 1168, p. 022022, IOP Publishing, 2019.
- [216] N. Bitar, S. Muhammad, and H. H. Refai, “Wireless technology identification using deep convolutional neural networks,” in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*, pp. 1–6, IEEE, 2017.
- [217] “How much data is enough? predicting how accuracy varies with training data size.” <http://web.science.mq.edu.au/mjohnson/papers/Johnson17Power-talk.pdf>. Accessed: 2019-11-06.
- [218] S. Geman, E. Bienenstock, and R. Doursat, “Neural networks and the bias/variance dilemma,” *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

- [219] “Build a convolutional neural network using estimators.” <https://www.tensorflow.org/tutorials/estimators/cnn>. Accessed: 2019-11-06.
- [220] “Least-squares (model fitting) algorithms.” <https://www.mathworks.com>. Accessed: 2019-11-06.
- [221] D. C. Cireşan, U. Meier, and J. Schmidhuber, “Transfer learning for latin and chinese characters with deep neural networks,” in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, IEEE, 2012.
- [222] S. S. Haykin, S. S. Haykin, S. S. Haykin, K. Elektroingenieur, and S. S. Haykin, *Neural networks and learning machines*, vol. 3. Pearson education Upper Saddle River, 2009.
- [223] Y. S. Abu-Mostafa, “Machines that learn from hints,” *Scientific American*, vol. 272, no. 4, pp. 64–69, 1995.
- [224] D. Cireşan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” *arXiv preprint arXiv:1202.2745*, 2012.
- [225] P. Kyasanur and N. H. Vaidya, “Routing and interface assignment in multi-channel multi-interface wireless networks,” in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 4, pp. 2051–2056, IEEE, 2005.
- [226] M. Bal, “An industrial wireless sensor networks framework for production monitoring,” in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, pp. 1442–1447, IEEE, 2014.
- [227] C.-H. Chuang, K.-Y. Chang, C.-S. Huang, and T.-P. Jung, “Ic-u-net: a u-net-based denoising autoencoder using mixtures of independent components for automatic eeg artifact removal,” *NeuroImage*, vol. 263, p. 119586, 2022.

- [228] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.
- [229] H. Wu, Y. Qu, S. Lin, J. Zhou, R. Qiao, Z. Zhang, Y. Xie, and L. Ma, “Contrastive learning for compact single image dehazing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10551–10560, 2021.
- [230] W. El-Shafai, S. A. El-Nabi, E.-S. M. El-Rabaie, A. M. Ali, N. F. Soliman, A. D. Algarni, A. El-Samie, and E. Fathi, “Efficient deep-learning-based autoencoder denoising approach for medical image diagnosis.,” *Computers, Materials & Continua*, vol. 70, no. 3, 2022.
- [231] M. RLE, “Single-channel rf challenge,” URL <https://rfchallenge.mit.edu/challenge-1/>. Accessed, pp. 11–07, 2022.
- [232] T. Jayashankar, G. C. Lee, A. Lancho, A. Weiss, Y. Polyanskiy, and G. W. Wornell, “Score-based source separation with applications to digital communication signals,” *arXiv preprint arXiv:2306.14411*, 2023.
- [233] A. K. Dubey and V. Jain, “Comparative study of convolution neural network’s relu and leaky-relu activation functions,” in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018*, pp. 873–880, Springer, 2019.
- [234] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu, and T. Liu, “Understanding and improving early stopping for learning with noisy labels,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 24392–24403, 2021.

- [235] M. D. McKinley, K. A. Remley, M. Myslinski, J. S. Kenney, D. Schreurs, and B. Nauwe-laers, “Evm calculation for broadband modulated signals,” in *64th ARFTG Conf. Dig*, pp. 45–52, Orlando, 2004.
- [236] V. Meghdadi, “Ber calculation,” *Wireless Communications*, 2008.
- [237] G.-B. Huang, Q.-Y. Zhu, and C. K. Siew, “Real-time learning capability of neural networks,” *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 863–878, 2006.
- [238] C. Zhang, Y. Liu, and H. Fu, “Ae2-nets: Autoencoder in autoencoder networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2577–2585, 2019.
- [239] Z. C. Lipton, “The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery,” *Queue*, vol. 16, no. 3, pp. 31–57, 2018.