OPEN-SOURCE CLASSIFICATION SYSTEMS FOR FREQUENCY-DOMAIN RF SIGNALS: ROBUST PHYSICAL LAYER MULTI-SAMPLE RATE PROCESSING

Robert David Badger

Submitted to the faculty of the University Graduate School

in partial fulfillment of the requirements

for the degree

Doctor of Philosophy

in the Luddy School of Informatics, Computing, and Engineering,

Indiana University

June, 2022

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Minje Kim, PhD

Ariful Azad, PhD

Lei Jiang, PhD

Lantao Liu, PhD

05/16/2022

Acknowledgments

Thanks to Naval Surface Warfare Center (NSWC) Crane for the PhD fellowship opportunity that provided the funding and time to conduct research. Specifically within NSWC Crane, I would like to thank Dr. Jon Dilger and Dr. Andrew Christianson. They provided key mentoring to me before I started this program and continue to provide important insight and encouragement during the course of the program.

I would also like to thank my committee members Ariful Azad, Lei Jiang, and Lantao Liu for taking time away from their busy schedules to serve on my committee. Dr. Azad, and Dr. Liu also served as my professors at IU Luddy for coursework where their knowledge and ability to clearly explain complex ideas was instrumental in my academic growth and progress.

A special thanks to my Advisor Minje Kim; Without his endless patience, academic acumen, and continual encouragement throughout my research, success would have been much more difficult. Dr. Kim told me at the beginning of this program that he wanted me to have “fun”. That is not something that I normally attribute to higher learning, however, this program was the most enriching and “fun” that I can ever remember from academia. His dedication to the professional development of his students is unparalleled along with his uncompromising vision of his students becoming “independent researchers”.

Finally, I want to thank my wife Kelly Badger. Kelly was always encouraging and patient as I spent countless hours researching, writing, and code debugging. Without her love and support through this process, I would not have succeeded.

Robert David Badger

OPEN-SOURCE CLASSIFICATION SYSTEMS FOR FREQUENCY-DOMAIN RF SIGNALS: ROBUST PHYSICAL LAYER MULTI-SAMPLE RATE PROCESSING

Digital signal processing (DSP) is widely used for digitized communication data and statistical signal processing (SSP) techniques are often applied to stochastic communication processes. DSP and SSP methods are also routinely used for modern radio frequency (RF) communication data equipment but hardware is unable to make intelligent processing decisions required for modern communication systems and data. The next evolvement of RF processing is intelligent signal processing (ISP), in which data uses machine learning (ML) models to provide superior processing benefits over standalone DSP and SSP systems. In this dissertation, informatics techniques are investigated to process software defined radio (SDR) open-source RF data sets in the frequency domain through the use of the singular value decomposition (SVD) algorithm. This algorithm reduces the dimension of the time-frequency representation of the IQ samples, forming a low-rank approximation of the original, that is then converted back to an RF data signal that properly activates a matched transceiver. This leads to a novel frequency domain approach that facilitates ISP to classify RF signals. The experimental results show that using all of the frequency domain data can achieve better performance than a frequency domain magnitude-only approach. Additional open-source RF datasets collected at various sample rates expanded RF classification across multiple frequency bandwidths. Next, multiple sample rate datasets are collected from multiple SDR hardware to classify waveforms from additional sample rates. Additionally, the usable RF spectrum can be dense with signals, and I demonstrate how multiple waveforms operating in a single RF sample can be properly classified. Finally, I investigate how multiple types of SDR hardware may be necessary to overcome phase noise differences that affect model efficacy.

Minje Kim, PhD

Ariful Azad, PhD

Lei Jiang, PhD

Lantao Liu, PhD

Contents

Acknowledgments **iii**

Abstract **iv**

1 Introduction **1**

1.1 Digital signal processing (DSP) and statistical signal processing (SSP) 2

1.2 Constellation diagrams 2

1.3 DSP and SSP implementation 3

1.4 Intelligent signal processing (ISP) 4

2 IQ data Singular Value Decomposition (SVD) **5**

2.1 Introduction 5

2.2 Background 8

2.3 Data Preparation: SDR and SigMF 9

2.4 STFT and SVD for Compression 13

2.5 Experimental Results 15

2.6 Analog waveforms: PTT radio 15

2.7 Digital waveforms 18

2.8 Signal reconstruction efficacy for voice communication 19

2.9 Compression Analysis 19

2.10 SVD summary 20

3 TF domain RF classification: Single sample rate/single SDR **21**

3.1 Introduction 21

3.2 Motivation 22

3.2.1 Time division multiple access (TDMA) example 24
3.3 RF Domain 24
3.4 Data Collection and Labeling 28
3.5 The Time-Frequency Representation 30
3.6 The Proposed RF Classification Pipeline 34
3.6.1 Experimental Setup 34
3.6.2 GUI-based patching 34
3.6.3 RF Signal Classes 35
3.6.4 Test-Time Inference 39
3.7 Discussion 40
3.8 TF domain classification summary 44
4 TF domain RF classification: Multi-sample rate/Multi-SDR 45
4.1 Related work 46
4.2 SDRs and format 47
4.3 RF signal isolation 48
4.3.1 Test setup 48
4.3.2 Test transceivers 49
4.4 Methods and Procedures 52
4.5 Patching 52
4.5.1 Computing Environment 53
4.6 Classification Methods (CNNs) 53
4.6.1 CNN Model 53
4.7 The basic training process with a multi-sampling rate treatment 54
4.8 Data augmentation 54
4.9 Multi-class patch treatment 55

4.9.1 Dataloader modification 55

4.9.2 CNN modification 55

4.9.3 Combining test files and labels 56

4.10 Experimental Setup 57

4.10.1 Data collection process and format 57

4.11 Experimental Results 58

4.11.1 Cross-sample-rate testing: single class per patch 58

4.11.2 Multi-class per patch 61

4.11.3 Waveform frequency occupation 65

4.12 Chapter Conclusion 66

5 IQ Phase investigation **73**

5.1 Phase test setup 73

5.2 N210 74

5.3 B210 74

5.4 B205 76

5.5 SDR phase conclusion 82

6 Conclusion **83**

Bibliography **83**

Curriculum Vitae

Chapter 1

Introduction

The radio frequency (RF) spectrum is a subset of the electromagnetic spectrum and spans approximately $30\,kHz$ to $300\,GHz$. Communication within the RF spectrum consists of hundreds of variations of modulations (analog and digital) and channel bandwidths. RF channel bandwidths range from $3\,kHz$ for narrow frequency modulation (FM) analog waveforms to hundreds of MHz for fifth generation (5G) cellular networks. These large differences in operating frequencies and bandwidths illustrate a diverse domain that is complicated to operate in and manage.

The RF spectrum is predominately managed through the use of specific allocation bands [1]. This methodology provides spectrum housekeeping for many types of RF transmissions, including cellular networks, television stations, radar, and military operating bands, but does not address many of the RF traffic issues in the industrial, scientific, and medical (ISM) bands [2] and dynamic spectrum sharing bands (DSS) [3]. ISM bands are the de facto operating bands for multitudes of new RF equipment, including internet-of-things devices, drones, and wireless communication networks (Wi-Fi) that vie for spectrum access. Since ISM bands do not require a government license to operate in, any transmission equipment operating in these bands is required to endure interference from other waveforms. Additionally, there are no regulatory protection from other transmitters operating in the same bands.

In addition to ISM bands, DSS allocation bands require cognitive radio to streamline traffic. DSS bands are currently facilitating mobile network operators to roll out new radio services [4] with cognitive radio playing an important role [5] [6], while other DSS allocation bands are shared by multi-tiered users, such as [7] [8]. The "Wild West" of ISM bands and DSS bands are chaotic without assistance from cognitive radio and results in substantial co-channel RF interference.

In this chapter, I introduce an intelligent system processing (ISP) system that helps optimize band usage, reduce co-channel interference, and promotes additional clear channel RF "real estate" for devices seeking spectrum access.

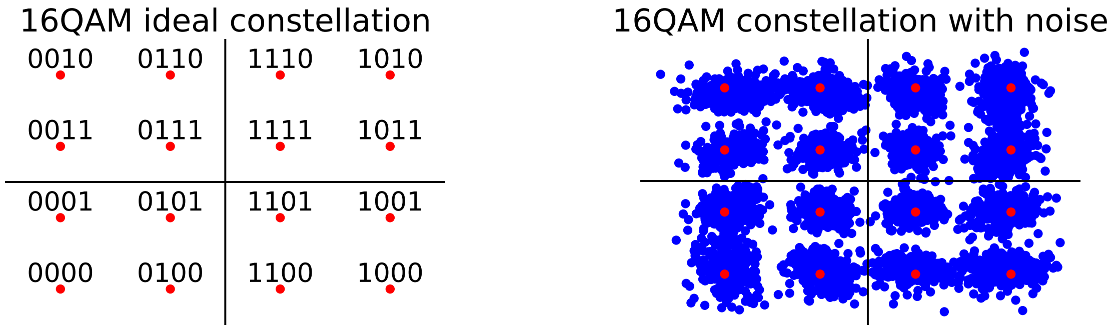
1.1 Digital signal processing (DSP) and statistical signal processing (SSP)

RF transceivers depend on DSP and SSP methods to process RF data in modern communications systems. These methods provide a traditional approach to some data processing tasks, including robust noise mitigation. Analog RF processing can consist of purpose-built integrated circuits that perform functions such as filtering and amplification on streamed RF signals directly from the initial RF chain (e.g., antennas, RF mixers), or circuits constructed from discrete components. As RF transceiver equipment historically advanced to the utilization of analog to digital converters (ADC) and digital to analog converters (DAC), DSP hardware performed the RF processing functions digitally.

1.2 Constellation diagrams

In RF digital modulation, constellation diagrams are often used to depict data symbols in the xy-plane. By custom, the x-axis is the real or in-phase plane and the y-axis is the imaginary or quadrature plane. The constellation symbols are populated to this plane via plotting magnitude and phase. Together, the in-phase and quadrature (IQ) plane can depict data from RF samples that produce a scatter plot. Constellation diagrams are an invaluable tool that provides insight into RF data symbol locations within the "constellation".

Fig. 1.1 depicts two 16 quadrature amplitude modulation (QAM) RF waveform constellations: In fig. 1.1(a), the QAM scheme uses 16 symbols which are arranged around the origin of the IQ-plane. Each constellation point represents 4 bits and is arranged using gray code [9]. RF constellations can have many symbols (e.g., 5G carriers can use quadrature phase shift keying (QPSK), 16QAM, 64QAM or 256QAM) and geometry. Ideally, a constellation can have infinite



(a) Ideal

(b) Noisey

Figure 1.1: 16QAM constellations

symbols, because data points are infinitesimally small within the constellation. In reality, there are many constraints that limit the number of data symbols within a given RF constellation. For example, Fig. 1.1(b) illustrates various sources of noise causing constellation symbol locations to expand and overlap with neighboring symbols. This can cause errors when a symbol is misidentified during demodulation.

1.3 DSP and SSP implementation

SSP methods take advantage of stochastic processes in the RF data streams and can be used to help correctly determine digital encoded data that may be difficult to determine due to noise and channel impairments (e.g., direct current (DC) impairments, IQ imbalance, phase mismatch from local oscillators). As errors increase, the error vector magnitude between the ideal symbol location and actual location becomes greater. SSP uses statistics from the data to help determine a samples most probable digital symbol. Additionally, DSP and SSP techniques can be substantiated into software via software defined radios (SDRs), this provides increased flexibility over hardware only solutions.

1.4 Intelligent signal processing (ISP)

In this dissertation, I propose an ISP system that will help optimize band usage, reduce co-channel interference, and promote additional clear channel RF "real estate" for devices seeking spectrum access. DSP and SSP methods are often used in conjunction to process data in modern communications systems, but they do not learn from data to provide insightful decisions. ISP enabled RF hardware can determine parametrics from live emitters currently operating in channels of interest. These learned parameters can then be used to re-optimize transceiver functions before accessing the spectrum. An ISP data driven approach will augment and provide significant improvement to current DSP and SSP capabilities.

Together, this "family" of signal processing techniques will fundamentally change the way hardware interacts with the spectrum. However, in order to harness ISP for use in the RF domain, I first consider how to best analyze RF IQ data and transform that data to a domain where informatics techniques will be used to uniquely process IQ data.

Chapter 2

IQ data Singular Value Decomposition (SVD)

This chapter contains material that was published in the 29th European Signal Processing Conference [10]

2.1 Introduction

In this chapter, I will show how to transform the RF data to a domain where informatics approaches are used to manipulate the waveforms. This is key for all subsequent chapters and lays the foundation for how I provide robust RF frequency domain classification.

Monitoring and collecting Radio Frequency (RF) data is an important part of spectrum management as RF electromagnetic waves permeate almost every facet of our lives. Cellular telephones, the internet, radio stations, satellites, and garage door openers will all access the RF spectrum, at some point, to transmit and receive information.

The software defined radio (SDR) [11] [12] [13] [14] is often used to collect RF data; it provides a standardized method for RF data collection that will facilitate ISP use. SDR receivers range from as little as twenty dollars (U.S.) to thousands of dollars, thereby providing ample price entry points suitable for many RF applications. RF data is sampled in the time domain, and the rate at which spectrum samples are collected is often adjusted by the user.

SDRs are widely used to collect RF signals, real-time spectrum monitoring, and as a software transceivers for RF communication. SDR hardware is simplified compared to specific application transceivers because SDR hardware usually only samples the spectrum or transmits IQ formatted data, typically without any additional hardware processing. The IQ stream is transferred to a host computer where software can store the IQ data or function as a transceiver for signal processing [15][16].

Existing methods for IQ compression are primarily used within RF communication equipment that is transferring the IQ data on internal buses. This type of IQ compression reduces bus bandwidth requirements through decimation/down-sampling, and is often in the time domain with a typical compression of four [17]. This is different from research presented here in that my proposed compression method is defined in the frequency domain and works in an offline manner, thus resulting in a significantly greater compression ratio.

My methods works on the time-frequency (TF) representation of an originally IQ formatted RF data. For a given discrete-time waveform signal,

$$x(n) = \sin(2\pi f n + \phi(n)), \tag{2.1}$$

where f, n and $\phi(n)$ are for the frequency and time indices, and time-varying phase shifts, respectively. Then, trigonometry can represent (2.1) as the sum of the orthogonal components:

$$\text{In-Phase} \quad i(n) = \sin(2\pi f n)\cos(\phi(n)) \tag{2.2}$$

$$\text{Quadrature} \quad q(n) = \sin\left(2\pi f n + \frac{\pi}{2}\right)\sin(\phi(n)). \tag{2.3}$$

While collecting and archiving RF data have many uses, IQ samples can generate very large files, depending on bandwidth and duration of collection. For example, IQ sample files can range from 40MB at 1 mega samples per second (msps) to over $1\,GB$ at $25\,msps$ for each collect of five seconds.

My goal is to compress this kind of data without loss of critical information to actuate the matched receiver. One way to achieve the reduction of data rate is to quantize the signal using a low bit-depth. In [18], the TF domain is sub-divided into multiple groups of subbands and quantized separately. Coupled with a thresholding technique that suppresses lower magnitudes, this method achieves a reasonable compression ratio. However, my proposed low-rank approximation-based approach is orthogonal to the quantization mechanism, making the two approaches complementary to each other.

Meanwhile, in [19], robust principal component analysis (RPCA) is used to remove unwanted interference from the RF signal in the TF domain. Given that RPCA is based on the low-rank approximation of the signals, this approach shares the same philosophical approach as mine. However, its experimental design is focused on interference suppression than compression performance. My work presents a singular vector decomposition (SVD)-based algorithm [20] to substitute the time-frequency representation of the IQ data with a low-rank approximation. I also present the compression performance on various analog and digital RF signals.

Additionally, I investigate the use of graphics processing units (GPU) for the time-frequency transformation and SVD implementations to see if parallel processing can expedite those processes. The SVD algorithm is fully supported in PyTorch version 1.6 to assist with this effort.

There are two different cases of reconstructed IQ data to consider: the analog and digital cases. First, if reconstruction can actuate the matched analog push-to-talk (PTT) receiver without any perceivable degradation in speech quality, it will be considered a *good* reconstruction, and vice versa. I will discuss more about the subjectivity of speech quality measurements in Section 2.8. The second case is within digital transceivers, where the bit error rate (BER) can monitor the reconstruction efficacy. However, the procedure proposed by Mitra [21] is not feasible for us due to the lack of knowledge with respect to the transceiver's physical layer architecture, schematic diagram of signal path, and equipment to measure BER. Instead, I assume that if the reconstructed signal can reliably actuate the matched transceiver with a high probability, the internal BER at the receiver must be lower than maximum allowable value. Further empirical measures of the reconstructed IQ signals are discussed in Section 2.7.

I claim that this work is the first attempt to compress RF signals using the well-defined low-rank approximation concept, achieving a sensible compression ratio to the best of my knowledge. The lack of a compression method in the literature may be due to the communication-oriented nature of the RF signal processing domain, in which RF signals are not considered as stored data to be compressed, although I have been motivated by this problem for informatics exploration purposes. While SVD itself has been widely studied in the literature, I fully utilize it in the time-frequency domain. Eventually, one can control the trade-off between the compression ratio and reconstruction quality based on the orthogonality of SVD.

2.2 Background

I convert the IQ data into the time-frequency domain with a short-time Fourier transform (STFT) function [22]. The discrete STFT of a discrete time signal $x(n)$ with a symmetric, typically bell-shaped window function $\omega(t)$ of length H (i.e., the number of consecutive non-zero elements) is defined as:

$$\text{STFT}\{x(n)\}(f,t) \equiv \boldsymbol{X}_{f,t} = \sum_{n=-\infty}^{\infty} x(n)\omega(n-tR)e^{-j2\pi fn}, \tag{2.4}$$

where t and R denote the frame index and hop size, respectively. STFT results in a complex matrix $\boldsymbol{X}$. I expand this operation to both I and Q channels, to convert the time-domain signals $i(n)$ and $q(n)$ into $\boldsymbol{I}$ and $\boldsymbol{Q} \in \mathbb{C}^{F\times T}$, where F and T are the number of frequency subbands and frames, respectively. The size of this complex matrix is defined by the bandwidth of the signal as well as the frame rate, i.e., the hop size between the windowed frames, which I set to be 50% of the frame size. Inverse of the STFT (iSTFT) is defined by applying inverse DFT to each spectrum $\boldsymbol{X}_{:,t}$, which recovers the windowed time domain signal $x(n)\omega(n-tR)$. An overlap-and-add process follows to recover the original $x(n)$ based on the constant overlap-and-add property of the window function.

Meanwhile, SVD factorizes a given matrix $A \in \mathbb{R}^{M\times N}$ into three factor matrices, $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ hold orthonormal basis vectors whose significance is represented by the corresponding singular values, stored in the diagonal matrix $\boldsymbol{\Sigma}$. SVD has been used for compressing images for over forty years [23, 24]. The compression relies on the assumption that the rank K of the input matrix $\boldsymbol{A}$ is lower than its dimensions, i.e., $K < M, N$, which is defined as the number of non-zero singular values. In practice, as I do not know the exact rank due to the noisy nature of the data, one can choose a reasonable rank $L < M, N$ with a risk of reconstructing a rank-deficient version of the input:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top} \approx \hat{\boldsymbol{A}}^{(L)} = \sum_{k=1}^{L} \boldsymbol{U}_{:,k}\boldsymbol{\Sigma}_{k,k}\boldsymbol{V}_{:,k}^{\top}. \tag{2.5}$$

These applications are similar to my approach in the sense that STFT converts the IQ stream into arrays. For example, by taking the magnitude of the STFT, I can visualize the TF representation as an image as shown in the figures in this paper. However, my method applies SVD to all real and imaginary coefficients instead of the magnitudes, in order to re-constitute them into IQ data without losing the phase information.

2.3 Data Preparation: SDR and SigMF

SDR hardware:

The basic SDR hardware is depicted in Fig. 2.1. This type of receiver is called a direct conversion receiver because it does not have an intermediate frequency. The RF signal is input to both mixers; the top path is mixed with the local oscillator as is the bottom path. However, the bottom path mixes the RF signal with a phase shift to the local oscillator. After the mixers, both paths go through further filtering, resulting in I and Q data streams. Down stream of the low pass filters, the IQ paths are sampled and digitized.

The sample rate of an SDR is customarily expressed in mega samples per second ($msps$), i.e., 10^6 samples per second $\equiv 1\, msps$. This also defines the frequency bandwidth (e.g., $1\, msps$ provides $1\, MHz$ of bandwidth, $25 msps$ provides $25\, MHz$ of bandwidth due to SDR design, and allows for

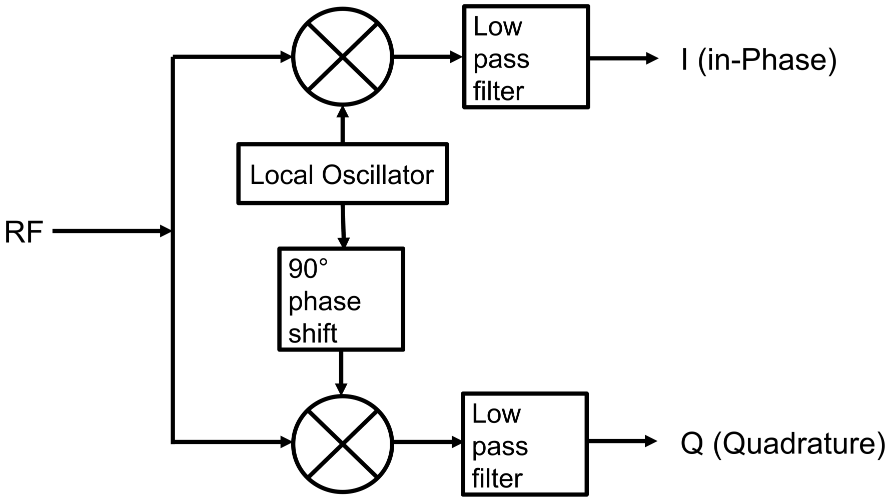


Figure 2.1: Basic SDR quadrature receiver

narrow and wide swaths of spectrum to be sampled through changing the local oscillator clock rate, decimation factor, and filtering.

Primarily three SDRs were used throughout this research and are depicted in Fig. 2.2.

GNU Radio (`gnuradio`) [25] is an open-source software library that provides a graphical environment for creating RF signal processing blocks to build RF "hardware" in software. This software is then used to control a SDR to operate in the RF spectrum.

This work extensively uses `gnuradio` to collect RF waveforms for analysis and transformation. Within `gnuradio`, an additional out-of-tree (OOT) module is used during signal collection. This module facilitates the collection of IQ data as well as labeled metadata in the format compatible to `SigMF` [26] [27] [28]. All data in this chapter is collected at the sampling rate of $1\,msps$ producing a bandwidth of $1\,MHz$. The signal duration is $5\,seconds$ for the results section 2.6, 2.7, and 2.9 and $15\,seconds$ for the efficacy calculations section 2.8.

The transceivers operate at or around the $433\,MHz$, ISM band. IQ data is recorded using an Ettus N210 SDR with SBX daughter board. Data is transferred over-the-wire between the SDR and CPU as a complex 16-bit integer [29]. The CPU converts the integer data to 32-bit float for use in `gnuradio` applications.

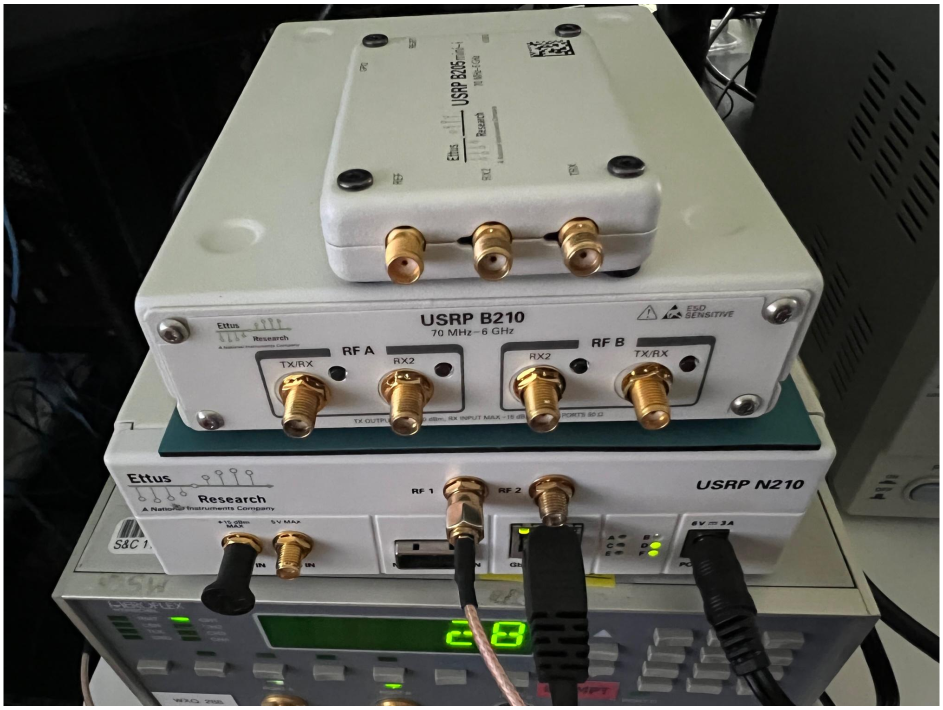


Figure 2.2: SDR hardware

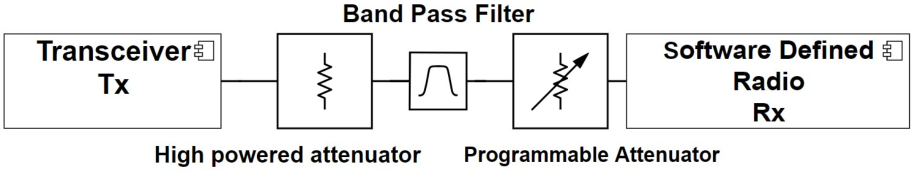


Figure 2.3: The closed loop coaxial setup.

Two methods are used to connect the transmitted waveform to the matched receiver: closed-loop and OTA. The former setup minimizes interference from outside sources. OTA, on the other hand, relies on the antennas, leading to a higher probability of signals of non-interest getting recorded.

Closed loop setup: The closed loop setup connects the Tx directly to the Rx through a coaxial network. The basic closed loop and OTA configuration consists of a high powered $30\,dB$ attenuator inserted from the Tx output, in-line with an $8\,MHz$ band-pass filter centered at $433\,MHz$. It is followed by a Mini-Circuits' VLM-33-2W+ coaxial RF limiter to protect the programmable attenuator and SDR from high-powered signals. The final component before the input to the programmable attenuator is a Mini-Circuits' BLK89-S+ DC-Block, to prevent DC power getting

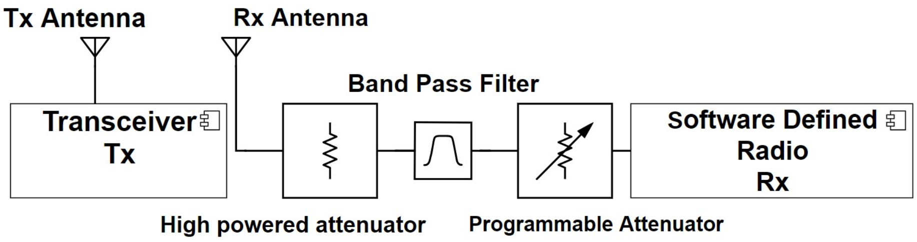


Figure 2.4: Over the air coaxial setup

into the front end of the attenuator and SDR. All transmitters are properly attenuated to ensure high signal-to-noise ratio (SNR) at the receiver, while also ensuring recorded signals stayed in the linear region of the receiver. Fig. 2.3 depicts a simplified schematic of the RF hardware setup for closed loop collection.

OTA setup: OTA collection was performed for transceivers that do not have a removable antenna or are not easily configurable for closed-loop setup. Fig. 2.4 illustrates a simplified schematic of the OTA setup. The primary difference between the two configurations is the use of antennas for waveform propagation. The OTA method featured Tx and Rx antennas that are optimized for the operating frequencies. The physical distance between the Tx and the Rx antennas was 2λ ($\lambda = wavelength$) or more to ensure the Rx is in the far field when recording IQ signals. An $8\,MHz$ bandpass filter is also applied to minimize extraneous signals.

SDR transmission: Transmitting a recorded IQ file via SDR will actuate a matched receiver, as if it had been sent by the original transmitter with minimal added artifacts. However, I feed the IQ data to my compression and decompression pipeline, i.e., STFT—low rank SVD—iSTFT, which replaces the original IQ data with a low rank approximation. The SVD is performed on a computer using the saved IQ data, then a new IQ file is reconstructed from the reduced rank matrix. Fig. 2.4 depicts the typical transceiver to SDR setup. When recording IQ files, the path is from the transceiver to the SDR. After reconstructing a waveform, the transmission path is from the SDR to the transceiver.

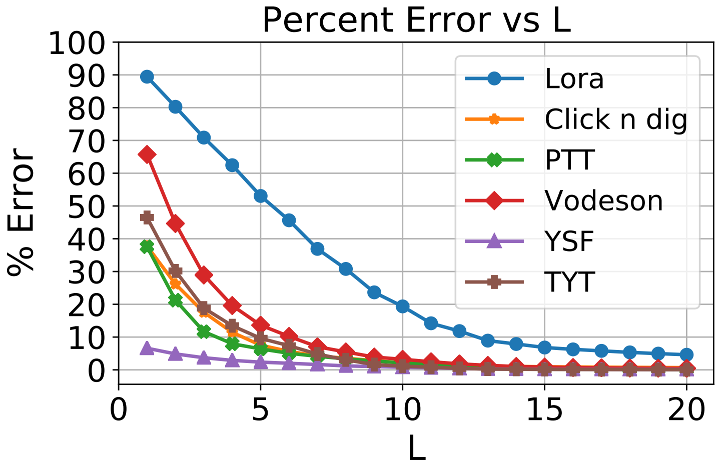


Figure 2.5: The reconstruction percent error by varying choices of L.

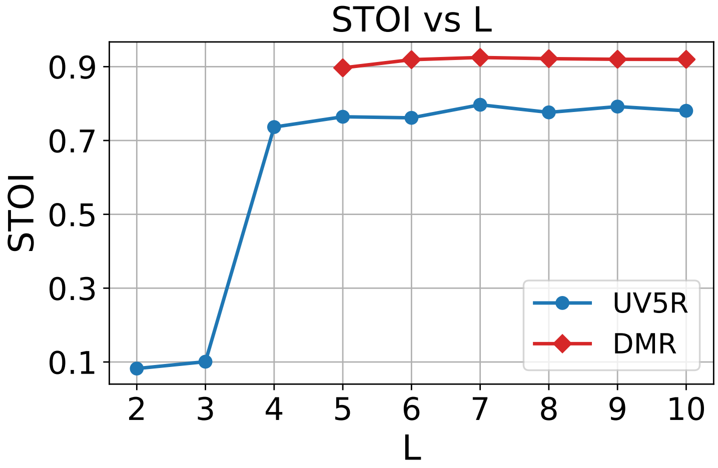


Figure 2.6: The speech intelligibility potential of the low rank approximation.

The author made the IQ datasets and the Python code used to analyze the data publicly available[1].

2.4 STFT and SVD for Compression

An EVGA® GTX1080Ti (3584 NVIDIA® CUDA® cores) GPU with PyTorch 1.6 is used to perform the STFT and iSTFT algorithms (`torch.stft` and `torch.istft`). While CPU implementations are also an option, both algorithms are, on average, 12.5 times faster on the GPU.

---

[1] https://saige.sice.indiana.edu/research-projects/rf-svd

Table 2.1: The rank values that meets certain percent error thresholds, 5% and 1% error.

Rank (Percent Error)	L ($<$ 5%)	L ($<$ 1%)
PTT	7 (4.13%)	14 (0.78%)
TYT®	7 (4.82%)	11 (0.80%)
Vodeson	9 (3.90%)	15 (0.84%)
Click 'n Dig®	7 (4.23%)	15 (0.95%)
YSF	2 (4.86%)	10 (0.79%)
LoRa™	19 (4.83%)	104 (0.99%)

STFT and iSTFT uses a frame size of 1024 samples and a 50% of overlap. Each frame is windowed by a Hann window of the same size. The frequency dimension is $F = 513$ (due to the complex conjugacy, I discard about half of the frequency dimensions), resulting in the frequency resolution at $976.56\,Hz$. This granularity is chosen due to the amount of GPU memory required to perform matrix operations, while it was sufficient to activate matched receivers during the low rank SVD reconstitution. The number of frames T varies depending on the length of the signal. I apply STFT to each of the I and Q channels, respectively, which results in four real-valued spectrograms, $\boldsymbol{I}^{(\text{Real})}$, $\boldsymbol{I}^{(\text{Imag})}$, $\boldsymbol{Q}^{(\text{Real})}$, and $\boldsymbol{Q}^{(\text{Imag})}$:

$$\text{STFT}\{i(n)\} \equiv \boldsymbol{I}^{(\text{Real})} + j\boldsymbol{I}^{(\text{Imag})} \tag{2.6}$$

$$\text{STFT}\{q(n)\} \equiv \boldsymbol{Q}^{(\text{Real})} + j\boldsymbol{Q}^{(\text{Imag})} \tag{2.7}$$

PyTorch implementation of the SVD algorithm is applied to each of the four spectrograms, respectively. As my compression ratio depends on the choice of rank L, I perform a simple test to choose a small enough, but working value. SVD is then performed on select recorded IQ waveforms and all selected IQ files are verified to properly actuate the associated receiver using SDR IQ playback. After factorization occurs, each waveform is reconstructed with a varying amount of singular vectors L. Once the waveform is reconstructed via iSTFT, it is re-transmitted OTA and tested to see if the matched receiver could successfully demodulate the reconstructed signal. Fig.

2.5 depicts the relationship between the chosen rank L and the Euclidean norm ratio in per cent error, $\mathcal{E}(L) = \big(\sum_{f,t}\big(\boldsymbol{A}_{ft} - \hat{\boldsymbol{A}}_{ft}^{(L)}\big)^2\big)^{1/2}\Big/\big(\sum_{f,t}\boldsymbol{A}_{ft}^2\big)^{1/2}$.

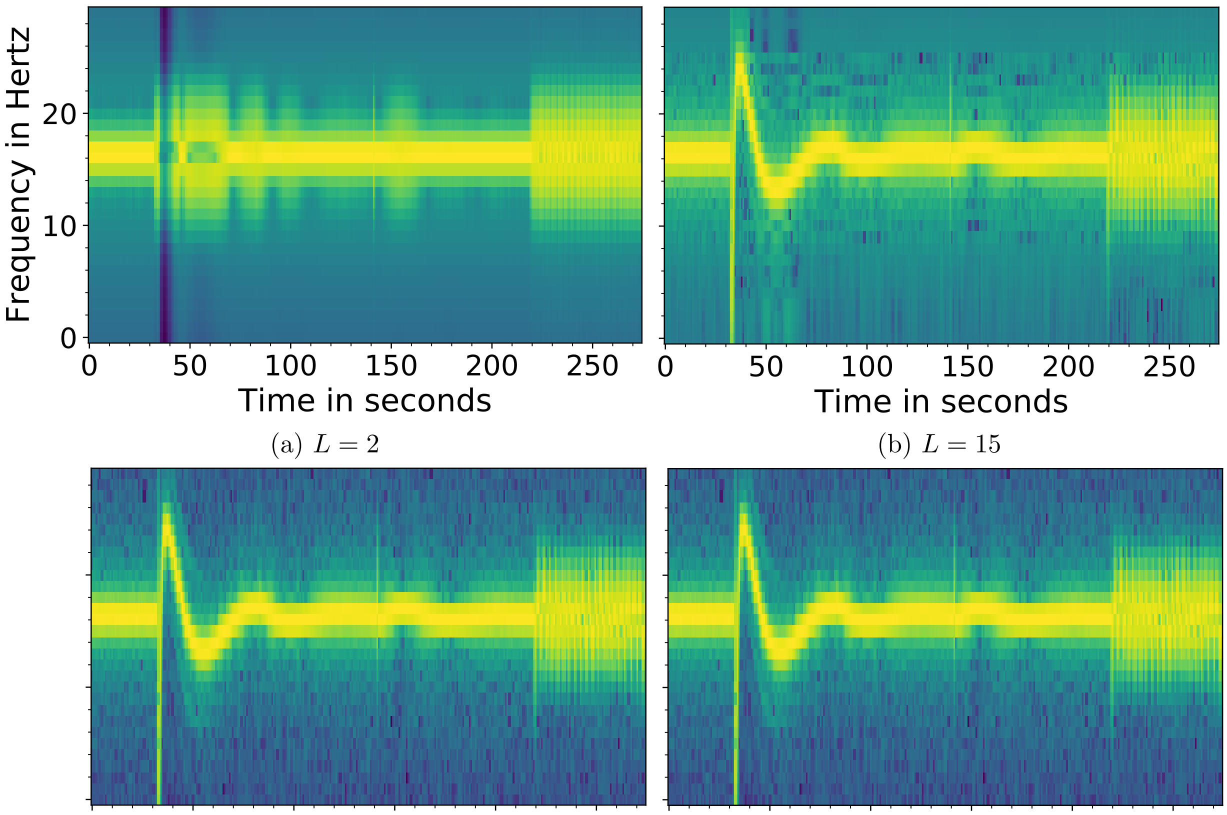
2.5 Experimental Results

My experiments are designed to empirically verify that the proposed low-rank approximation can actuate the receiver with a significantly lower amount of bits. However, note that its data-driven nature limits my ability to prove the performance guarantee. My fundamental assumption is that the rank of the RF signals' time-frequency representation may be lower than their actual dimensions as in many other signal domains (e.g., audio and video), while the actual working rank can be found only empirically. My SVD-based approach supports this search, as SVD's orthogonality lets the user incrementally add additional latent variables until a successful reconstruction quality is achieved (the "elbow" method shown in Fig. 2.5).

2.6 Analog waveforms: PTT radio

The Baofeng UV5R analog PTT radios use frequency modulation (FM) and are relatively easy to reproduce from a low rank approximation due to receiver design. Simply presenting enough in-channel energy to break squelch, the analog receiver chain attempts to demodulate the signal.

$L=3$ is the minimum rank to actuate the PTT receiver and be intelligible, while it introduces noticeable audio artifact. With four vectors, the sound quality is perceptually indistinguishable from the uncompressed original. Corresponding ranks for 5% and 1% errors are $L = 6$ and $L = 15$, respectively. Fig. 2.7 zooms into the details of spectral changes as L increases. Note that figures 2.7 and 2.8 are frequency components in the Y-axis, as FFT bins. I can see that the full rank approximation recovers most of the details, while $L = 15$ also captures an important structure. A more detailed analysis on the audio quality is discussed in Section 2.8.



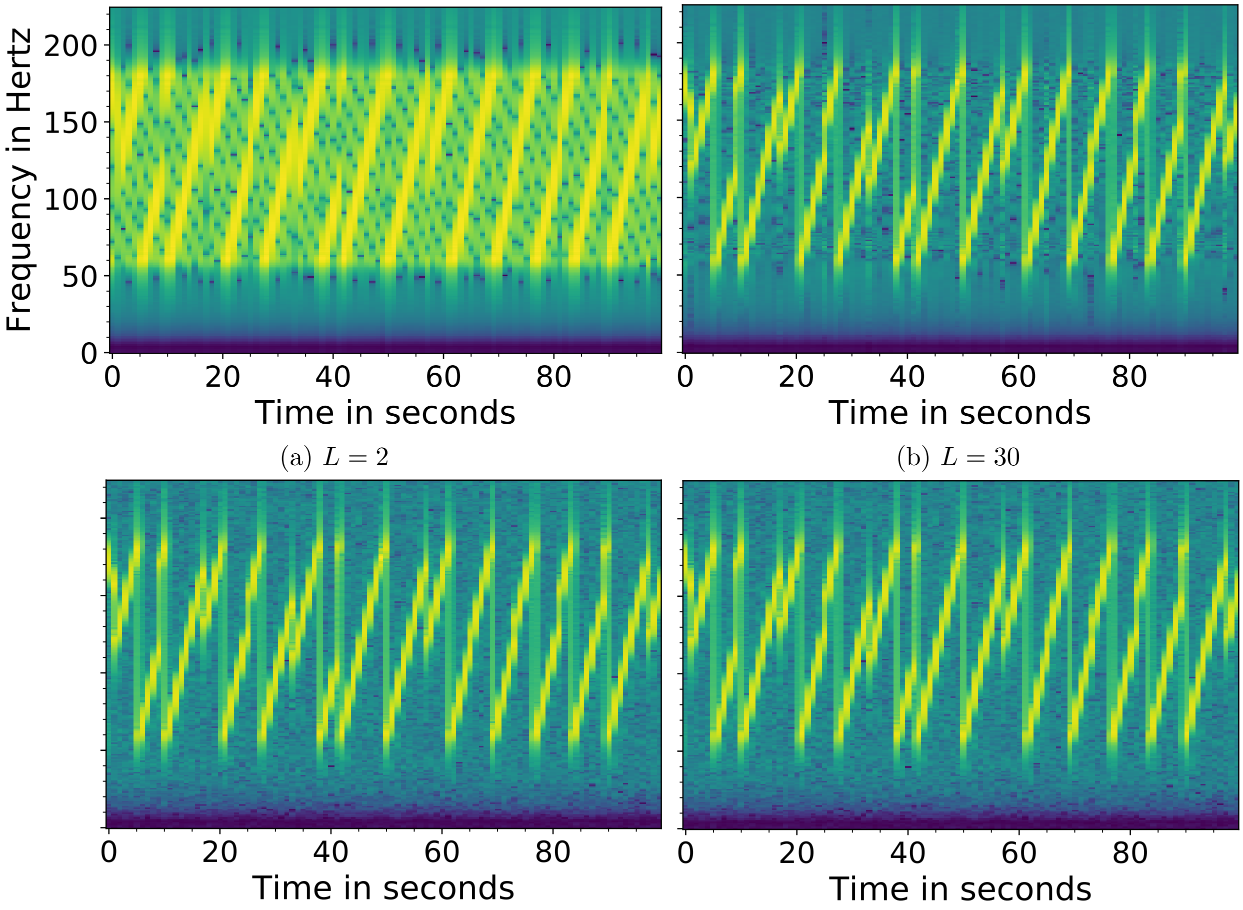
(a) $L = 2$

(b) $L = 15$

(c) $L = 1024$

(d) Original

Figure 2.7: The reconstructed PTT spectrograms with varying ranks.



(a) $L = 2$

(b) $L = 30$

(c) $L = 1024$

(d) Original

Figure 2.8: The reconstructed LoRa$^{\text{TM}}$ spectrograms with varying ranks.

2.7 Digital waveforms

Five different transceiver models are investigated for the digital setup:

- TYT® DMR radio: A digital mobile radio (DMR) that uses four-frequency shift keying (4FSK) modulation and time division multiple access channels.

- Vodeson HD03 Doorbell: Operates in the 434, MHz band.

- Click 'n Dig® D2 FOB: Operates in the 434, MHz band.

- Yaesu System Fusion (YSF) C4FM: YSF has four modes of operation, one analog and three digital C4FM modes. The results published here are the voice full-rate mode [30]. In this mode the entire 12.5 kHz bandwidth is used.

- LoRa™ FMCW or CSS: It is a frequency modulated continuous wave (FMCW) / chirp spread spectrum (CSS) waveform [31]. The tested configuration for LoRa™ is a HopeRF RFM98 module [32] with a spreading factor of 9 and bandwidth of 125 kHz.

In Fig. 2.5 and Table 2.1 I can see that significantly small values for L, less than 10 most of the time, can recover 95%of the original matrix in terms of the Euclidean norm ratio. For a higher reconstruction quality of 99%, more ranks are required, but most of the time this does not exceed 15. Meanwhile, LoRa™ is the most difficult case that requires substantially higher ranks, which is due to the granularity needed for the receiver to discern where the CSS begins and ends in the time-frequency domain shown in Fig. 2.8.

A significant difference that distinguishes digital modulation from analog (i.e., PTT transceivers) is that analog receivers will generally break squelch and activate the receiver; if the signal has enough of the proper modulation characteristics, then poor to excellent communication can be achieved. On the other hand, in order to activate digital receivers, enough packets or symbols have to be received and properly demodulated. This means that the actuation of receivers can indirectly determine the quality of the reconstruction, which is generally the case for all digital modulation experiments. All modulations were initiated with a small L; the minimum is reported in Table 2.2.

2.8 Signal reconstruction efficacy for voice communication

I employ short-time objective intelligibility (STOI) measures [33] to empirically determine the quality of received IQ voice transmissions for select analog and digital PTT waveforms. A ten second voice file from the LibriSpeech corpus [34] is played from a speaker, which the Tx unit captures. The RF waveform is captured using the OTA setup depicted in Fig. 2.4. After performing SVD, the recovered IQ signal is transmitted to the original receiver, which extracts the speech signal and plays it using its own speaker.

The audio recording process creates some additional artifact, which I minimize by keeping the microphone and PTT speaker's physical orientation constant. The received signal is temporally aligned to the playback of the Rx that did not go through the compression process, by using their cross-correlation. The STOI measurements in Fig. 2.6 for the Baofeng UV5R analog PTT transceiver coincides with listening to the audio signals: after about five vectors, it is hard to hear an increase in audio quality. The digital TYT® MD-380 DMR waveform is reconstituted and triggers the matched receiver with as few as five vectors. The sound quality is digitally clear and no improvement was audible as L increased, which explains the STOI values that saturate at $L = 5$.

2.9 Compression Analysis

Actual compression ratios are calculated based on original IQ data size divided by the sum of the saved arrays' file size. For a given windowed frame of size H, FFT gives $2H$ real and imaginary coefficients; half of them will be kept due to the complex conjugacy. Since the 50% overlap doubles the frame rate, STFT ends up with twice more coefficients, i.e., $FT = 2J$ for a signal with J samples. Meanwhile, when $L < F, T$, the compression ratio is $(FT/2)/(FL + L + LT)$, which is in the order of $\mathcal{O}\big((FT)/(F+T)\big)$ when $L \ll F, T$.

Table 2.2: Compression analysis. Minimum L indicates the minimum rank required to actuate the receiver.

Modulation	Compression Ratio	Actual L	Minimum L
PTT	24.57	10	2
TYT®	24.27	10	5
Vodeson	121.01	2	2
Click 'n Dig®	34.69	7	7
YSF	40.48	6	6
LoRa™	8.64	28	28

2.10 SVD summary

In this chapter, electromagnetic waves are sampled with SDRs. The SVD algorithm is successfully employed to reduce the vectors necessary to re-constitute, transmit, and activate the matched receiver. This reduces the required vectors down to $0.2 - 2.8$ percent of the original vectors, and provides data compression ratios from about 8.64 to 121. The compression ratios significantly reduce storage and handling of RF IQ data. Finally, I determined a way to transform RF IQ data via STFT into four channels. These four IQ STFT channels are key for TF RF domain classification in subsequent chapters.

Chapter 3

TF domain RF classification: Single sample rate/single SDR

This chapter contains material that was published in the 29th European Signal Processing Conference [35]

In this chapter I analyze how to perform RF frequency domain classification using all the information from the STFT, and I compare the efficacy of my proposed method versus classification models that only use "spectrograms" or the IQ data magnitude.

There are primarily two sources of RF signal data for machine learning: spectrum sampling from commercial-off-the-shelf (COTS) transceivers via SDRs [36][12][13] or software-generated synthetic datasets. The synthetically generated IQ data can either be augmented and used to train NN models or streamed to a SDR that transmits the data into the RF spectrum–either OTA or closed-loop–and then sampled by a receiver SDR into IQ data and stored. The latter approach can add additional augmentation to the IQ data that may not be easily artificially performed.

I open-source my project, which includes all the training and testing datasets and the source codes. Moreover, I also provide a graphical user interface (GUI) that researchers can easily utilize to build their own labeled datasets[1].

3.1 Introduction

ISP waveform classification can help optimize transceiver spectrum operating specifications. One of the primary spectrum challenges is co-channel interference. This occurs when more than one waveform is occupying the same channel in a given period of time. This can result in degradation of channel quality and adversely effect data throughput. Many modern waveforms that operate in common shared frequency bands can mitigate interference with orthogonal coding and frequency

---

[1] https://saige.sice.indiana.edu/research-projects/rf-classification

hopping schemes. However, many legacy waveforms do not use this methodology to decrease co-channel interference and are subject to channel degradation. Cognitive spectrum awareness, in part, through waveform classification can help mitigate this problem.

RF transceiver equipment depend on DSP and SSP methods to process RF data in modern communications systems. These methods provide invaluable signal processing, including robust noise mitigation. However, these processing methodologies do not provide for systems to learn from the data within their domain and make intelligent decisions that can allow a transceiver to modify its primary operating parameters. The optimization and orderly use of the ISM and DSS [37] [38] bands are critical for RF domain access expansion and will promote full spectrum utilization.

3.2 Motivation

In this chapter, I develop a machine learning-based data-driven ISP system. ISP is a critical enabling technology for the RF domain that will fundamentally change the way transceivers access the RF spectrum. First, ISP enabled transceivers can classify the waveforms operating within their spectrum band and then determine what waveforms may have a higher tiered priority access to the band [7] [8]. Additionally, data-driven decisions could help determine unoccupied RF channels to operate in and also perform more complicated decisions, such as determining an optimal modulation, bandwidth, and adjust data rate for optimized transceiver operation. Waveform classification, transceiver fingerprinting [39] [40] [41], radar object detection and classification [42], in conjunction with traditional DSP and SSP functions are all capabilities benefiting from a data driven RF processing approach. As a result, my proposed RF classification pipeline could provide ISP enabled transceivers the ability to mitigate problems associated with operating in ISM and DSS allocation bands and promote streamlined traffic flow.

The RF spectrum is finite and novel management techniques are beginning to be employed to optimize spectrum access. For example, in the $3.5\,GHz$ U.S. band, prioritized spectrum sharing was adopted in 2017. This $3.5\,GHz$ band has been designated as a dynamic protection area. This band uses DSS to increase spectrum access by utilizing a multi-tiered user approach. Hardware dynamically adapts to users accessing the spectrum [43] based on a three tier level protocol. Transmitting in these bands requires hardware that is cognitive and able to discern which frequencies that higher priority waveforms are occupying then find an appropriate free channel for transceiver operation.

RF classification could also be key in helping optimize these types of operating bands, as well as traditional ISM bands, where transmitted waveforms can be co-channel without the operator's cognizance. To understand how RF classification can help optimize the RF spectrum, let's use some examples. In amateur radio [44], operators are taught to use radio "etiquette" and obey specific operating rules. Rules regulating operating frequencies, transmit power, and modulation are some of the basics. Radio etiquette includes not transmitting on a frequency if someone else is actively transmitting, knowing when it is acceptable to "break" into a channel to talk, and making sure transmit power is limited to only what is needed for a good communication signal. Another analogy to consider would be automotive traffic; on a busy city street, traffic lights help keep traffic flow regulated and help minimize accidents. Additionally, drivers know that when they hear a siren and see emergency vehicles, that they are to pull over to allow the emergency vehicles to pass safely.

ML enabled cognitive transceivers could determine unoccupied frequencies, higher priority waveforms by their classification, and hopping waveforms. Knowledge of spectrum traffic conditions could allow the cognitive transceiver to modify transmit power, operating frequency, bandwidth, modulation, data rate, and determine open channels for using an appropriate hopping waveform.

3.2.1 Time division multiple access (TDMA) example

To illustrate optimizing spectrum occupancy through an ISP enabled system, consider a time division multiple access (TDMA) frequency hopping waveform in an ISM band. In this TDMA example, the incumbent waveform meets dwell requirements and provides robust noise mitigation for itself, but it is not using all available time slots. Now consider a new transceiver preparing to enter the spectrum that needs to operate in or around the incumbent waveform. I envision an ISP-enabled device that would classify the incumbent waveforms, determine the approximate operating frequency (i.e., within resolution of fast Fourier transform (FFT) [45] [46] bin), as well as the temporal parameters of the incumbent frequency hop. Knowing the frequency, dwell time, and hop sequence, the ISP-enabled system could recommend several courses of action in order to avoid co-channel interference, e.g., by using a different hop sequence to avoid interference in time or by choosing a different set of channels to avoid interference in frequency. These are only a few examples of potential decisions that could be made possible due to ISP spectrum cognizance.

3.3 RF Domain

Consider the seven level Open Systems Interconnection (OSI) concept model in Fig. 3.2 [47]. Each transmitter/receiver (Tx/Rx) contains seven levels, with the physical layer being the lowest level where RF waveforms are transmitting or receiving OTA. Most OSI levels could be used to perform ML on data to provide useful data analytics. However, In OSI levels two through seven, data would need to be collected within the targeted receiver hardware or by using another receiver that has knowledge of the OSI details of the intended receiver. My research focuses on level one (RF physical layer) of the OSI model to perform waveform classification. This could be efforted without any a priori knowledge of higher OSI hardware level information and can be used with many properly configured SDR as the receiver.

Before physical layer waveform transmission, many modern cellular and wireless network proto-

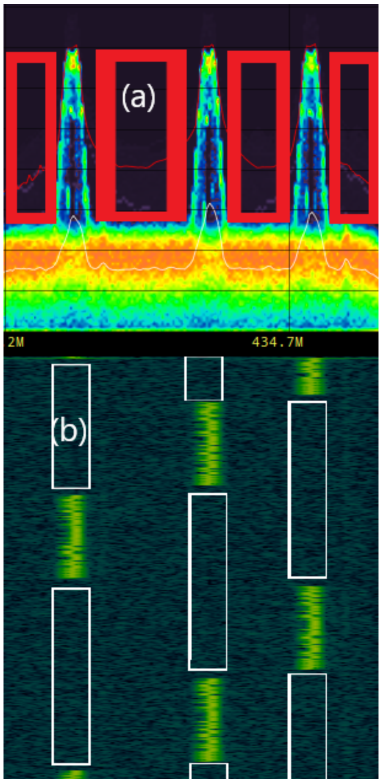


Figure 3.1: Drone TDMA waveform

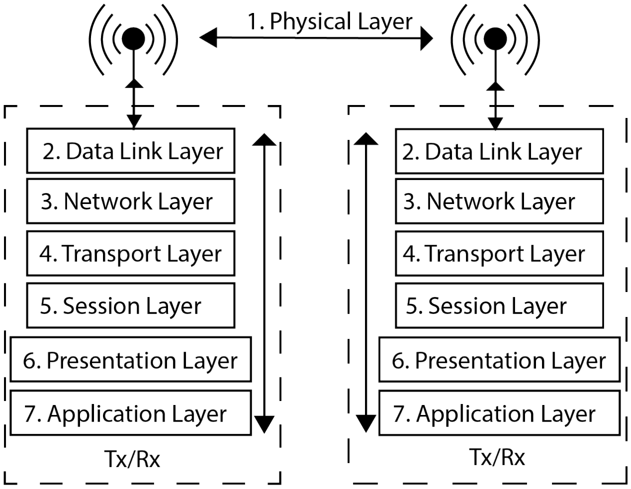


Figure 3.2: Seven layer OSI model

cols (e.g., Wi-Fi IEEE 802.11, wideband code division multiple access (W-CDMA)) typically include additional processes after packaging data into symbols, but before final transmission. Widely used encoding techniques include direct-sequence spread spectrum, and frequency-hopping spread spectrum. These encoding schemes reduce the effects of channel impairments (e.g., multi-path, channel fading) and provide processing gain.

This encoding makes it difficult to determine the underlying type of digital encoding scheme for physical layer classification (e.g., QAM, QPSK). For example, A priori knowledge of the pseudo noise code would be needed for despreading a spread spectrum waveform, along with synchronization. However, the general type of waveform can be classified at the physical layer (e.g., W-CDMA, long-term evolution (LTE), global system for mobile communications (GSM).

An alternative to direct-sequence spread spectrum and frequency-hopping spread spectrum techniques is orthogonal frequency-division multiplexing. In this method of encoding, the data stream is divided in to many sub-carriers that then use standard modulation schemes, such as QAM or QPSK. The specific sub-carriers modulation scheme could be classified at the physical layer but may require knowledge of a number of additional parameters to separate and classify the orthogonal channels from one another. Therefore, because my classification system is at the physical layer, classified waveforms are limited to what can be learned from a waveform's spectral properties, and are not necessarily able to determine an underlying modulation if spread spectrum techniques are employed. However, the overall classification of the waveform (i.e., W-CDMA, LTE, GSM, etc.) can be determined.

IQ data is natively sampled in the time domain, but it can be transformed to other domains for classification. In addition to the time domain, frequency and cyclostationary domains are utilized, and IQ data can be plotted to create IQ constellation signatures [48]) that can be used classify waveforms. Many of these methods assume no physical layer encoding scheme (e.g., direct-sequence spread spectrum, and frequency-hopping spread spectrum) and use artificially derived datasets to train ML models for modulation classification.

SDRs sample and quantize the analog RF spectrum data into IQ time domain data. Similarly, synthetically derived data is generated and saved in IQ format or transmitted over-the-air (OTA), using SDRs, then saved at the receiver side via SDR. However, before the IQ data is used in a machine learning model, such as a CNN, the data domain should be considered. Classification of IQ data can be accomplished in multiple domains, and it is useful to consider the alternatives.

Time Domain: Because IQ data is in the time domain, it does not allow expert features to be easily extracted, which is a significant benefit when it comes to heuristic features. However, the sampled IQ training data must be clear of other signals, for which an end-to-end neural network can perform feature learning and classification holistically at the cost of learning features only from the data. Meanwhile, the sampled bandwidth will also fix the classification bandwidth. The center operating frequency of the SDR would be the only insight into the approximate frequency band the classified waveforms are occupying, without further processing.

Frequency Domain: To convert the IQ data to the frequency domain requires a discrete Fourier transform (DFT) or STFT to capture the temporal dynamics of the signal over time. The magnitude of the IQ spectrogram is often used in frequency domain classifiers for training [43, 49]. This method works well with many waveforms, but the magnitude of the STFT will forfeit any phase information in the waveform that may be useful for classification. I propose to use the 3D tensor representation to encompass all the details of the IQ data in the time-frequency space as an input to the CNN classifiers (Section 3.5).

Other Domains: Other domains are also used for RF classification. For example, image processing on the IQ data from plotting the IQ constellation data[50]. Waveforms that have clear constellation signatures [48] are good candidates for this domain (e.g., quadrature phase shift keying (QPSK), quadrature amplitude modulation (QAM)).

Additionally, The Wigner-Ville, Choi-Williams, Quadrature Mirror Filter Bank (QMFB), and cyclostationary domains are widely used to classify low probability of intercept (LPI), low probability of detect (LPD) radar waveforms [42].

STFT time-frequency (TF) domain: Out of the potential choices, I propose an STFT-based TF representation, or a *spectrogram*, that fully retains real and imaginary components of IQ signals after the transformation. Given that the TF representation will be in the shape of an image with four channels, it is natural for us to employ 2D CNN models for the classification problem. However, in order to deal with the issue of the large spectrogram size and the relative sparsity, I additionally propose a patching mechanism that processes smaller regions of the input data sequentially. To effort this, I also propose a data collection and labeling method that expedites one's effort.

3.4 Data Collection and Labeling

Creating synthetic datasets is popular among researchers. Open-source software like `gnuradio` enables researchers the ability to generate datasets of many types of waveforms for RF ML, and affords datasets to be shared with others in the community that may not have the ability to create their own datasets. Additionally, software augmentation of these synthetic waveforms generates additional datasets that can simulate real-world RF environmental conditions that may be difficult to reproduce naturally, and varying the amount and type of augmentation can produce even more training data. Synthetic datasets serve an invaluable service for researchers in that the labeled datasets can help the research community focus on research versus dataset collection.

On the contrary, a dataset consisting of recordings of real-world RF signals can provide additional features that may not be created synthetically. However, those *labeled* real-world RF signals are not trivial to acquire due to the hardware and software requirements (e.g., attenuators, coaxial network, a variety of transceivers, etc.). Additionally, recording radio signals in the "wild" can introduce ethical considerations with respect to personally identifiable information (PII)—using personally controlled transceivers helps reduce PII concerns.

I use `gnuradio` in conjunction with `SigMF` to collect datasets of ten common types of waveforms in the ISM bands at $434\,MHz$ and $915\,MHz$, as well as the $70\,cm$ ($420 - 450\,MHz$) push-to-talk (PTT) amateur radio bands. All RF transceivers used for dataset collection were under my control to eliminate PII issues.

Data collection and format: The `SigMF` formatted data was collected OTA using an Ettus universal software radio peripheral (USRP)[14]. `SigMF` data consists of the IQ data file, and an accompanied JavaScript^{TM} object notation (JSON) formatted labeled meta data file. Each waveform is recorded for ten seconds at 1 mega-sample-per-second (msps) with 16-bit integer precision and saved as a 32-bit float IQ. The OTA data collection location is an indoor lab with many reflections that provided natural multi-path augmentation.

OTA Setup and SDR models: The OTA training, validation, and testing waveforms are recorded using an Ettus SDR N210[51] with SBX daughter board[52] and B205-mini. In-line and programmable attenuators are used to maintain a consistent SNR to the SDR during data collection, and waveforms are recorded with $30 - 40\,dB$ of SNR.

GUI software: I provide open-source tools with which researchers can collect and label their own real-world signals for RF classification instead of using synthetically generated datasets, in addition to those provided with this paper. Two custom GUIs, written in Python, allow the user to extract labeled feature patches from the IQ data to train machine learning models. Following the process, Fig. 3.3 (a), my first GUI (i.e., the interface Fig. 3.4) allows the user to select a broad swath of spectrogram area, where a visible amount of signal activities can be found, then saved. Then using the second GUI (i.e., the interface Fig. 3.5), as illustrated in Fig. 3.3 (b), the selected area of the spectrogram randomly extracts a number of small patches from various locations. The user interacts with the second phase by defining the width, height, and total number of patches to produce. As a result, the 3D patches are stored for later use as the input “images” for the CNNs

(Fig. 3.3 (c)).

Note that labeling is naturally done by assuming all patches extracted from a recording belong to the same class known to the user who operates the GUI system. The user can also easily change or add labels to the saved meta-data file, if desired.

3.5 The Time-Frequency Representation

I convert the RF signals into a time-frequency domain using STFT, as discussed in chapter 2 using equation 2.4. The Hamming window function is again maintained with a 50% hop size, to facilitate proper signal reconstruction. With a proper choice of the windowing function and hop size, the data does not lose much information during the transformation. The benefit of a TF domain transformation allows visualization of both the frequency and time variations contained in the data, which are otherwise difficult to represent. The TF representation is then a suitable format as an input to the CNN models where the input is defined as a stacked set of “images”, a 3D tensor with 4 channels.

While TF domain classification is often defined with the magnitude of the spectrogram, I propose to stack up both real and imaginary Fourier coefficients on top of each other. Therefore, I apply the STFT to each of the I and Q channels, respectively, and this generates four real-valued arrays (e.g., spectrograms). These are presented in chapter 2, equations 2.6 and 2.7.

Finally, the STFT IQ arrays are stacked to form a rank-3 (3D) tensor with four input channels. Similarly, I computed the I and Q channel magnitudes to produce a stacked 3D tensor with two channels to compare with each four channel network model. The GUI process described in Section 3.4 follows to extract small patches for model training.

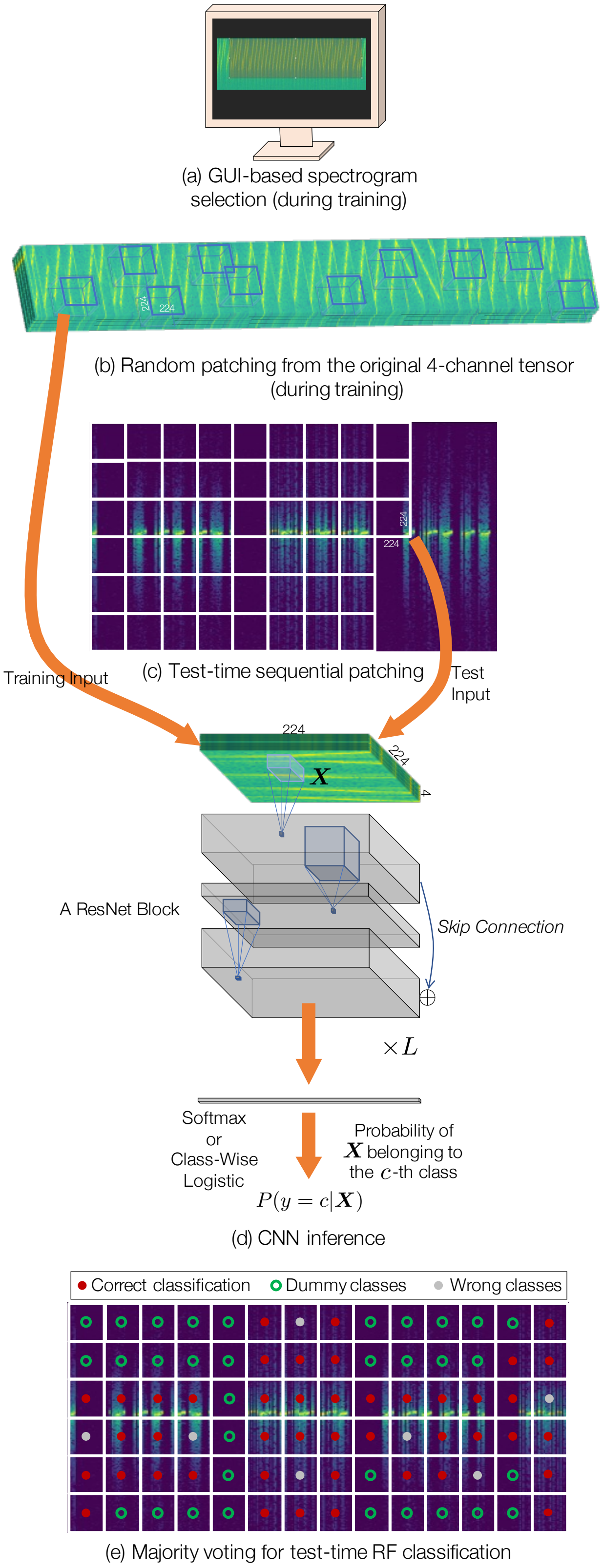
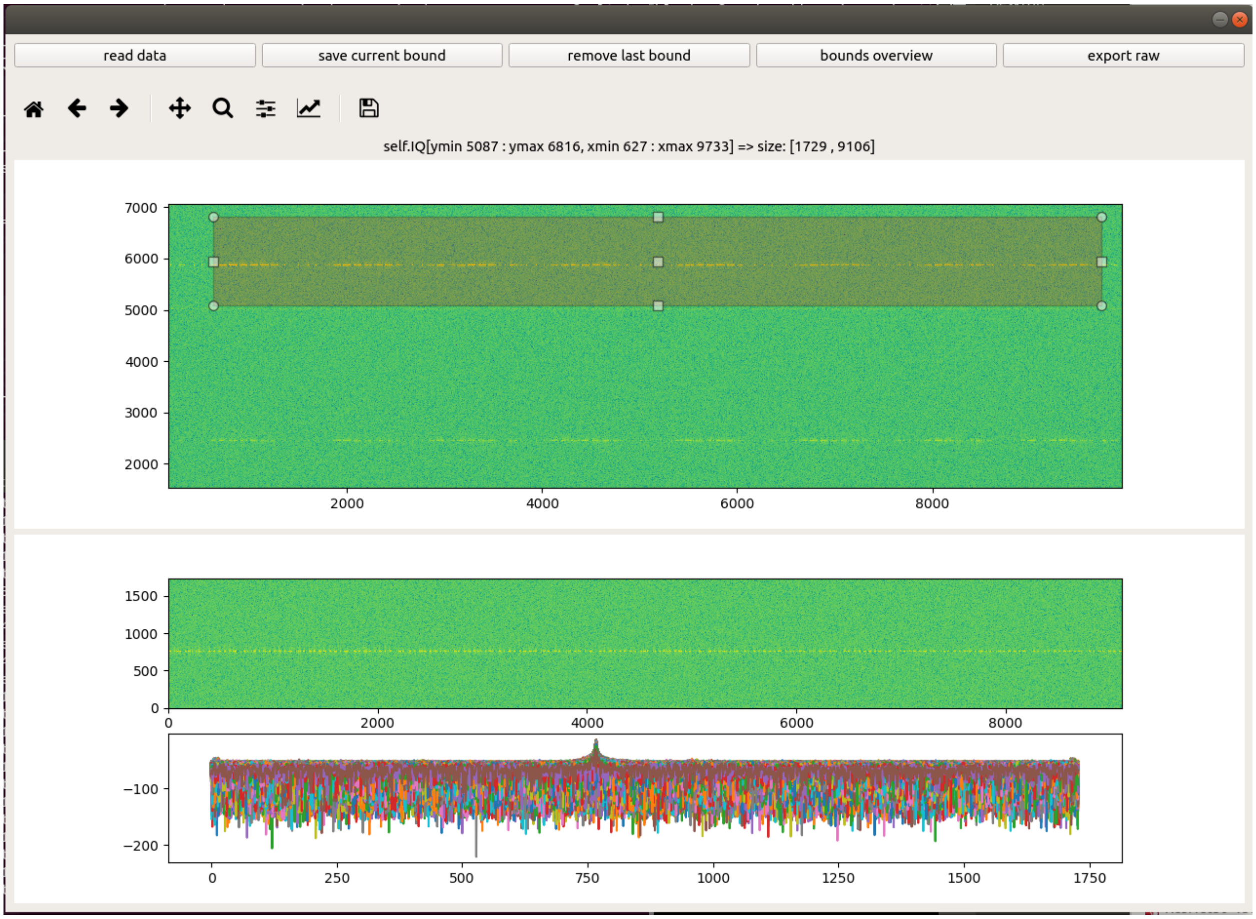
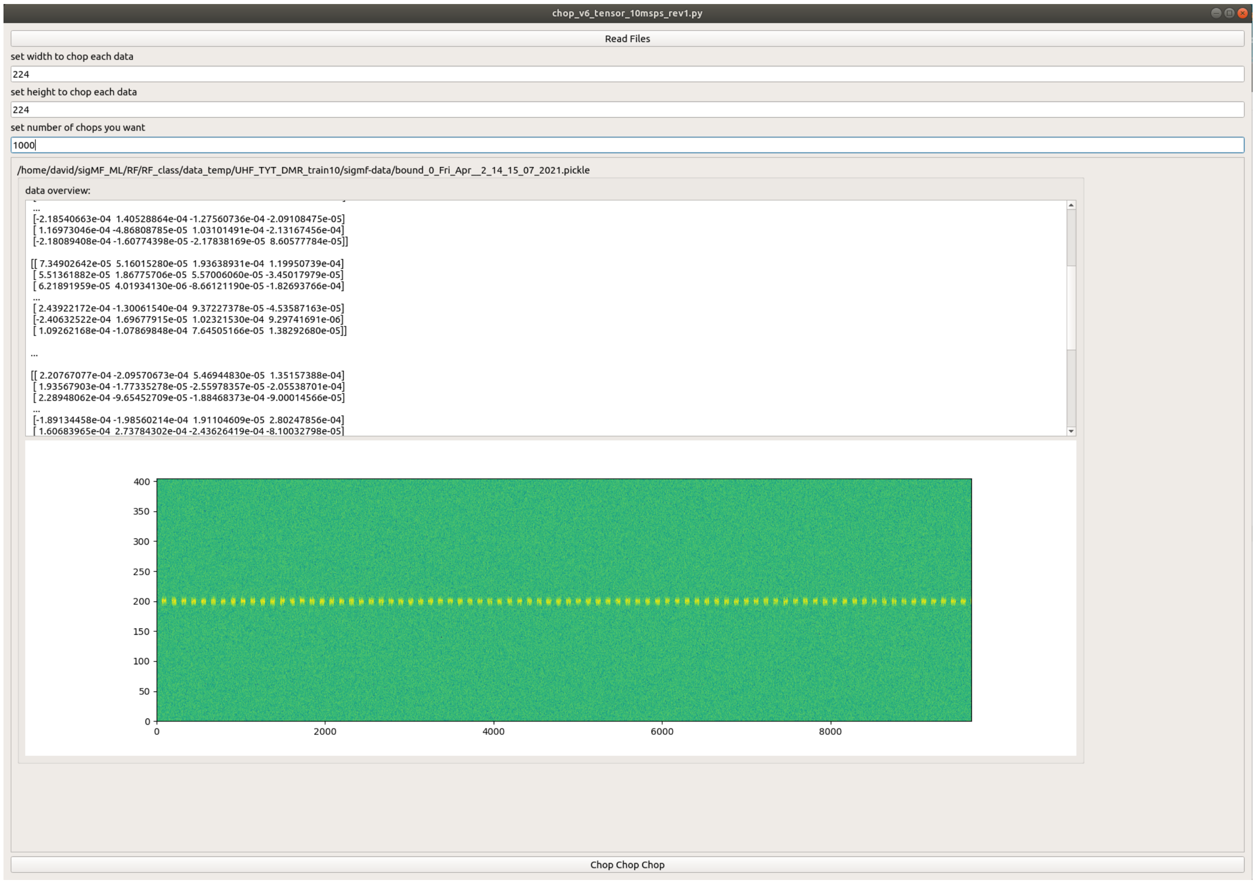

(a) GUI-based spectrogram selection (during training)
224
224
(b) Random patching from the original 4-channel tensor (during training)
224
224
(c) Test-time sequential patching
Training Input
Test Input
224
224
4
X
A ResNet Block
Skip Connection
$\times L$
Softmax or Class-Wise Logistic
Probability of X belonging to the c-th class
$P(y = c|X)$
(d) CNN inference
● Correct classification
○ Dummy classes
● Wrong classes
(e) Majority voting for test-time RF classification

Figure 3.3: Overview of the proposed RF classification system.



```
self.IQ[ymin 5087 : ymax 6816, xmin 627 : xmax 9733] => size: [1729 , 9106]
```

Figure 3.4: Feature extraction GUI interface



set width to chop each data
224
set height to chop each data
224
set number of chops you want
1000

/home/david/sigMF_ML/RF/RF_class/data_temp/UHF_TYT_DMR_train10/sigmf-data/bound_0_Fri_Apr__2_14_15_07_2021.pickle

data overview:

```
 ...
 [-2.18540663e-04  1.40528864e-04 -1.27560736e-04 -2.09108475e-05]
 [ 1.16973046e-04 -4.86808785e-05  1.03101491e-04 -2.13167456e-04]
 [-2.18089408e-04 -1.60774398e-05 -2.17838169e-05  8.60577784e-05]]

[[ 7.34902642e-05  5.16015280e-05  1.93638931e-04  1.19950739e-04]
 [ 5.51361882e-05  1.86775706e-05  5.57006060e-05 -3.45017979e-05]
 [ 6.21891959e-05  4.01934130e-06 -8.66121190e-05 -1.82693766e-04]
 ...
 [ 2.43922172e-04 -1.30061540e-04  9.37227378e-05 -4.53587163e-05]
 [-2.40632522e-04  1.69677915e-05  1.02321530e-04  9.29741691e-06]
 [ 1.09262168e-04 -1.07869848e-04  7.64505166e-05  1.38292680e-05]]

...

[[ 2.20767077e-04 -2.09570673e-04  5.46944830e-05  1.35157388e-04]
 [ 1.93567903e-04 -1.77335278e-05 -2.55978357e-05 -2.05538701e-04]
 [ 2.28948062e-04 -9.65452709e-05 -1.88468373e-04 -9.00014566e-05]
 ...
 [-1.89134458e-04 -1.98560214e-04  1.91104609e-05  2.80247856e-04]
 [ 1.60683965e-04  2.73784302e-04 -2.43626419e-04 -8.10032798e-05]
```

Figure 3.5: Random patch generator GUI interface

3.6 The Proposed RF Classification Pipeline

3.6.1 Experimental Setup

Computing Environment: Model training was implemented via Pytorch [53] using GPU computing performed on multiple GPUs via Pytorch, including:

- RTX 8000 (4608 NVIDIA® CUDA® cores) with 45 GB of GPU memory.
- EVGA® GTX 1080Ti (3584 NVIDIA® CUDA® cores) with 11 Gb of GPU memory.
- RTX 2080 Max-Q (3072 NVIDIA® CUDA® cores) with 8 GB of GPU memory.

All listed GPUs generally provided ample memory to run most of the "light-weight" network variations, however, only the RTX 8000 with 45 Gb of GPU memory was able to run the parameter heavy VGG and ResNet (i.e., $> ResNet18$) architectures *and* compute the STFT for IQ test files greater than 1 $msps$.

3.6.2 GUI-based patching

To train the CNN-based RF signal classification system, I first generate patches from the TF representation of the training signals using the aforementioned GUI process.

The default patch size is 224 × 224 × 4; this provides sufficient spectral diversity for training the network on the selected waveforms. This patch size may need to be reconsidered for much wider bandwidth waveforms (i.e., 4G, 5G, and Wifi). If wider bandwidths are trained with the current patch size, a *grouper* function that would consolidate like classified adjacent patches may be helpful.

Training Data and Patching: For this chapter, a total of six waveforms per class were recorded for each phase of the supervised learning, i.e., training, validation, and testing. 1,700 patches were collected per training waveform, totaling 10,200 patches per class, while I use 2,000 per class for validation from validation recordings.

Each patch is saved as a 3D tensor of (time)×(frequency)×(channels), where the third one holds real and imaginary channels of both I and Q signals. Width and height dimensions of 224×224 were chosen for the GUI-based patching because this matched the VGG and ResNet native architectures designed for visual object classification [54], as well as provided sufficient spectral diversity for training the network on the selected waveforms.

3.6.3 RF Signal Classes

I focus my research on relatively narrow RF waveforms (i.e., channel bandwidth $\leq 250\,kHz$) in a spectrum space of up to $25\,MHz$ of bandwidth. This means that I could have as little as one narrow-band waveform (i.e., $3\,kHz$ channel width) or many waveforms within this frequency space. My approach is to not only classify one or more waveforms, but also to determine frequency and temporal parameters. I focus on the operating frequency for this research effort, but the temporal relationships are inherent in my method and can be used to provide of temporal metrics (e.g., hopping waveform timing along with channel resolution).

A variety of ten waveforms are selected for collection. The abbreviation in parentheses after the waveform name is how the results identify the particular waveform class. The classes consists of analog PTT transceivers (NFM) and three types of DMR transceivers (GD55, TYT, and YSF).

The collection of transceivers used throughout this research are depicted in Fig. 3.6.

All transceivers/transmitters are connected to external power when possible or receive new batteries for optimal transmit power. Waveforms in figures 3.7(g)-(i) all use small *button* style batteries and are designed to operate with much less than $20\,dBm$ of transmit power. Antenna selection for both transmitter and receiver hardware is optimized to match transceiver wavelengths. Waveforms are selected based on a couple of criteria: modulation diversity (e.g., analog, digital), and not to exceed the usable bandwidth of the lowest sample rate (i.e., $(1msps)/2 = 512\,kHz$).

The following waveforms are illustrated in Fig. 3.7. For the plots, the time and fft axis are

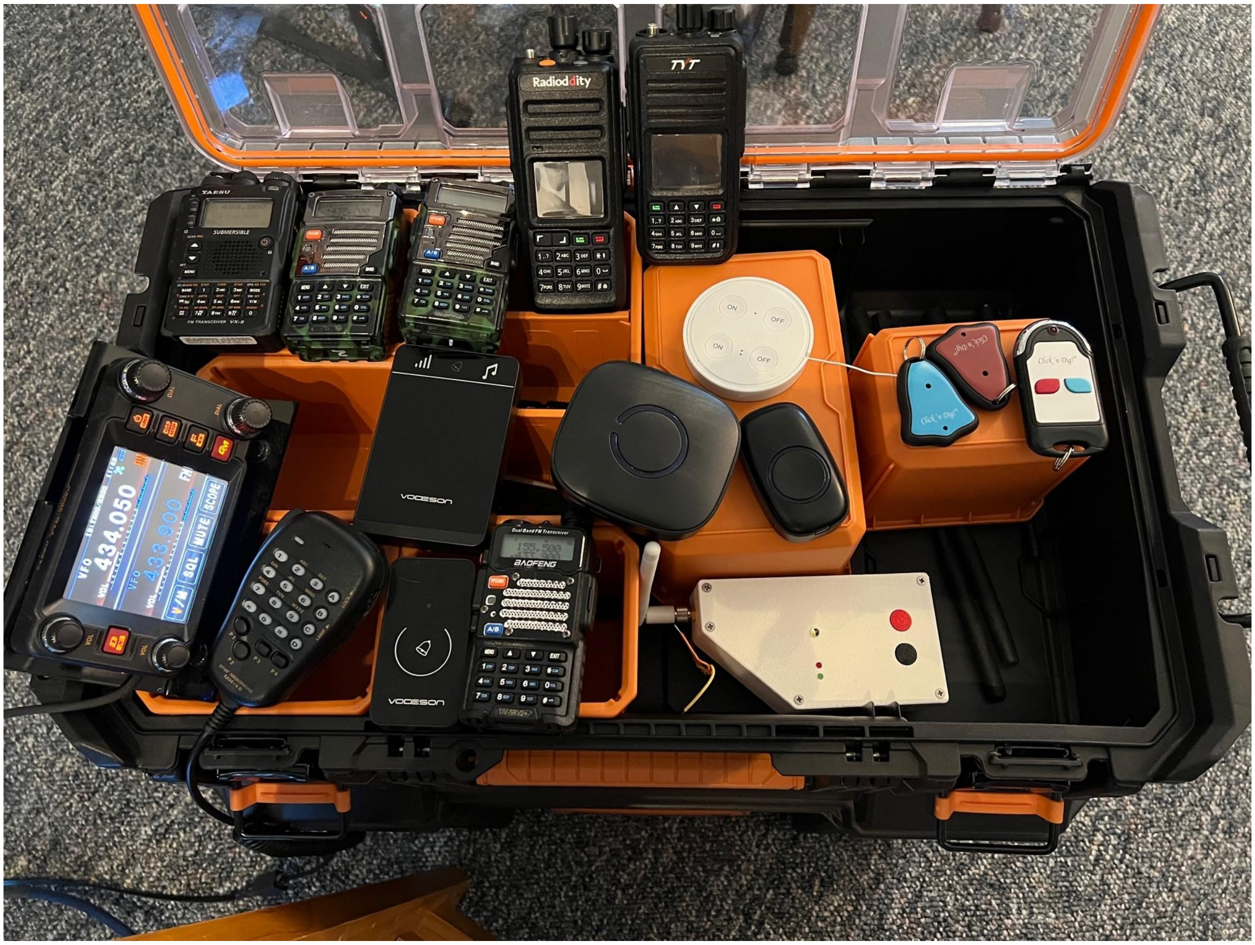


Figure 3.6: Transceivers

chosen to exhibit good spectral details of each waveform. The time axis is limited to 300 fft bins and correlates to 150 $\mu seconds$ of time displayed, but the frequency fft bin axis selection is dependent on waveform bandwidth and waveform similarity.

The frequency fft bin resolution is matched for 3.7(a), and (b), illustrating the difference in the two LoRa$^{\text{TM}}$ examples and displays $\sim 307\,kHz$ of bandwidth. Note also the frequency fft bin resolution for Figures 3.7(c)-(i): The value chosen helps give a sense of scale to these waveforms with a displayed bandwidth of $\sim 61\,kHz$. Fig. 3.7(j) frequency fft bin resolution selection is unique for illustration purposes and displays $\sim 102\,kHz$ of bandwidth.

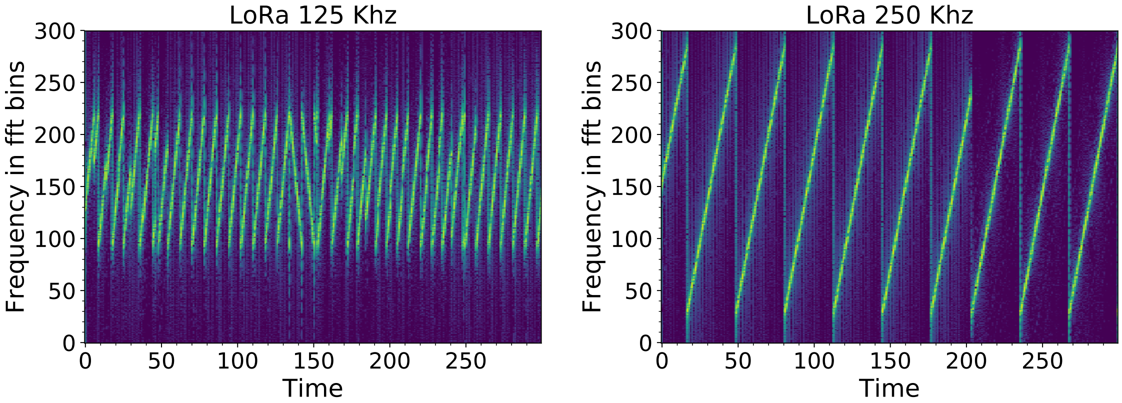
Two bandwidths of the long-range (LoRa$^{\text{TM}}$) waveform are chosen, LoRa$^{\text{TM}}$ $125\,kHz$ and $250\,kHz$ (i,e., lora125, and lora250). Figures 3.7(a) and (b) illustrate these waveforms.

Analog PTT transceivers are used to collect the narrow frequency modulation (NFM) waveforms. The NFM waveform collection are spectrally the narrowest bandwidth of all classes with a channel bandwidth of $\sim 3kHz$. This waveform is depicted in Fig. 3.7(c).

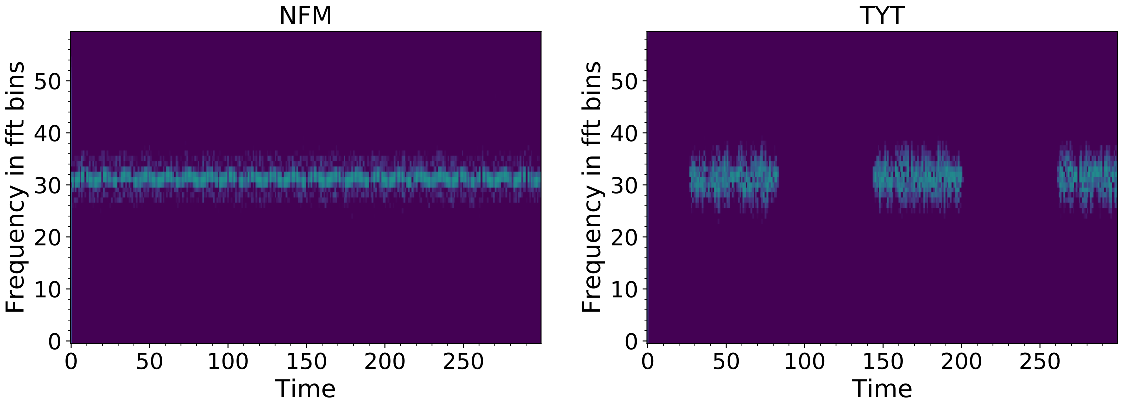
Three DMRs (TYT, GD55, YSF) are chosen and illustrated in Fig. 3.7(d)-(f). Two ISM RF doorbells (Vodeson and Sado) are selected; these waveforms are depicted in Figures 3.7(g), and (h). A key finder (click) is selected and depicted in Figure 3.7(i). Finally, a RF light switch controller (light) is selected and is illustrated in Fig. 3.7(j).

Two additional classes account for spectrogram areas that contained noise and the DC artifact present at the center bin of the discrete Fourier transform due to the SDR design (direct conversion receiver) [55] [56].

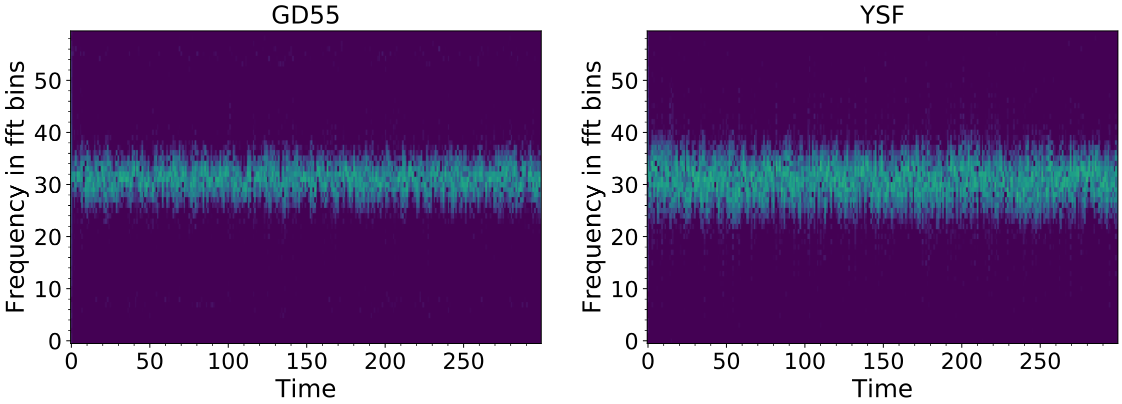
Data Augmentation: RF transmissions often have a variety of channel impairments. Multipath wave propagation, noise (e.g., environmental, transceiver, etc.), oscillator frequency drift, IQ imbalance, and phase noise are common. For this chapter, the only augmentation was to SNR. I inject additive white Gaussian noise (AWGN) with a “loudness” randomly chosen from a range between 0.1 to $10.0\,dB$ for each patch.



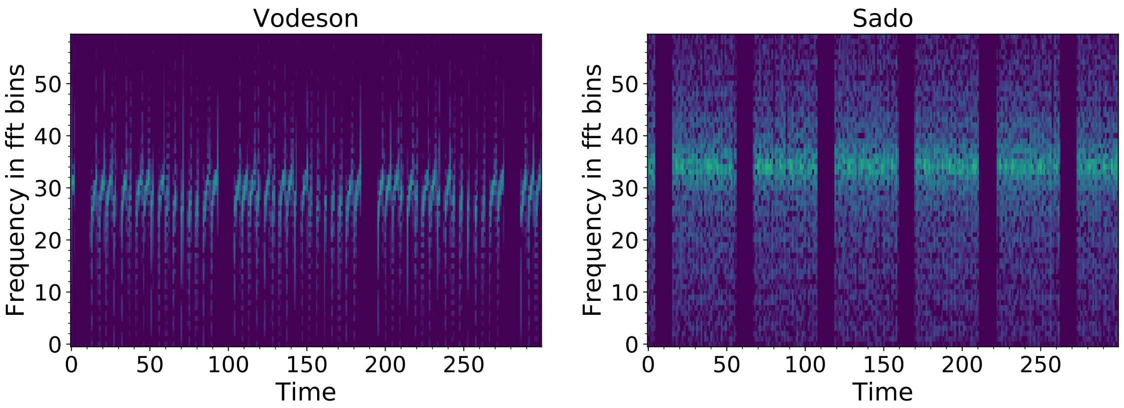
(a) LoRa$^{\mathrm{TM}}$ 125 kHz (b) LoRa$^{\mathrm{TM}}$ 250 kHz



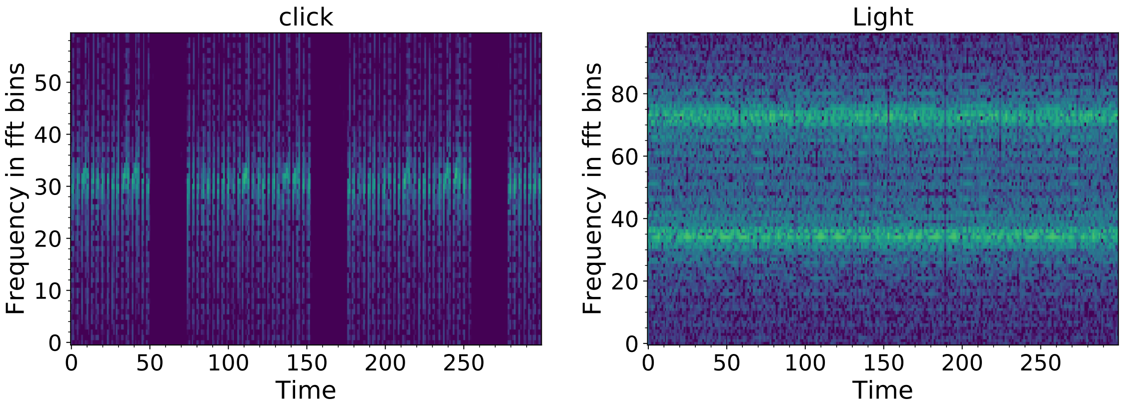
(c) NFM (d) TYT



(e) GD55 (f) YSF



(g) Vodeson (h) Sado



(i) click (j) Light

Figure 3.7: Waveform Classes

CNN Models: Primarily, three CNN network models are used to classify these waveforms: VGG16 [57] with 16 hidden layers, and ResNet [58] with 18 and 50 hidden layers, respectively. Each optimized model was run for 100 epochs.

3.6.4 Test-Time Inference

3.6.4.1 Single-class patch case: majority voting

During the testing phase, the goal is to determine which class the observed signal belongs to, regardless of how many patches I can extract from it. As a result, during the test period, I take the whole four-channel spectrogram and extract non-overlapping patches consecutively (i.e., Fig. 3.3 (c)). All patches are serially batched to the model, which then combines the classification results to get a final choice using a majority voting mechanism (Fig. 3.3 (e)). A 10 *second*, 1 *msps* test signal, for example, turns into a $513 \times 19,532 \times 4$ tensor, which results in 174 spectrum patches.

The test-time signal classification is based on the assumption that for a given time period there is only one dominant signal class. Indeed, I recorded the real-world test signals in a controlled environment to meet this constraint. However, a naïve approach that simply selects the majority class out of the patch-by-patch classification results is not a solution because signal sparsity in the time-frequency space will classify most of the patches as the `noise` class.

The winner-take-all majority vote strategy is depicted in Fig. 3.3 (e). I first count all the *meaningful* predictions that belong to the ten critical classes (red or grey filled circles in the figure). The `noise` and `DC center frequency` (hollow green circles) are dummy classes and not counted. It will equate to the sum of the red and gray dots in the figure, $13 + 3 = 16$. I can see that 13 of the 16 patches belong to a single class (Correct for the vodeson class), whereas the gray dots indicate that those patches are misclassified into other non-dummy classes. As a result, the entire region of these 6×16 patches is classed as the red dot class, i.e., Vodeson class.

The operating frequency can also be inferred from the majority voting strategy. The frequency bin for each patch is recorded, and as classified patches accumulate, the distribution curve in Fig. 3.8 indicates the probable operating frequency of the classified waveform.

Since there is no efficient way to label each patch, I record a 10 second long signal containing only one dominant class's activity and performed inference on its patches. Likewise, the classification results are for the entire 10 second long recording.

3.7 Discussion

I used three popular 2D CNN network architectures: VGG16, ResNet50, and ResNet18, where the number indicates the number of hidden layers.

The ResNet architecture is utilized in this research; it features a number of benefits for RF

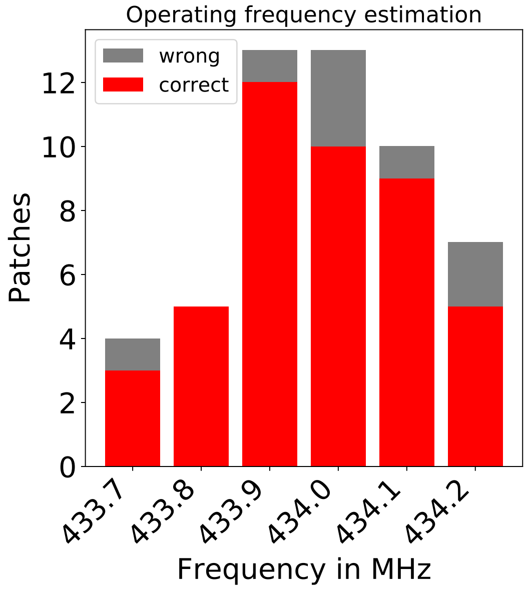
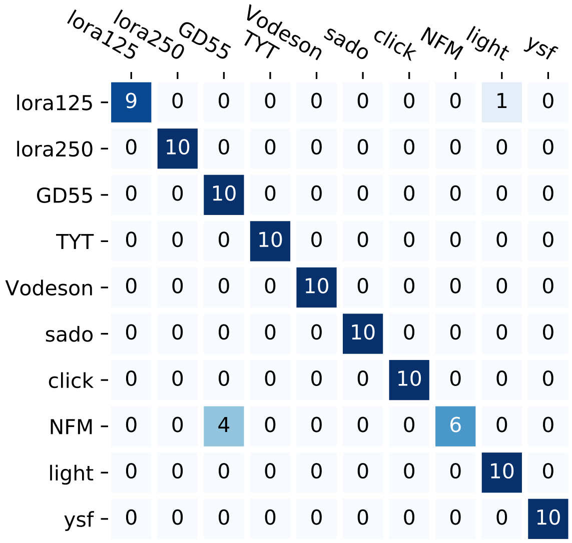


Figure 3.8: Determining operating frequency from classified patches.

classification. First, ResNet (residual network) addresses the problem of vanishing/exploding gradients through the use of "skip" connections. Skip connections take the outputs from the previous block, input them to the next block and simultaneously feed forward around the current block and into the next block. I utilize the ResNet 50 incarnation of this architecture; this features 50 hidden layers, $L = 30$ blocks and 16 skip connections.

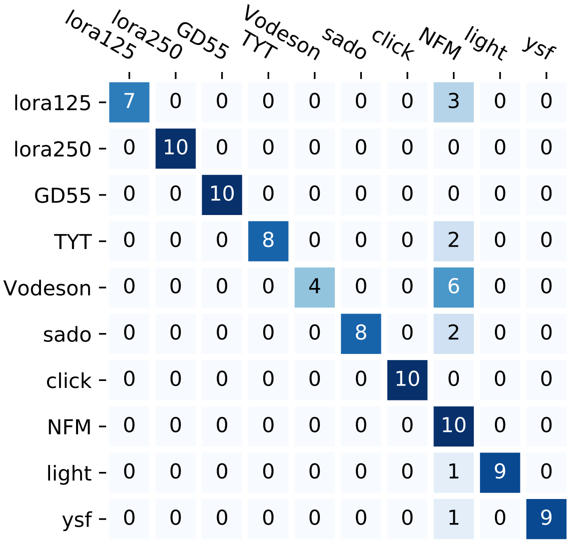
Each network model is further distinguished by 2 and 4 channel 3D tensor input, where the former uses magnitudes of the DFT coefficients from I and Q channels, while the latter uses both real and imaginary coefficients of the IQ data. These network combinations produce six different test configurations. The models were trained using the Adam optimizer [59], and initial learning rates were found using validation: 1×10^{-4} for the ResNet models and 1×10^{-6} for the VGG models. With an early stopping strategy, I stored the models that gave the best validation performance and used them for testing.

All models achieved more than 99% validation accuracy, although the test-time performance varied due to the discrepancy between validation and testing, as well as the models' different char-



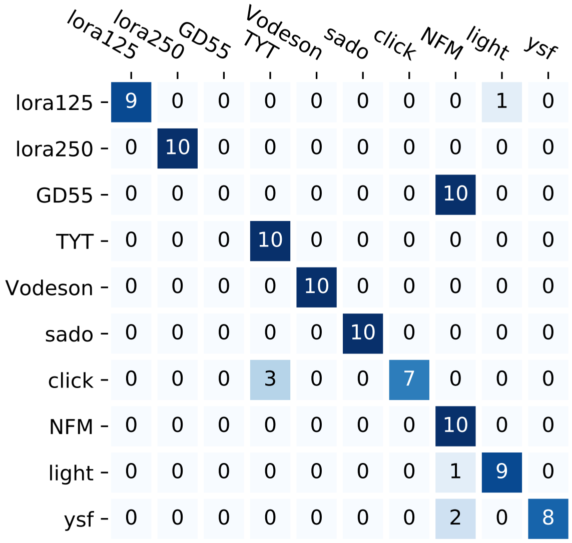
	lora125	lora250	GD55	TYT	Vodeson	sado	click	NFM	light	ysf
lora125	9	0	0	0	0	0	0	0	1	0
lora250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	10	0	0	0	0	0
sado	0	0	0	0	0	10	0	0	0	0
click	0	0	0	0	0	0	10	0	0	0
NFM	0	0	4	0	0	0	0	6	0	0
light	0	0	0	0	0	0	0	0	10	0
ysf	0	0	0	0	0	0	0	0	0	10

(a) ResNet50 4 channel (**95%**)



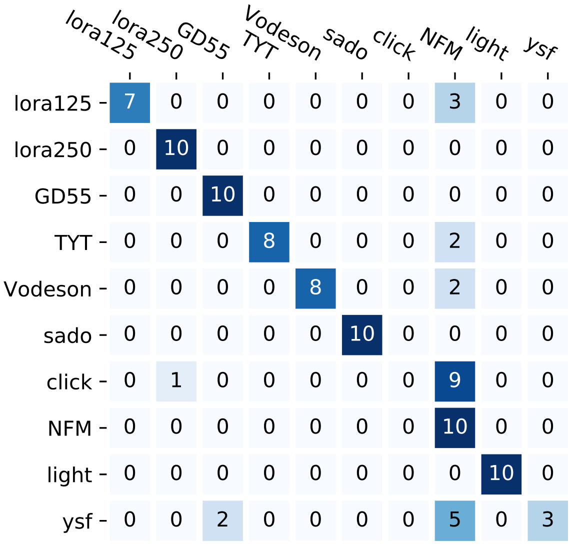
	lora125	lora250	GD55	TYT	Vodeson	sado	click	NFM	light	ysf
lora125	7	0	0	0	0	0	0	3	0	0
lora250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	8	0	0	0	2	0	0
Vodeson	0	0	0	0	4	0	0	6	0	0
sado	0	0	0	0	0	8	0	2	0	0
click	0	0	0	0	0	0	10	0	0	0
NFM	0	0	0	0	0	0	0	10	0	0
light	0	0	0	0	0	0	0	1	9	0
ysf	0	0	0	0	0	0	0	1	0	9

(b) ResNet50 2 channel (**85%**)



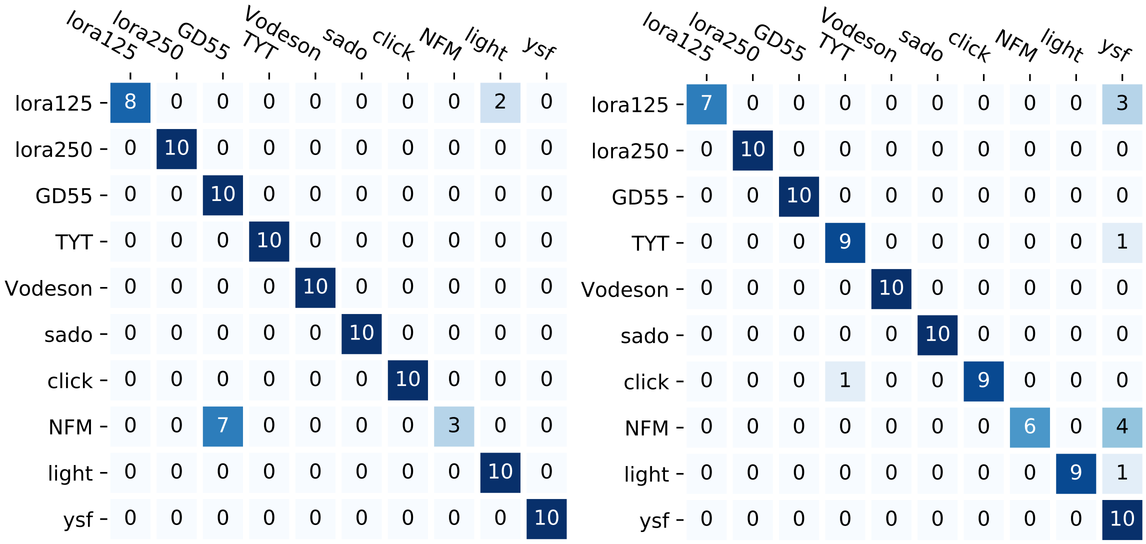
	lora125	lora250	GD55	TYT	Vodeson	sado	click	NFM	light	ysf
lora125	9	0	0	0	0	0	0	0	1	0
lora250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	0	0	0	0	0	10	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	10	0	0	0	0	0
sado	0	0	0	0	0	10	0	0	0	0
click	0	0	0	3	0	0	7	0	0	0
NFM	0	0	0	0	0	0	0	10	0	0
light	0	0	0	0	0	0	0	1	9	0
ysf	0	0	0	0	0	0	0	2	0	8

(c) ResNet18 4 channel (**83%**)



	lora125	lora250	GD55	TYT	Vodeson	sado	click	NFM	light	ysf
lora125	7	0	0	0	0	0	0	3	0	0
lora250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	8	0	0	0	2	0	0
Vodeson	0	0	0	0	8	0	0	2	0	0
sado	0	0	0	0	0	10	0	0	0	0
click	0	1	0	0	0	0	0	9	0	0
NFM	0	0	0	0	0	0	0	10	0	0
light	0	0	0	0	0	0	0	0	10	0
ysf	0	0	2	0	0	0	0	5	0	3

(d) ResNet18 2 channel (**76%**)



	lora125	lora250	GD55	TYT	Vodeson	sado	click	NFM	light	ysf
lora125	8	0	0	0	0	0	0	0	2	0
lora250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	10	0	0	0	0	0
sado	0	0	0	0	0	10	0	0	0	0
click	0	0	0	0	0	0	10	0	0	0
NFM	0	0	7	0	0	0	0	3	0	0
light	0	0	0	0	0	0	0	0	10	0
ysf	0	0	0	0	0	0	0	0	0	10

(e) VGG16 4 channel (**91%**)

	lora125	lora250	GD55	TYT	Vodeson	sado	click	NFM	light	ysf
lora125	7	0	0	0	0	0	0	0	0	3
lora250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	9	0	0	0	0	0	1
Vodeson	0	0	0	0	10	0	0	0	0	0
sado	0	0	0	0	0	10	0	0	0	0
click	0	0	0	1	0	0	9	0	0	0
NFM	0	0	0	0	0	0	0	6	0	4
light	0	0	0	0	0	0	0	0	9	1
ysf	0	0	0	0	0	0	0	0	0	10

(f) VGG16 2 channel (**90%**)

Figure 3.9: The confusion matrix of different systems in comparison. The total classification accuracy is presented in parenthesis.

acteristics. Each class consists of 10 signals, whose length is fixed to 10 seconds. The classification accuracy is then defined by the number of correctly classified divided by 100. In Fig. 3.9 I present the confusion matrix for more analysis, where y-axis (i.e., rows) are the ground-truth class labels and x-axis (i.e., columns) are the prediction.

The first observation I make is the overall acceptable performance of all systems. Except for the ResNet18 2 channel input case, all the systems were able to achieve more than 80% accuracy, showcasing the robustness of the proposed classification scheme. I also point out that the data augmentation process that injects Gaussian random noise to the input patches helped stabilize the test-time performance. The comparison between the ResNet models leads to the conclusion that the deeper ResNet50 models ($\sim 23.5\,M$ trainable parameters) outperform the shallower ResNet18 models ($\sim 11\,M$), and imply that deeper models are more favorable.

On the other hand, VGG16 models showed an interesting behavior: their overall performance is good (91% with 4 channel input and 90% with 2 channels), but given that they have many more parameters (134M), the performance is not impressive. This behavior is expected because

ResNet's more advanced features, such as skip connections, are known to outperform VGG. For example, considering the test-time inference complexity of these models, as well as the performance, ResNet50 on the 4 channels should be the choice rather than VGG16. In an extreme environment where minimal resource usage is required, ResNet18 should be the choice.

It is also noticeable that the proposed 4 channel input tensors greatly outperform the 2 channel inputs. It is because the 4 channels retain all the details about the phase information, which the 2 channel data is missing. The difference is more salient in the smaller ResNet models than the VGG16 models.

3.8 TF domain classification summary

This chapter explored RF classification by defining it as an "image" classification problem on multi-channel input. I show that popular 2D CNN models, such as ResNet and VGG, can classify waveforms in the time-frequency representation. I also observed higher accuracy when using all the magnitude and phase information of the TF representation of the IQ signals compared to using the magnitude-only approach that is more popular in the literature. I open-sourced my project not only to improve reproducibility, but to help the researchers create their own dataset via my proposed GUI-based annotation system.

Chapter 4

TF domain RF classification: Multi-sample rate/Multi-SDR

In the last chapter, I focused my design on an ISP classification system to train on only one sample rate for a specific SDR. This may meet requirements for a number of purposes; however, many SDRs can sample the RF signals at different sample rates. Sampling only the bandwidth of interest can reduce noise into the system by focusing only on the spectrum of interest. To this effort, I design a system to ideally train on multiple given sample rates, then inference on an unseen sample rate.

In addition to a multi-sample rate SDR, using various SDR devices should be considered. ML models are highly sensitive to subtleties in data. Different noise figures and performance specifications of SDRs can affect classification performance significantly unless they are properly incorporated into the training procedure. In this chapter, I use multiple SDRs for data collection as well as for testing. Multiple sample rates and multi-device choices present a number of challenges that require a hybrid approach to training the classifier. To the best of my knowledge, identifying various discrepancies observed in the SDRs and overcoming them via a data-driven approach is a novel contribution of this work.

While a RF spectrum can be very sparse, it can also be extremely dense. It is not uncommon for more than one waveform to be in close channel proximity to another waveform, or to partially or completely occupy the same channel as another waveform. Multi-source waveform classification then becomes challenging as the number of waveforms increases and the proximity to each other decreases. A cognitive transceiver would need adequate spectral “situational awareness” to make informed changes to waveform parameters. The proposed method provides for classification of up to three waveforms per patch, with various degrees of superposition.

4.1 Related work

In the time domain, O'Shea et al. [60] [61] [62] perform motivating examples of RF classification, as well as [63] and [64].

In addition to the time domain, IQ data can be transformed to other domains that offer additional benefits that the time domain does not provide. The cyclostationary bi-frequency domain utilizes the *wide-sense* stationarity of 2^{nd} order statistical moments of RF waveforms [65]. This domain is used in [66] [67], and [42] to perform RF classification on communication and radar waveforms. These classification methods differ from my approach in that I choose an alternative TF domain to perform the classification.

Machine learning (ML) models trained in [49], [68] and [69] use artificially derived datasets. The synthesized datasets have merit as they can be acquired more freely than real-world recordings. In addition, OTA waveform affects can be simulated through data augmentation to mimic real-world channel conditions (e.g., multi-path, IQ imbalance, etc.). However, I differentiate this chapter through the use of commercial-off-the-shelf (COTS) hardware and OTA transmission to organically collect labeled RF IQ datasets. This allows for natural channel augmentation, as well as real-world hardware specific imperfections (e.g., IQ imbalance, phase noise, etc.), to be introduced into my datasets.

The aforementioned works differ from my approach in another respect. In the last chapter, I focused on physical-layer RF classification by transforming IQ data from the time domain to the TF domain via STFT. Prior research, such as [69] [43] and [49], also use TF domain IQ data but are different from my approach. I use *all* the information from the TF transform versus their lossy magnitude-only representation of the TF transform or only partial phase information.

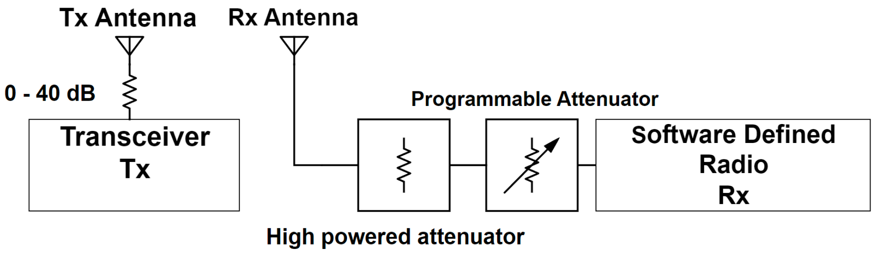
The TF domain provides notable benefits thanks to its ability to visualize the spectrum. First, it allows us to use graphical user interfaces (GUIs) to visualize and interact with the signals, which is an important part of my data collection. I collect data from my COTS hardware OTA, this means that features of interest could be very sparse and also contain signals that are of no interest. Visualization is then vital for sorting out good data and with my open-sourced GUI, users can visualize the RF signals via STFT and then draw bounding boxes around the useful sub-area, which is then sub-divided into small patches.

Second, the TF representation can be considered as a spatially structured array, a rank-4 tensor to be specific. Accordingly, the patch can be seen as an "image" with four channels, which is a common representation used in various computer vision systems based on the convolutional neural network (CNN) architecture (e.g., VGG [57] and ResNet [58]). Users can then conveniently inherit the network architectures from the existing CNN classifiers with only minor modifications to the framework (i.e. number of classes and input channels). Third, during inference, I can determine the operating frequency (i.e., within the resolution of a frequency subband) of classified waveforms, patch by patch, as well as keep track of temporal aspects of the waveform.

4.2 SDRs and format

New datasets were collected for multi-sample rate classification using four Ettus USRP SDRs: This included an N210 [51] with SBX daughter board [52], B200, B210 [70], and B205-mini. The data consisted of an IQ data file and JavaScript^{TM} object notation (JSON) formatted labeled meta data file, as before. These new dataset collects contain recorded training, validation and testing waveforms from multiple SDR models, as well as various sample rates.

For this research effort, all settings were automatic with respect to attaining the final sample rate. For automatic sample rate selection, only the sample rate was requested in the GNU radio flow graph. The decimation rate (DEC) for the B2xx SDRs is derived from the master clock rate (MCR) automatically chosen at run time. Table 4.1 illustrates the four MCRs that were observed during runtime, along with the calculated DEC. Similarly, the N210 SDR uses a fixed clock rate of $100\,MHz$, requiring different decimation rates. Table 4.2 has the derived decimation rates for each



(a) Transmitter/Receiver setup

Figure 4.1: Standard test setup

requested sample rate.

4.3 RF signal isolation

During OTA signal collection, it is important to choose operating frequencies that would minimize extraneous RF signals that are operating in the same recording bands. Any foreign in-band signal could cause problems with the feature extraction process, and contaminate the training, validation, and test data. Signal isolation is accomplished through ISM frequency band selection, along with received signal level (RSL) setup and hardware attenuation. Through most of the recordings, any in-band external signals were well below SNR threshold that would incur CNN model concerns (i.e., below the receiver's noise floor).

4.3.1 Test setup

The standard hardware setup is depicted in Fig. 4.1. The attenuators before the transmitter antenna varied from $0 - 40\,dB$, depending on the transmitter power. Also, the high powered $30\,dB$ attenuator before the receiver antenna is only used when high powered transmitters are used.

4.3.1.1 SNR

The CNN model is very sensitive to changes in SNR: A robust model would need a large diversity of training and validation data with many SNR levels. I limited that scope of change to about $10\,dB$

of SNR delta. In-line and programmable attenuators are used to maintain Received Signal Levels (RSL) between -55 to $-65\,dBm$ at the SDR performing the recording. From experience, $-60\,dBm$ approximates the RSL level that would be received from the average 5W PTT transceiver at $100\,m$ when operating OTA in the $2\,m$ and $70\,cm$ amateur radio bands in a clear line-of-site configuration. These power levels were chosen to provide ample SNR, provide realistic signal levels, and keep the received signal relatively *clean* and within good linear operating range of the SDR. The actual gain settings for the SDR were modest; zero to $20\,dB$ of gain was typically used. The manufacturer recommends gain values to be at least half of the maximum allowed values (i.e., $\sim 35dB$), in order to maintain dynamic range, but this was not feasible due to my recording setup and did not adversely affect my results.

4.3.2 Test transceivers

4.3.2.1 Chirp spread spectrum

The HopeRF RFM98 transceiver module is used for LoRa$^{\text{TM}}$ waveforms. This waveform features digital chirp spread spectrum technology for low data rate, but large processing gain requirements. Figures 3.7(a), and (b) illustrate these waveforms. Two bandwidths are selected for this research: 125, and $250\,kHz$. The transceiver module output power is rated at $20\,dBm$ and required no external attenuation and minimal attenuation and gain at the SDR receiver. A dedicated $5\,V$ power supply is used to power the transceiver.

4.3.2.2 Handheld Analog PTT

The Yaesu VX-8DR (5/2.5/1/.05 Watt), and Baofeng UV-5R (4/1 watt) are used to collect the NFM waveforms. These transceivers are reduced to their lowest power setting (i.e, .05 and 1 Watt respectively) and external attenuators are mounted to the *SubMiniature version A* (SMA) antenna connectors to achieve proper RSL values at the receiving SDR.

4.3.2.3 Handheld DMR PTT

The first handheld DMR, the TYT MD-380 (5/1 Watt), uses Time-Division Multiple-Access (TDMA) to double the number of users on a single $12.5\,kHz$ channel. The TYT waveform is depicted in Fig. 3.7 (d): Note that only one of the two available TDMA channels are used. The second one, Radioddity GD-55 (10/4 Watt), appears to use both TDMA time slots for a single user and is depicted in Fig. 3.7(e). In both transceivers, output power is set to the lowest value, and attenuation is adjusted similarly to that of the analog PTT radios.

4.3.2.4 Base station DMR PTT

The Yaesu FTM-400DR (50/20/5 Watt), depicted in Fig. 3.7(f), has four modes of operation: one analog and three digital modes. I used the DN mode: A $12.5\,kHz$ channel is split into two $6.25\,kHz$ adjacent channels. One channel is reserved for digital voice, while the other half of the channel is for data error correction. The digital modes are a proprietary standard that would not demodulate on either the TYT or GD55 transceiver, but both the TYT and GD-55 transceivers would demodulate on the FTM-400DR. The output power is set to $5\,Watts$, and input power is achieved using a dedicated $12\,V$ power supply. A high power attenuator $(30\,dB)$ is used at the receiver antenna, in conjunction with programmable low power attenuators before the SDR receiver.

4.3.2.5 ISM doorbells

The vodeson HD03 (Type MN21/23 $12\,V$ battery) and SadoTech model C (A23 $12\,V$ battery) ISM doorbells transmit power is well under $20\,dBm$. The OTA configuration required minimal attenuation and moderate gain at the SDR receiver.

Table 4.1: B2xx Auto sample rate selection

msps vs. MCR and DEC	MCR	DEC
$1\,msps$	$32\,MHz$	32
$5\,msps$	$40\,MHz$	8
$10\,msps$	$40\,MHz$	4
$25\,msps$	$25\,MHz$	1

Table 4.2: N210 Auto sample rate selection

msps vs. MCR and DEC	MCR	DEC
$1\,msps$	$100\,MHz$	100
$5\,msps$	$100\,MHz$	25
$10\,msps$	$100\,MHz$	10
$25\,msps$	$100\,MHz$	4

4.3.2.6 Key finder

The Click 'n Dig® model D2 (Type 27 $12\,V$ battery) transmit power is similar to the ISM doorbells. The OTA configuration required no attenuation and moderate gain at the SDR receiver.

4.3.2.7 Light switch

The LoraTap® transmitter (part number BA101KS-915E26WHI) is the only waveform from the $915\,Mhz$ ISM band and is illustrated in Fig. 3.7(j). Because the operating frequency is much higher than the $434\,Mhz$ ISM band, the SDR gain settings were increased to mitigate OTA path loss, and low transmitter power (two 2032 $3\,V$ *button* batteries) to achieve the $-60\,dBm$ RSL level target values.

$$\text{Sample Rate} \quad msps = \frac{MCR}{DEC} \tag{4.1}$$

4.4 Methods and Procedures

The feature extraction process that is laid out in section 3.4 is used to extract training and validation patches.

4.5 Patching

I record ten new waveforms per class for each phase of the supervised learning, i.e., training, validation, and testing for the four sample rates, $1, 5, 10$ and $25\,msps$. Then, approximately $1,000$ patches are collected per training waveform, totaling over $10,000$ patches per class, plus additional *noise* class training sample patches. This yields a total of about $362,200$ training samples. While $2,000$ patches per class are extracted for validation from the validation recordings, resulting in about $96,000$ validation sample patches. For one class/patch testing, I use 10 RF signals per class, where patching is done sequentially to capture all source-specific activities. For testing, the GUI and random patching process is not involved.

Like before, two additional classes were used to account for spectrogram areas that contained noise and the DC artifact. First, to carefully handle the effect of noise floor, I also used training patches from the noise area. There are discernible waveform artifacts in the noise floor throughout a given test file that can cause the models to classify *noise* patches incorrectly if they were not handled properly.

Extracting *noise patches* from the ten primary class waveforms significantly reduces incorrect classification during validation and testing. This means that I specifically extracted patches that appear to have no features from the primary signal classes. This results in a training set that is unbalanced, due to the significant increase in noise training patches, but a significantly improved validation and testing accuracy.

These extracted *noise patches* are then labeled as the *noise* class. In addition, DC artifacts are present at the center bin of the discrete Fourier transform due to the SDRs direct conversion receiver design [55] [56]. These two classes are dummy classes and should not be considered during testing as the primary signal classes of interest, although their explicit use in classification plays a significant role in improving the detection accuracy. The complete training sets then consist of twelve classes in total. Again, Each patch is saved as a 3D tensor as mentioned in section 3.6.2.

There are a total of $361,200$ patches used for the combined 1 and $25\,msps$ training to represent 12 classes, while the network batchsize was set to 64 patches. $49,200$ of these patches are for $1\,msps$ noise class, and $90,000$ were $25\,msps$ noise class. It is notable that 38.5% of all training data was to properly discern noise. This is due to the ML models miss-classifying noise as one of the ten primary classes. There are very low SNR (i.e., negative SNR) artifacts that are only detectable by the CNNs. The additional noise patches are harvested from non feature areas (i.e., noise) of the Click 'n Dig®, GD55, LoRa™ $125\,kHz$, LoRa™ $250\,kHz$, NFM, and YSF training data files. This was in addition to pure *noise* training data extracted features.

4.5.1 Computing Environment

The computing environment again consisted of using PyTorch 1.6 [53] using graphical processing unit (GPU) computing: dual RTX8000 (4608 NVIDIA® CUDA® cores) with 45GB of GPU memory.

4.6 Classification Methods (CNNs)

RF classification is performed in the frequency domain via STFT: All real and imaginary information was retained after transformation and used in a ResNet [58] based 2D CNN models. As before, the TF domain facilitated feature extraction for supervised model training.

4.6.1 CNN Model

Similar to chapter three 3.6.3, the network ResNet model with 50 hidden layers is adapted to four channels, and then limited the number of classes to twelve. However, in this case, each optimized model is run for 10-15 epochs for each of the three scenarios (i.e., 1 class/patch, 2 classes/patch, 3 classes/patch).

4.7 The basic training process with a multi-sampling rate treatment

I train the CNN model to work on four sample rates. This could have required a dataset consisting of all four sample rates, but for this effort, I choose two extreme sample rates, i.e., 1 and 25 $msps$. The classifier is then expected to interpolate the remaining in-between sample rates. The IQ data comes from several SDRs, with various accuracies, including different phase accuracies across the same hardware at different sample rates(see Chapter 5).

To train this classification system that estimates one class per patch, I use a softmax function in the output layer, which takes the flattened and dimension reduced feature as the input (Fig. 3.3 (d)). The flattening and dimension reduction step is required as the ResNet's bottleneck blocks do not reduce the feature dimension and can result in an unacceptably high dimensional feature vector. The most basic model assumes that there is only one class active per patch. I then use a softmax output layer for this basic model and Adam optimizer used with a default learning rate [59].

4.8 Data augmentation

Data augmentation is widely used, but simulating issues that COTS components may exhibit is not always possible. Instead, I try to variate the COTS RF transmitters and record them OTA to include a number of natural channel impairments (e.g., multi-path, noise, oscillator drift, IQ imbalance) into the data collection. The only purposely added augmentation was to the background noise floor. I injected additive white Gaussian noise to match the receiver's measured noise floor (i.e, $\sim -92dBm$) in order to assure that model would not "learn" from the static noise within a given tensor patch.

Additionally, COTS hardware often feature “built-in” artifacts or *augmentation* (e.g., noise figures, power amp third-order intermodulation distortion (IMD3), design implementation, and manufacturing process flaws) that could influence ML model classifiers.

4.9 Multi-class patch treatment

The advanced classification scenario is when multiple classes of RF signals are overlapping each other in the TF domain. The goal is to properly identify all those activities, although the task is more challenging than the basic single-class case, which is a straightforward classification problem. To address this case, I repurpose my training dataset and the model's final output in order to detect and classify up to three classes that operate withing varying overlap within the same patch.

4.9.1 Dataloader modification

First, the multi-class support requires modifications to the data preparation. Out of all 1 and 25 *msps* training patches, I combine up to three patches randomly selected from different classes. They are added in the TF domain as the four-channel tensor representation essentially retains all information coming from the IQ signals compared to the magnitude-only “spectrograms”. Although it is not natural, I deliberately allow patches to merge from both sample rates as it can expose the model to a greater variety compared to mixed patches of like sample rate data.

4.9.2 CNN modification

I also tweak the softmax output layer to properly address multi-class patches. Originally, a softmax layer predicts the posterior probability values $P(y = c|\boldsymbol{X})$ that sum to one: $\sum_{c=1}^{C} P(y = c|\boldsymbol{X}) = 1$, where c indicates one of the C class labels. Now, for multi-class handling, the model performs a class-wise logistic regression, which still outputs C different posterior probability values $P(y = c|\boldsymbol{X})$, but with no sum-to-one constraint. It means that each class can appear in the input patch with its own probability. In theory, all classes could be present and $\sum_{c=1}^{C} P(y = c|\boldsymbol{X}) = C$, although I expect up to three classes due to the training data preparation. Likewise, it is equivalent to say that I turned the model into performing C individual binary classification tasks, each of which detects the presence of the corresponding class.

4.9.3 Combining test files and labels

Once the combination for testing is determined, the entire test files are combined in the time domain to create a single IQ file. Then, the labeled integer classes are first converted to a binary equivalent, e.g., $[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ for $C = 2$, next a bitwise logical OR operation is applied with each binary label to produce a proper binary multi-class label. For a test signal with classes $C = \{2, 6, 8\}$ active, for example, the corresponding binary label vector is $[0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0]$.

I want to test up to three signals in a given patch, but this approach has a limitation in practice. There is no control over how close the waveforms are to each other when the IQ data files are combined, which occasionally leads to patches that are with only one or two classes that are co-channel.

The test-time classification accuracy is computed to take into account the multi-class presence within one patch. To test multi-class patches, I perform a bitwise `AND` operation between the predicted classes along with the binary label of the combined class test file. Finally, a threshold value for correct number of classifications is used to determine if multi-class classification was correct. The threshold chosen affects the overall test file classification; therefore, I provide a range of plotted accuracies vs. threshold in Fig. 4.10

Ultimately, to address the issues with combining IQ files and determine multi-class classification efficacy, I also examined the validation accuracy. During validation, individual patches with labels are combined then batch passed to the classifier, then compared against the predicted patch to determine correct classification. This is depicted in Fig. 4.3

4.10 Experimental Setup

4.10.1 Data collection process and format

4.10.1.1 Overview of the data collection process

IQ waveforms are recorded for ten seconds at four sample rates: 1, 5, 10 and 25 $msps$. Integer 16-bit IQ data from the SDR is saved as 32-bit floating-point representations for ML model processing. The OTA data is collected indoors with multiple reflections that provided natural multi-path augmentation. The average distance from transmitter to receiver is approximately four feet, within the far field for all hardware used. Attention is paid to minimize leakage paths around coaxial attenuation routes due to short physical Tx/Rx distances.

For higher sample rates (i.e., 5, 10, and 25 $msps$), I maintain the frequency resolution of 976.56 Hz by scaling up the DFT resolution to match higher sample rates. For example, for 5 $msps$, the original choice of 1024 frequency bins are multiplied by five. This means that the frequency bin resolution is $5\,MHz/(1024 * 5) = 976.56\,Hz$. The same method is used to scale up the 10 and 25 $msps$ frequency bins. I'm motivated to train on data from any sample rate, then inference on any alternate sample rate by keeping the temporal and frequency resolution relative. Although in theory this process must guarantee standardized patches regardless of their original sample rate, in practice the patches from different sample rate are considered different by the CNN models. Then, I train the model from the two extreme sample rates, 1 and 25 $msps$, and let it interpolate the other sample rates in-between during the test time. The investigation into this issue is detailed in Chapter 5.

4.10.1.2 Software

Similarly to section 3.6.2, All datasets collected consist of COTS transmitters, transceivers, and transceiver modules. I use open-source SDR software GNU's-not-Unix (GNU) radio [71] to record the labeled IQ waveform data to file.

4.10.1.3 Operating bandwidths

Similarly to chapter 3, GNU radio with out-of-tree module `SigMF` [28] is used to collect IQ waveforms from ten different types of COTS hardware that operate in and around the 434 MHz and 915 MHz ISM bands. Additionally, IQ data from push-to-talk (PTT) radios operating in the 70 cm U.S. band ($420 - 450\,MHz$) are also collected.

I focus on relatively narrow RF waveforms (i.e., channel bandwidth $\leq 250\,kHz$) in a spectrum space of up to 25 MHz of bandwidth. This means that I could have as little as one narrow-band waveform (i.e., 3 kHz channel width), or many waveforms within this frequency space. My approach is to not only classify one or more waveforms, but also to determine frequency and temporal parameters. I provide operating frequency results for this chapter, but the temporal relationships are inherent in this method and can be used to provide temporal metrics (e.g., hopping waveform timing along with channel resolution).

4.11 Experimental Results

4.11.1 Cross-sample-rate testing: single class per patch

Three different models are trained from three sample rate setups, i.e., 1, 25, and *mixed* (i.e., *mixed* $\equiv$ 1 and 25 $msps$). Each of these models are tested by four different kinds of test signals with four test-time sample rates, i.e., 1, 5, 10, and 25, i.e., there are test signals with mismatching sample rates. The results are illustrated in Figures 4.5, 4.6, and 4.7 as a confusion matrix, as before, where columns are the predicted classes while the rows are the ground truth. Therefore, all rows should add up to ten because there are ten test files per class. In cases that each row does not sum to ten, that means that the model incorrectly determined the test data belonged to the *noise* class or *DC center frequency* class. A perfect classification would be the case where all the diagonal elements are ten, which sum to 100.

4.11.1.1 Patch validation accuracy

Before discussion of test-time model performance, first the validation accuracy of the models are examined. Here, the validation process is on the set-aside patches annotated by the ground-truth class labels. I can then compute the patch-by-patch classification accuracy without involving the majority voting algorithm, which is only to consolidate results from multiple patches.

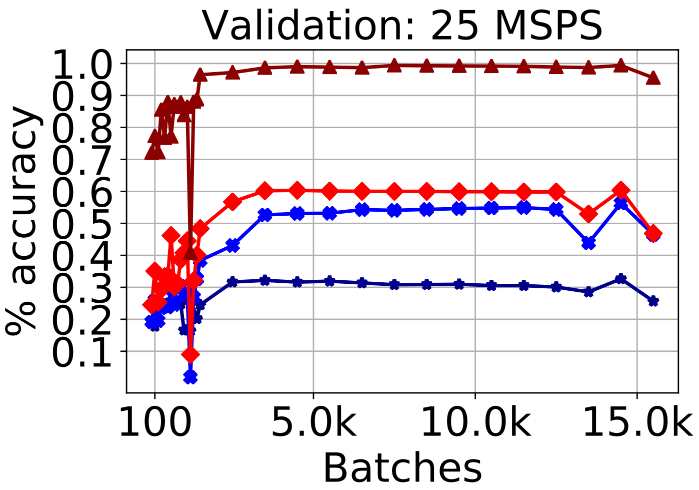
I examine the validation accuracy every 100 training batches for the first 1500, then every 1000 training batches thereafter. Early stopping is done by storing the model that gives the best validation accuracy and using it for testing.

Fig. 4.2 illustrates the validation accuracy of the three trained models. Fig. 4.2 (b) and (c) show the models trained from 1 and 25$msps$ examples, respectively, while (a) represents the model trained from the mixture of the two sample rates. First, in Fig. 4.2 (b), the validation accuracy from the matching sample rate, 1, reaches a near perfect accuracy after about $1,000$ batches. However, validation on the 5, 10, and 25 $msps$ data indicates accuracy between $\sim 20 - 63\%$, with decreased accuracy as the sample rate increases. Similarly, in (c), the model suffers from mismatching sample rates for the validation of 1, 5, and 10 $msps$ data. These two examples identify the issue of phase noise introduced by some devices as I discuss in Chapter 5.

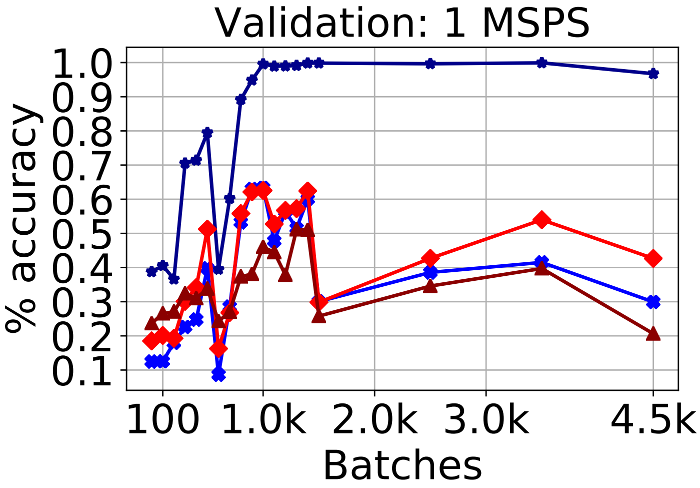
To address this issue, I propose to train the CNN model using two datasets recorded in the two extreme sample rates, 1 and 25. Then, I expect that the model interpolates the other sample rates in between the two. In 4.2 (a), the interpolated sample rates, 5 and 10 $msps$ demonstrate better performance with the mixed sample-rate strategy, with an accuracy over 80%.

4.11.1.2 Test accuracy

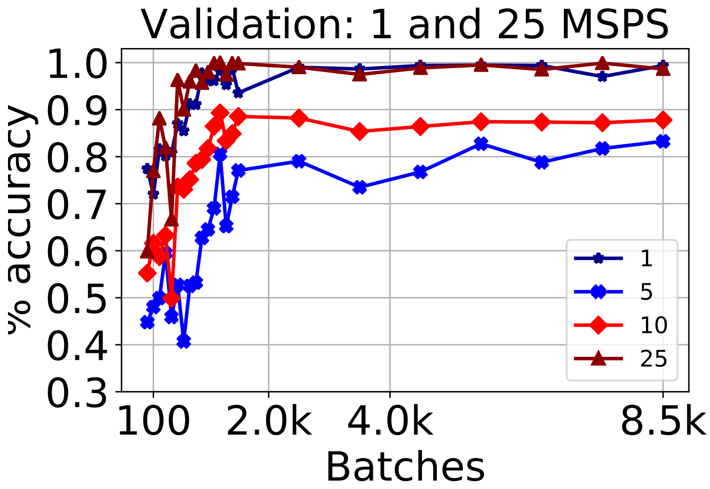
Fig. 4.5 depicts the test results from 1 $msps$ training. Here, the test was done on the whole test signal based on the majority voting algorithm that consolidates patch-by-patch classification



(a) 25 *msps* training



(b) 1 *msps* training



(c) mixed *msps* training

Figure 4.2: Validation accuracy for 1, 25, and mixed *msps* training

results. For this particular model, since it is trained from the lowest sample rate, the accuracy decreases as the test-time sample rate increases. Meanwhile, there is no clear relationship between miss-classifications and waveform bandwidth or modulation types. The model performs well on the matching sample rate, 1 *msps* Fig. 4.5 (a). On the interpolated sample rates (b)-(d), the model fails to adequately perform.

Fig. 4.6 depicts the results from 25 *msps* training. On the contrary to the 1*msps* case, the accuracy increases as the sample rate decreases. The 25 *msps* testing never achieved the accuracy that the validation results suggested were possible.

The proposed mixed sample rate training is shown in Fig. 4.7. The accuracy for all sample rates performs well when compared to the component rates of in Fig(s). 4.5, and 4.6. What is noticeable is that combining the training data for 1, and 25 *msps* patches significantly increases not only the interpolated rates of 5, and 10 *msps*, but also those matching sample rates. This indicates that the patches extracted from different sample rates increase the diversity within the training set, leading to a classifier robust to the variations caused by different devices and sample rates. Considering that the training data is derived from multiple SDRs and sample rates, it appears that the model is able to overcome many of the issues associated with phase noise (see Chapter 5).

4.11.2 Multi-class per patch

In addition to the single-class per patch case, I also examine the proposed multi-class per patch model's performance on the test signals that contain up to three classes.

4.11.2.1 Multi-class mixing with data loader

The method for training the classifier for multiple classes per patch requires combining patches using the data loader. The data loader first makes a random list of all training patches (i.e., 1, and 25 *msps*): For two classes per patch, the first half of the list is combined with its corresponding patch in the second half of the list resulting in half the original number of training patches. Similarly, for three classes per patch training, the data loader list is divided into thirds and combined. Accordingly, the training epochs is therefore approximately doubled and tripled to keep the training relative to each class per patch run.

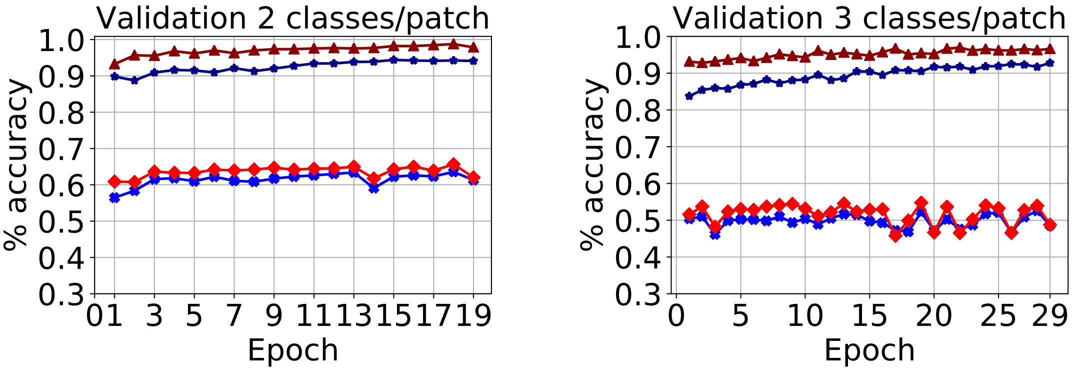
4.11.2.2 Patch validation accuracy

Given the observation that multiple sample rate training is useful, training patches are mixed from both 1 and 25 $msps$ training sets. This mixture of sample rates is not a scenario that would happen during validation, testing, or during live inferencing, but it produces better accuracy than when the training data is separated by sample rate.

Combining the validation patches for two and three classes per patch is similar to the way training data was combined, except that *only* like sample rates are combined. Fig. 4.3 illustrates the two and three patch/class validation accuracy vs. epoch. When I compute the validation accuracy, I consider all K classes that are present in the patch. For example, if there are {LoRa125, NFM, and YSF} in the patch, while the classifier predicts that {LoRa125, NFM, and Light} are there, it is a misclassification; the classifier knows the number of active classes. Note that I use a more sophisticated evaluation metric for test-time classification discussed in Sec. 4.11.2.3.

Fig. 4.3 (a) indicates 90% accuracy for two classes per patch for 1 and 25 $msps$ patches, with 5 and 10 $msps$ validation tightly coupled around 60% accuracy. Three patches per patch accuracy is illustrated in (b) and still indicates good classification for the mixed sample rate validation accuracy, while 5 and 10 $msps$ validation patch accuracy drops to 50 % accuracy. Note that the chance of correct classification is $1/\binom{10}{2} = 1/45$ and $1/\binom{10}{3} = 1/120$ when $K = 2$ and $K = 3$, respectively.

The validation patches are extracted using the previously discussed GUI process; therefore, each tensor patch contains features from exactly one waveform, that when combined with other validation patches, results in waveforms that are all occupying approximately 218.8 kHz (i.e., $224 \times 976.56\,Hz$) of bandwidth. This means that the validation scenario more rigorously tests the ML



(a) 2 class/patch

(b) 3 class/patch

Figure 4.3: Validation accuracy for 2 and 3 classes/patch scenario

models multi-class per patch classification capabilities. These patches are originally extracted randomly in frequency and time, allowing a greater freedom of variability of waveform overlapping in the frequency space versus when compared to combining IQ test files, where waveform frequencies often do not change during the IQ data collection process.

4.11.2.3 Test-time performance on multi-class per test signal examples

When I artificially mix signals at the patch level for training and validation, I can control the exact number of classes per patch. In the real-world test scenario, however, mixing occurs in the signal level. Therefore, although I still fix the number of classes within a test signal to be $K = 2$ or $K = 3$, an individual patch of the signal is not guaranteed to contain all K classes. It is because the IQ data files are very sparse in nature; the propensity for waveforms to actually combine such that three waveforms are always in the same patch is less likely. To test the accuracy of the classifier for this scenario, I create a *detection* mechanism based on counting and thresholding.

First, the classifier uses class-wise logistic regression as its last layer activation. In this way, the classifier predicts a binary vector, $\boldsymbol{y}_j \in \mathbb{R}^{10}$, for the j-th patch of the test signal. The sum of all the prediction vectors from the patches in the test signal yields a count vector $\boldsymbol{Y} = \sum_j \boldsymbol{y}_j$, whose value ranges from 0 (i.e., the class activity is not detected) to J (i.e., all patches contain the class). Note that I ignore the *noise* and *DC center frequency* classes.

Second, at the end of the test, for a given class of interest, c, if the sum of the total matched classes $\boldsymbol{Y}_c$ exceeds a pre-determined threshold τ, then this constitutes a *match* for the c-th class in the test file. The system's trade-off between the false positive and true positive ratios is sensitive to the choice of the threshold τ: if it is too low, too many classes will be detected although they do not exist (i.e., false positives), while a too high value can fail to detect important class activities. I vary this threshold to gauge the system's performance on the multi-class test examples.

I construct two sets of test signals. When $K = 2$, I combine $\binom{10}{2} = 45$ test signals that originate from 10 classes. For $K = 3$, the set amounts to 120 examples. This combination testing assures that every combination of the ten waveforms is tested.

Fig. 4.10 and Fig. 4.11 illustrate the models' receiver operating characteristics (ROC) for two- and three-classes-per-signal cases, as well as their area under the curve (AUC) scores. To compute each true positive ratio (TPR) and false positive ratio (FPR) pair, I try different τ values that result in different TPR-FPR value pairs. When I test two classes per signal $K = 2$, I can see from Fig. 4.10 that all four sample rates do well, with the general exception of YSF.

The YSF, GD55, and TYT waveforms all use a 4FSK based modulation. During single class per patch training, these three similar modulations are trained with a specific label, and the classifier does a good job of discerning the differences between these modulations (Sec. 4.11.1.2). However, during training for two and three classes per patch, the binary labels contain a class for each training patch. When multiple training patches for the 4FSK modulated waveforms are mixed together, the model does not know which label belongs to which mixed waveform, so it can start generalizing the relationship between these very similar classes.

There are also instances of training multiple classes that would only have one of these 4FSK waveforms mixed with a different class, which should help the model discern the difference between the 4FSK waveforms. Examining Fig. 4.10 plots (a)-(d) illustrates that of all the tested classes, the 4FSK modulations are exhibiting poor ROC curves, as well as much lower AUC scores. The model generally favors the TYT, and GD55 class over the YSF for each sample rate.

Fig. 4.11 illustrates the receiver's characteristics for the three classes per patch tests. I see similar class characteristics when compared to the two classes per patch results. However, the three classes per patch scenario expects the model to possibly discern the differences between all the 4FSK modulations in a single patch. During training there are significant patches that would contain all three classes with matching labels that can further blur the distinction between which of the three 4FSK waveforms belong to which label.

Additionally, the poor results for NFM in Fig. 4.11 (d) appear to illustrate that at a wide frequency bandwidth (e.g., $25\,msps$) lacks sufficient model generalization for the NFM waveform (e.g., the NFM waveform is typically $\sim 3\,kHz$).

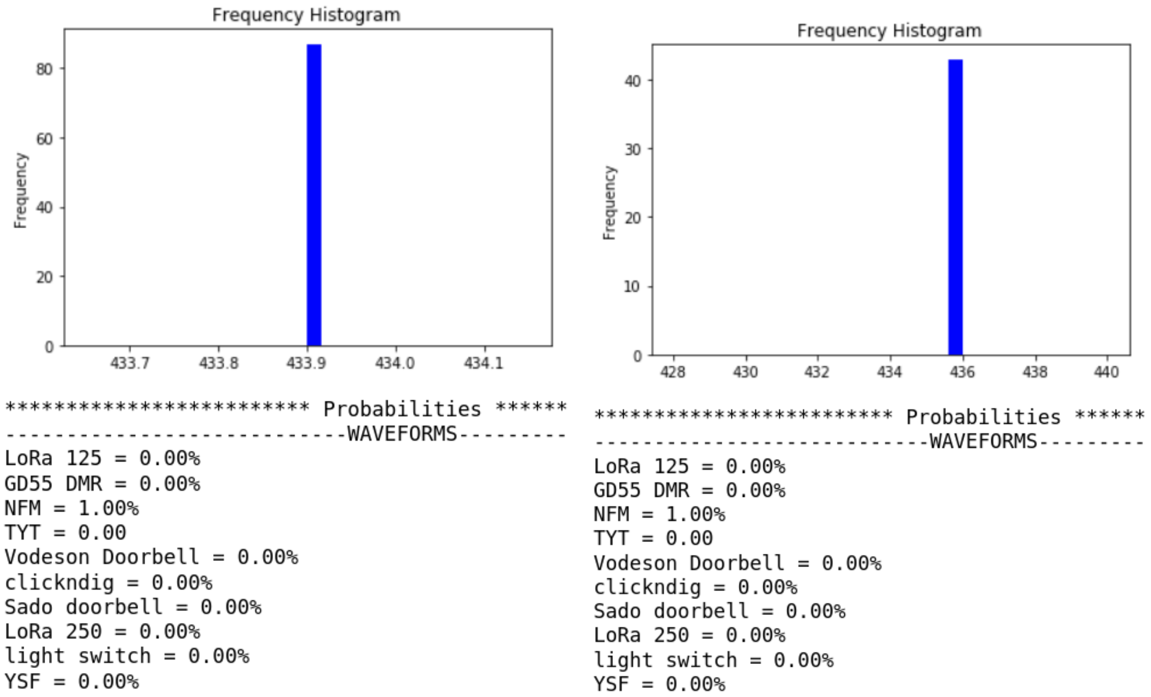
Overall, with the exception of the YSF waveform, the two and three class per signal scenarios show a robust ability for the models to properly classify the waveforms, depending on the receiver's thresholding value.

4.11.3 Waveform frequency occupation

In addition to the classification application, frequency domain classification and the patch method naturally provides the ability to determine the frequency of the classified waveforms within $\sim 1\,kHz$ of accuracy. It is a convenient feature that comes with TF domain processing, as once a patch is extracted, I know exactly which subbands it occupies. With successful classification, I can track back the frequency area that the same-class patches originate.

4.11.3.1 Multiple waveform co-located test cases

With RF analysis tools such as a real time spectrum analyzer, I can usually determine the correct type of waveform in a channel. However, if the channel has more than one waveform co-located, this becomes more difficult. I directly tested the classifier's ability to test several IQ data files that had two and three waveforms combined and operating in the same channel. For this test I used



```
************************ Probabilities ******
--------------------------WAVEFORMS---------
LoRa 125 = 0.00%
GD55 DMR = 0.00%
NFM = 1.00%
TYT = 0.00
Vodeson Doorbell = 0.00%
clickndig = 0.00%
Sado doorbell = 0.00%
LoRa 250 = 0.00%
light switch = 0.00%
YSF = 0.00%
```

(a) NFM 1 *msps*

```
************************ Probabilities ******
--------------------------WAVEFORMS---------
LoRa 125 = 0.00%
GD55 DMR = 0.00%
NFM = 1.00%
TYT = 0.00
Vodeson Doorbell = 0.00%
clickndig = 0.00%
Sado doorbell = 0.00%
LoRa 250 = 0.00%
light switch = 0.00%
YSF = 0.00%
```

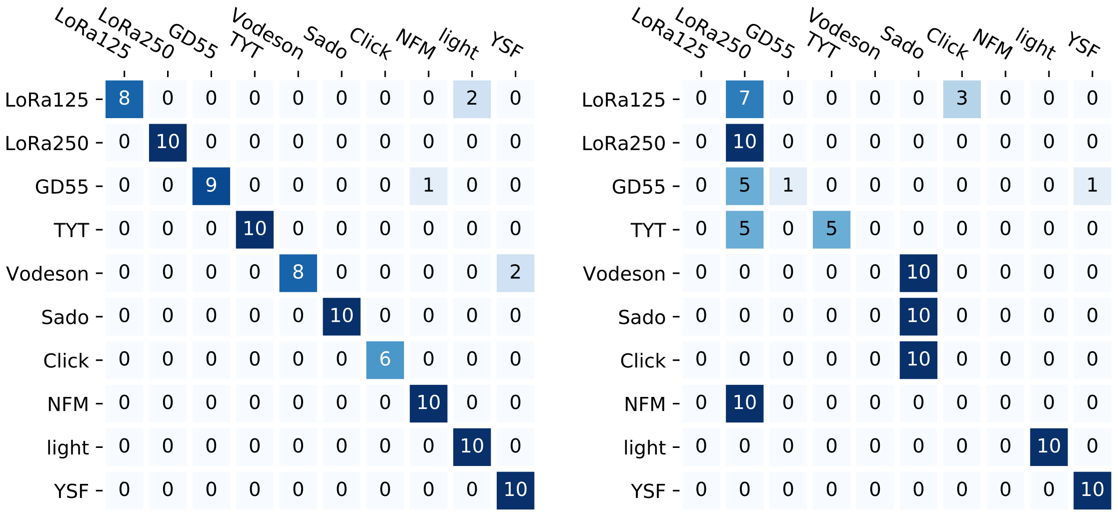
(b) NFM 25 *msps*

Figure 4.4: Operating frequency

1 *msps* test files and three different narrow bandwidth waveforms. Classified patches (excluding noise and center bin fft) were printed, along with the probable waveform. The results in Fig. 4.9 (a), and (b) indicate that the 4-channel STFT tensor method can provide accurate results from multiple waveforms per patch.

4.12 Chapter Conclusion

This chapter explored RF TF multi-channel classification using multiple sample rates and multiple hardware devices. I observed higher accuracy when using training data from a mixture of sample rates, and was able to interpolate additional sample rates for which the classifier had never been trained. Additionally, I showed that SDRs can exhibit levels of phase noise across different models, and across a particular model's sample rates. These differences show that a robust classifier benefits from a variety of SDR derived data sources, and SDR hardware.

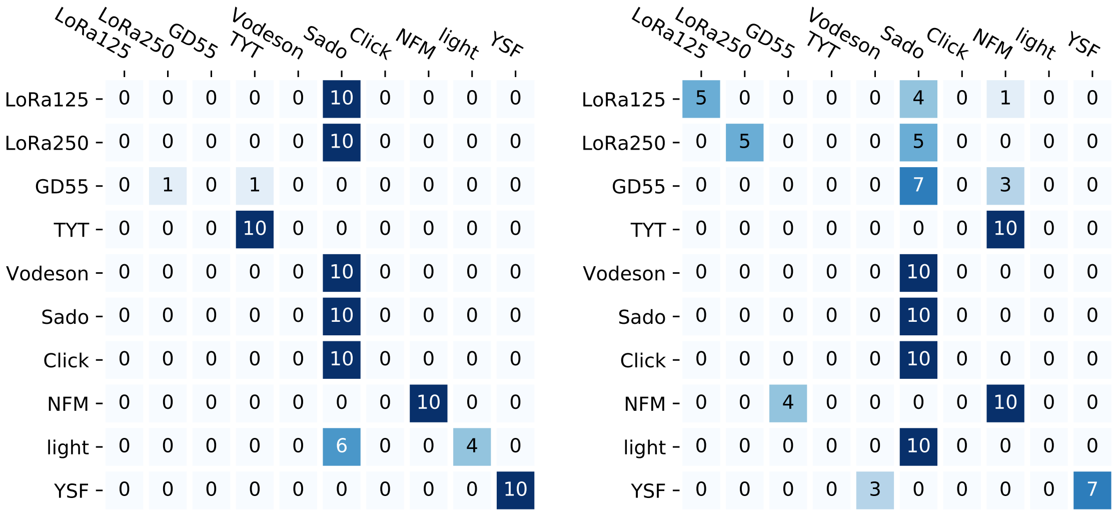


	LoRa125	LoRa250	GD55	TYT	Vodeson	Sado	Click	NFM	light	YSF
LoRa125	8	0	0	0	0	0	0	0	2	0
LoRa250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	9	0	0	0	0	1	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	8	0	0	0	0	2
Sado	0	0	0	0	0	10	0	0	0	0
Click	0	0	0	0	0	0	6	0	0	0
NFM	0	0	0	0	0	0	0	10	0	0
light	0	0	0	0	0	0	0	0	10	0
YSF	0	0	0	0	0	0	0	0	0	10

(a) $1\,msps$ 91%

	LoRa125	LoRa250	GD55	TYT	Vodeson	Sado	Click	NFM	light	YSF
LoRa125	0	7	0	0	0	0	3	0	0	0
LoRa250	0	10	0	0	0	0	0	0	0	0
GD55	0	5	1	0	0	0	0	0	0	1
TYT	0	5	0	5	0	0	0	0	0	0
Vodeson	0	0	0	0	0	10	0	0	0	0
Sado	0	0	0	0	0	10	0	0	0	0
Click	0	0	0	0	0	10	0	0	0	0
NFM	0	10	0	0	0	0	0	0	0	0
light	0	0	0	0	0	0	0	0	10	0
YSF	0	0	0	0	0	0	0	0	0	10

(b) $5\,msps$ 46%



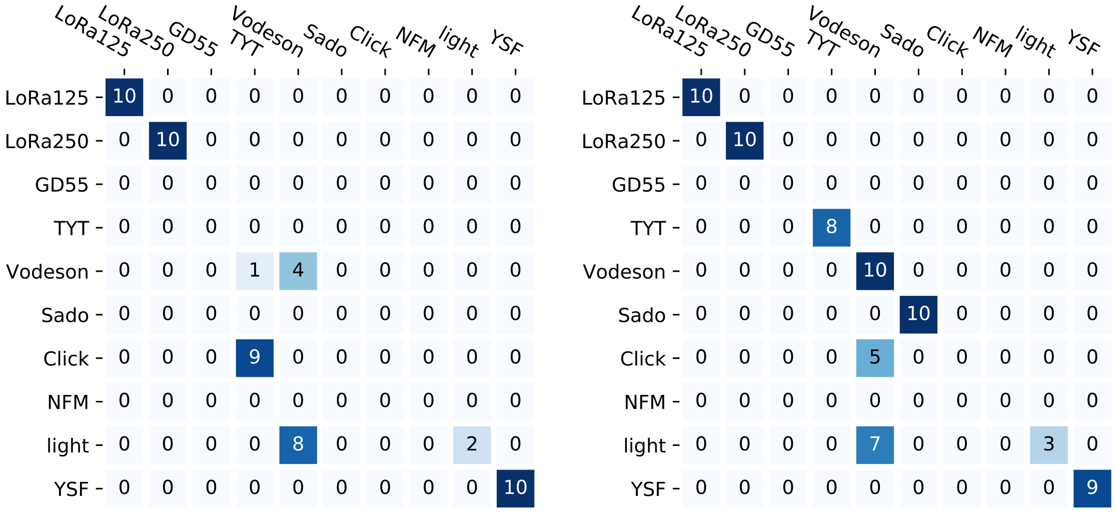
	LoRa125	LoRa250	GD55	TYT	Vodeson	Sado	Click	NFM	light	YSF
LoRa125	0	0	0	0	0	10	0	0	0	0
LoRa250	0	0	0	0	0	10	0	0	0	0
GD55	0	1	0	1	0	0	0	0	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	0	10	0	0	0	0
Sado	0	0	0	0	0	10	0	0	0	0
Click	0	0	0	0	0	10	0	0	0	0
NFM	0	0	0	0	0	0	0	10	0	0
light	0	0	0	0	0	6	0	0	4	0
YSF	0	0	0	0	0	0	0	0	0	10

(c) $10\,msps$ 44%

	LoRa125	LoRa250	GD55	TYT	Vodeson	Sado	Click	NFM	light	YSF
LoRa125	5	0	0	0	0	4	0	1	0	0
LoRa250	0	5	0	0	0	5	0	0	0	0
GD55	0	0	0	0	0	7	0	3	0	0
TYT	0	0	0	0	0	0	0	10	0	0
Vodeson	0	0	0	0	0	10	0	0	0	0
Sado	0	0	0	0	0	10	0	0	0	0
Click	0	0	0	0	0	10	0	0	0	0
NFM	0	0	4	0	0	0	0	10	0	0
light	0	0	0	0	0	10	0	0	0	0
YSF	0	0	0	0	3	0	0	0	0	7

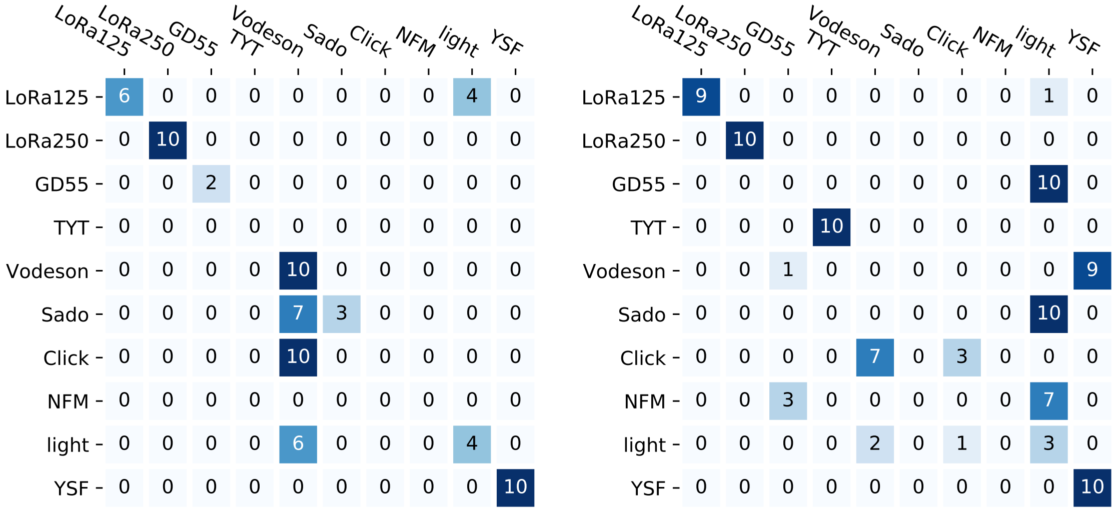
(d) $25\,msps$ 37%

Figure 4.5: $1\,msps$ training



(a) 1 *msps* 36%

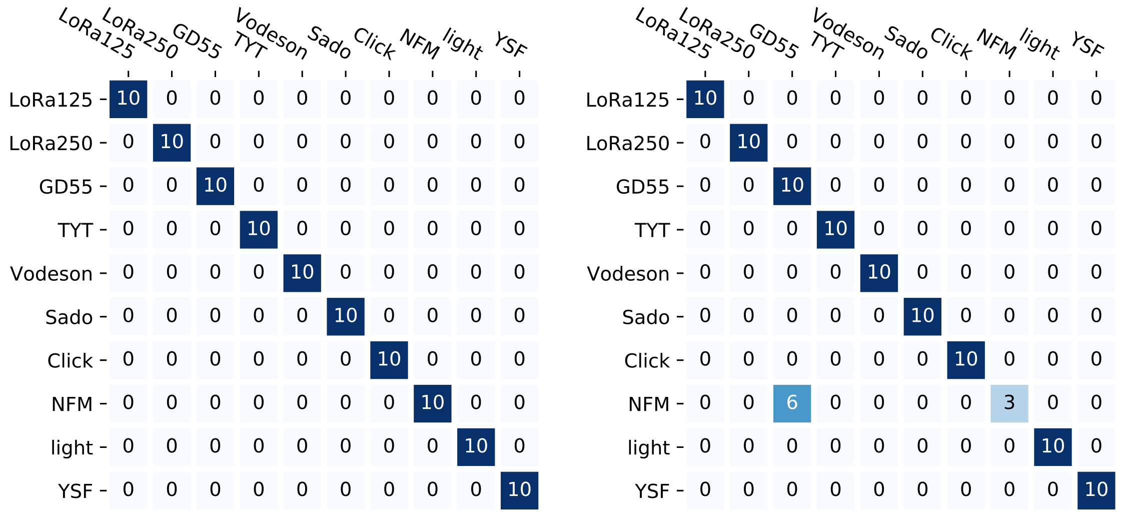
(b) 5 *msps* 60%



(c) 10 *msps* 45%

(d) 25 *msps* 45%

Figure 4.6: 25 *msps* training

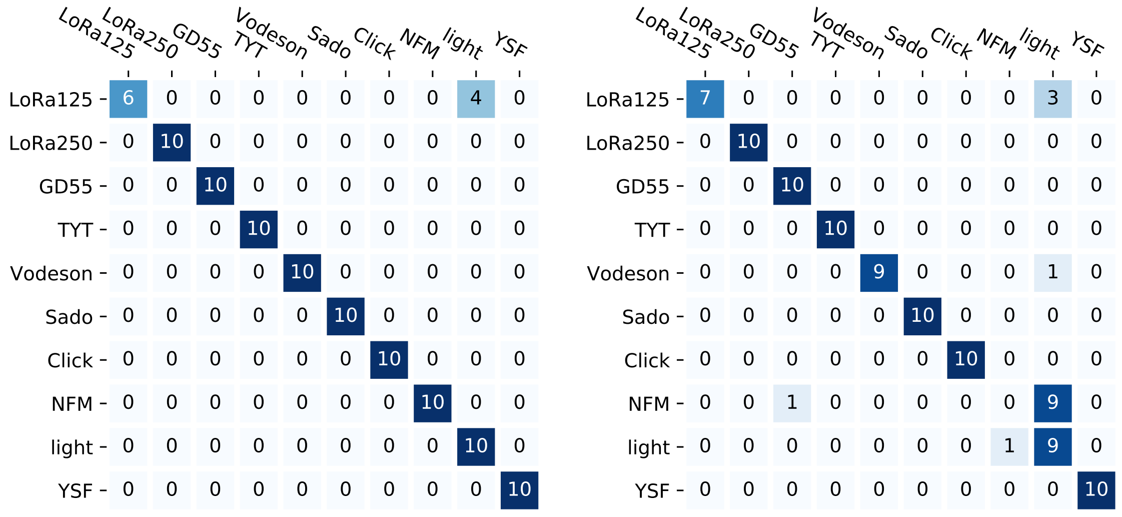


	LoRa125	LoRa250	GD55	TYT	Vodeson	Sado	Click	NFM	light	YSF
LoRa125	10	0	0	0	0	0	0	0	0	0
LoRa250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	10	0	0	0	0	0
Sado	0	0	0	0	0	10	0	0	0	0
Click	0	0	0	0	0	0	10	0	0	0
NFM	0	0	0	0	0	0	0	10	0	0
light	0	0	0	0	0	0	0	0	10	0
YSF	0	0	0	0	0	0	0	0	0	10

(a) 1 $msps$ 100%

	LoRa125	LoRa250	GD55	TYT	Vodeson	Sado	Click	NFM	light	YSF
LoRa125	10	0	0	0	0	0	0	0	0	0
LoRa250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	10	0	0	0	0	0
Sado	0	0	0	0	0	10	0	0	0	0
Click	0	0	0	0	0	0	10	0	0	0
NFM	0	0	6	0	0	0	0	3	0	0
light	0	0	0	0	0	0	0	0	10	0
YSF	0	0	0	0	0	0	0	0	0	10

(b) 5 $msps$ 93%



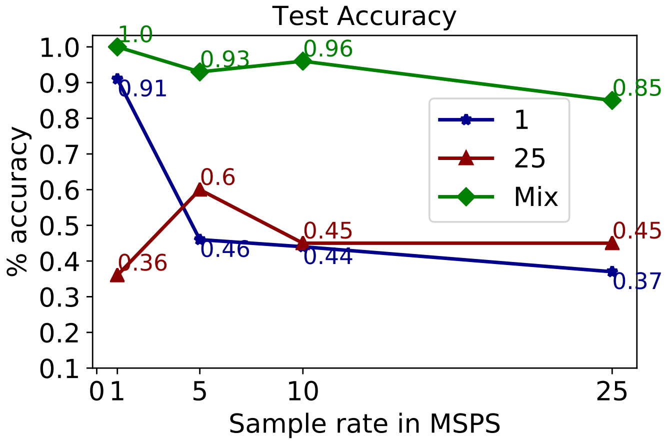
	LoRa125	LoRa250	GD55	TYT	Vodeson	Sado	Click	NFM	light	YSF
LoRa125	6	0	0	0	0	0	0	0	4	0
LoRa250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	10	0	0	0	0	0
Sado	0	0	0	0	0	10	0	0	0	0
Click	0	0	0	0	0	0	10	0	0	0
NFM	0	0	0	0	0	0	0	10	0	0
light	0	0	0	0	0	0	0	0	10	0
YSF	0	0	0	0	0	0	0	0	0	10

(c) 10 $msps$ 96%

	LoRa125	LoRa250	GD55	TYT	Vodeson	Sado	Click	NFM	light	YSF
LoRa125	7	0	0	0	0	0	0	0	3	0
LoRa250	0	10	0	0	0	0	0	0	0	0
GD55	0	0	10	0	0	0	0	0	0	0
TYT	0	0	0	10	0	0	0	0	0	0
Vodeson	0	0	0	0	9	0	0	0	1	0
Sado	0	0	0	0	0	10	0	0	0	0
Click	0	0	0	0	0	0	10	0	0	0
NFM	0	0	1	0	0	0	0	0	9	0
light	0	0	0	0	0	0	0	1	9	0
YSF	0	0	0	0	0	0	0	0	0	10

(d) 25 $msps$ 85%

Figure 4.7: Mix sample rate training



(a) ResNet50 Accuracy Vs. Sample rates

Figure 4.8: Test accuracy of 1 class/patch: Accuracy vs. sample rates

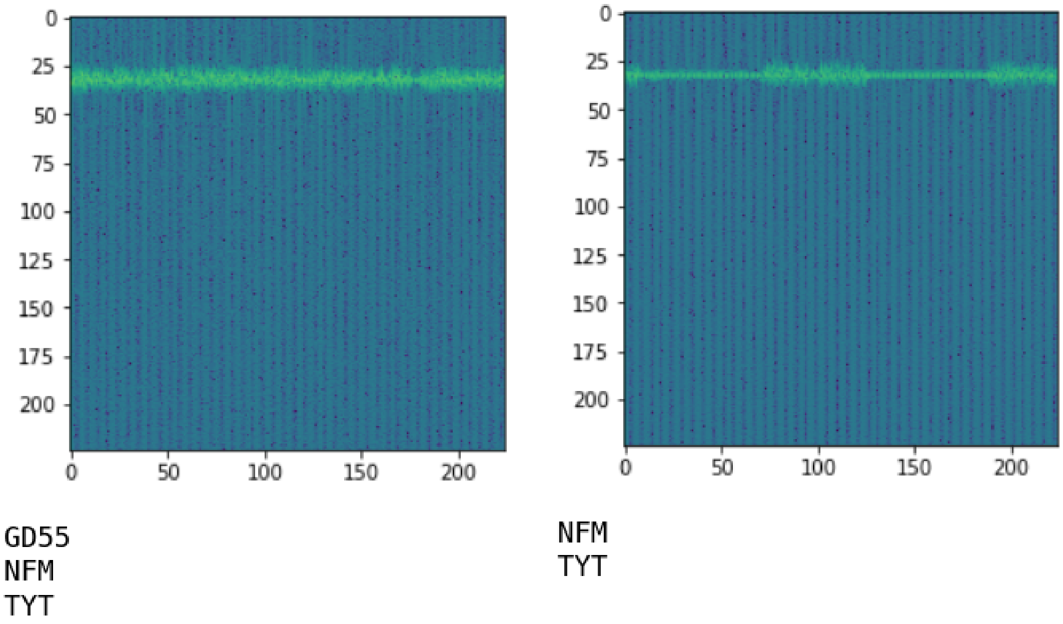
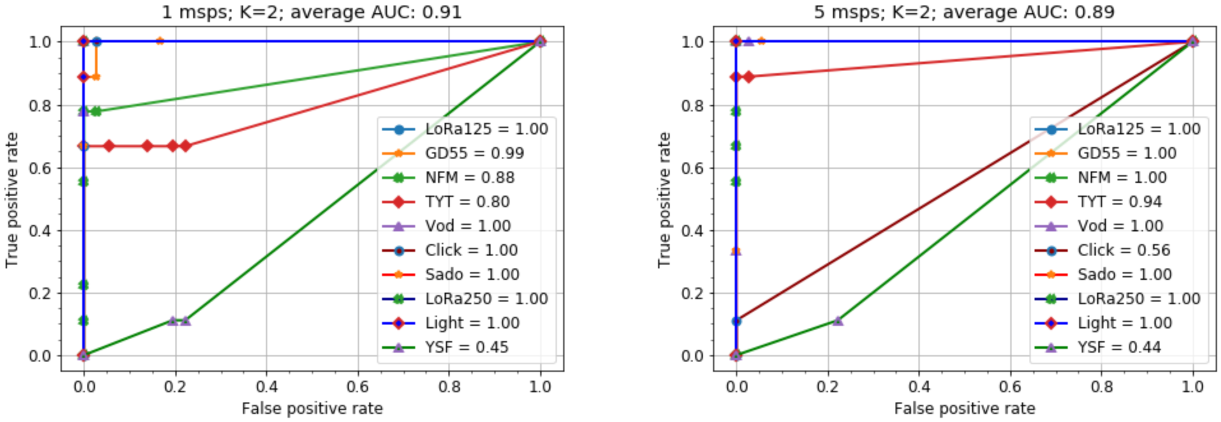
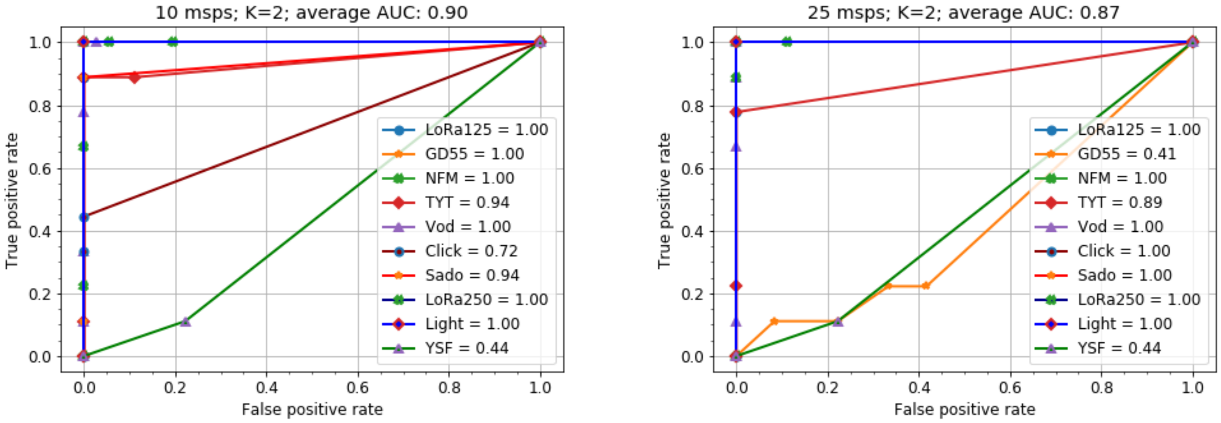


Figure 4.9: Correct classification 1 msps: patch co-channel

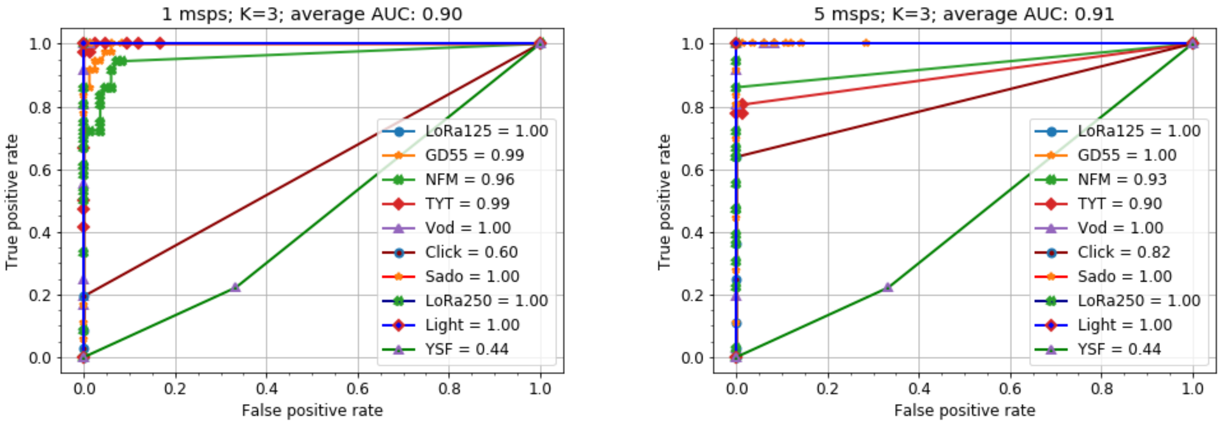


(a) $1\,msps$ (b) $5\,msps$



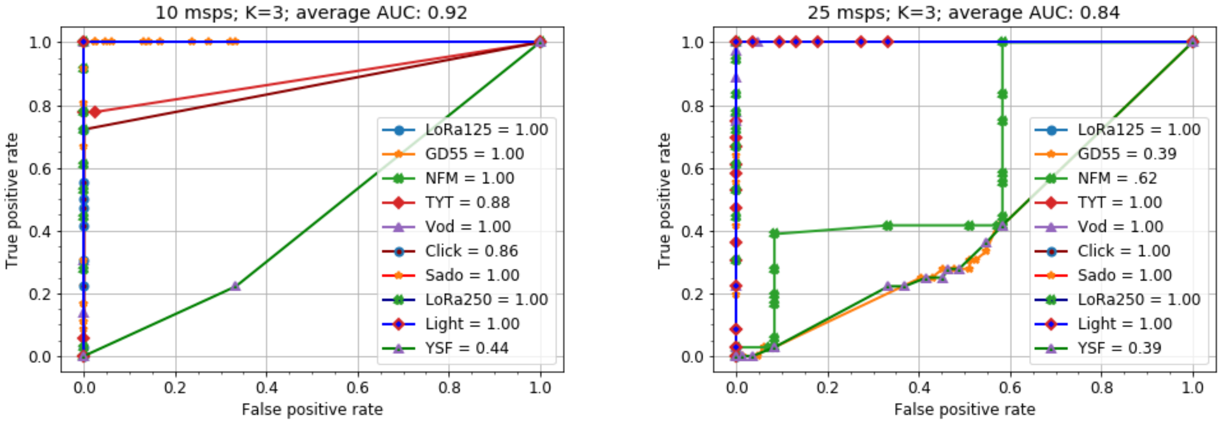
(c) $10\,msps$ (d) $25\,msps$

Figure 4.10: Roc for 2 classes/patch



(a) $1\,msps$

(b) $5\,msps$



(c) $10\,msps$

(d) $25\,msps$

Figure 4.11: Roc for 3 classes/patch

Chapter 5

IQ Phase investigation

Maintaining temporal and frequency bin resolution at different sample rates should allow the CNN to train at a given sample rate and inference on any subsequent chosen sample rate (i.e., train on $1\,msps$, and inference $25\,msps$). In practice, this does not provide adequate performance. I suspect time domain *aperture jitter* [72], and *oscillator jitter* was leading to frequency domain *phase noise* [73] [74], and thereby affecting the CNN performance across multi-sample rate domains. [75] specifically tested the phase noise between two of the SDRs that I use in my research (i.e., N210 and B205-mini) and noted different spectral spur locations, as well as differences in phase noise characteristics as the carrier wave (CW) offset frequencies increased. These differences could affect the classification results by training on one model of SDR, and inferencing on another. I mitigate some of the effects of phase noise through multiple sample rate training data, as well as collecting datasets from multiple SDR hardware models. Future research will investigate additional SDR hardware and IQ corrections at the SDR hardware.

5.1 Phase test setup

The phase relationship between SDR hardware and sample rates are examined. I chose a *GNU radio* generated sawtooth waveform, as well as a FMCW waveform generated by the LoRa$^{\mathrm{TM}}$ RF transceiver module as baseline waveforms for analysis. The angle between I and Q of a sawtooth waveform can be used to plot a subsequent sawtooth waveform. This principal is used to locate distortion caused by the SDRs. A SDR is used to transmit a *GNU radio* created flowgraph, $1\,kHz$, 3 amplitude sawtooth waveform at $433.7\,Mhz$. The transmitting SDR is connected to the receiving SDR via a closed-loop coaxial setup. Attenuation and gain are adjusted as necessary to maintain $-50\,dBm$ at the receiver. Spectrally, the test sawtooth, and LoRa$^{\mathrm{TM}}$ waveforms are anchored at

$F_c + BW/4$ for each SDR receiver sample rate, where F_c is the center frequency of the receiver, and BW is the bandwidth of the sample rate (i.e., $1\,msps \equiv 1\,MHz$ of bandwidth).

The recorded IQ time domain data is then read into *Python numpy* arrays as complex data (i.e., I+jQ), then the function *angle* is used to convert each IQ data file into an array of angles in radians. The plots depict the phase relationship between I and Q in the complex domain.

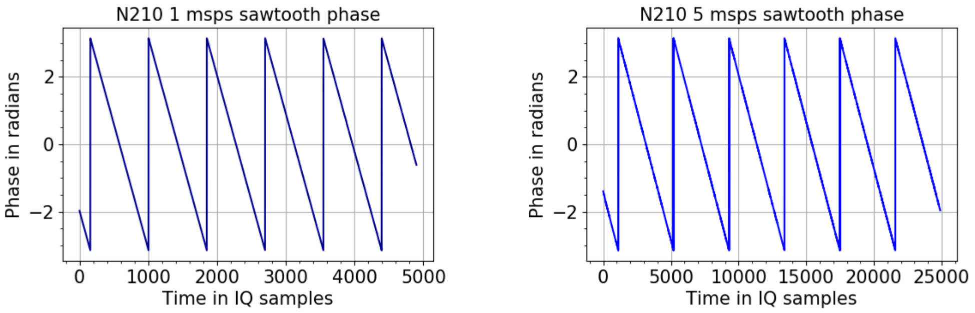
Since I'm depending on a SDR to transmit the *GNU radio* sawtooth, I also compared phase relationships against the LoRa^{TM} RF transceiver module. This module produces FMCW waveforms that shift their start frequency to denote a different symbol. I expected the FMCW waveform to produce results that would be similar to those of the standard *GNU radio* sawtooth waveform, but mitigate any possible bias from only using a SDR for both transmit and receive results.

5.2 N210

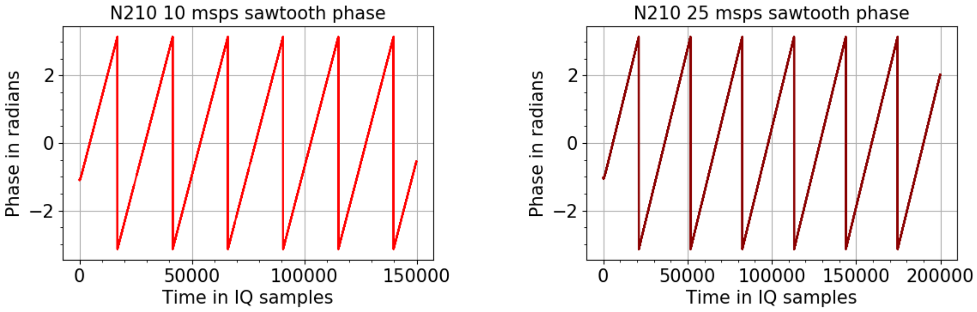
The N210 accuracy is measured and compared in [76] and [77]; The results from that test indicate that the N210 should produce good phase accuracy results. When I used the N210 in my testbed, this SDR produced results that were consistent across multiple sample rates. The phase plots in Fig. 5.1 and Fig. 5.3 are expected, with the exception of phase reversal when (a) and (b) are compared to plots (c), and (d) in both phase plots. The sawtooth and LoRa^{TM} constellation diagrams in Fig. 5.2, and Fig. 5.4 are each plotted against a noise collect from matching sample rates. The constellations are similar between sawtooth and LoRa^{TM}.

5.3 B210

The B210 SDR features an automatic adjustable master clock. The phase plots in Fig.5.5 and 5.7 show a number of differences between (a)-(d). The time scale on all B210 plots is kept the same for comparison. The B210 constellation diagrams are shown in Figures 5.6 and 5.8 and each plotted against a matching noise collect at their associated sample rate. The sawtooth constellations are

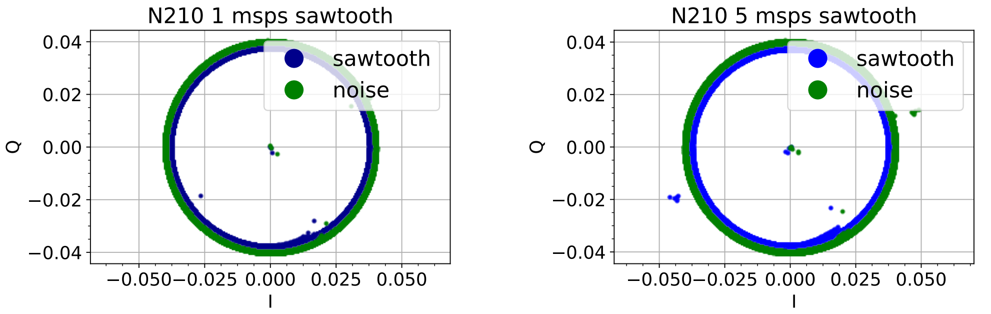


(a) $1\,msps$ (b) $5\,msps$

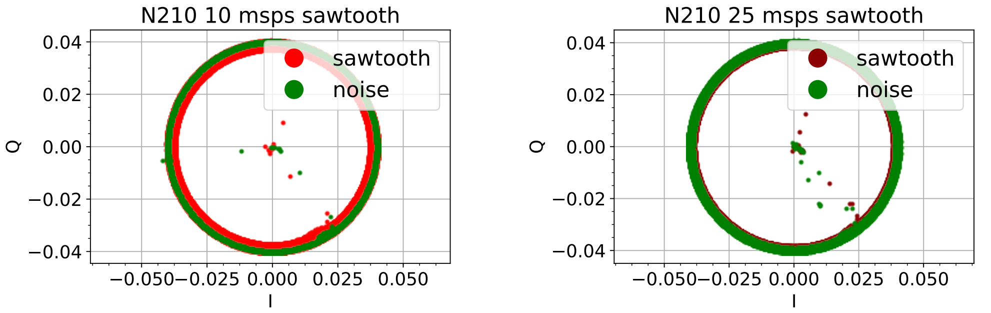


(c) $10\,msps$ (d) $25\,msps$

Figure 5.1: N210 $1\,kHz$ sawtooth IQ phase

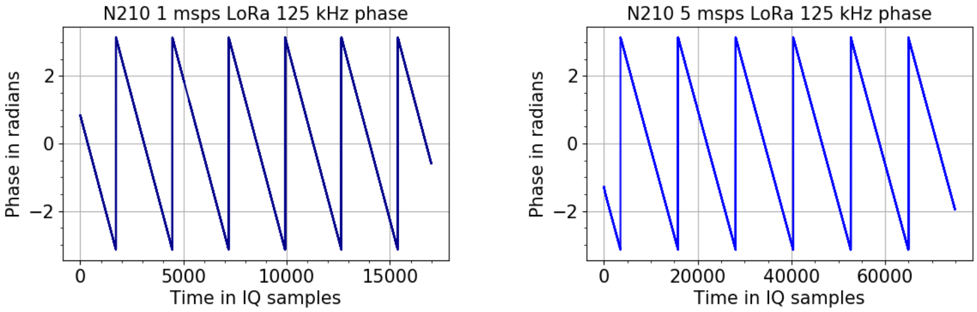


(a) $1\,msps$ (b) $5\,msps$

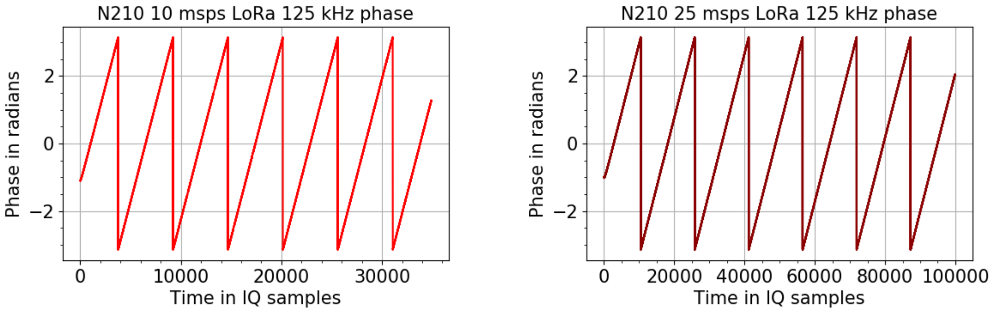


(c) $10\,msps$ (d) $25\,msps$

Figure 5.2: N210 $1\,kHz$ sawtooth IQ constellation



(a) 1 $msps$ (b) 5 $msps$



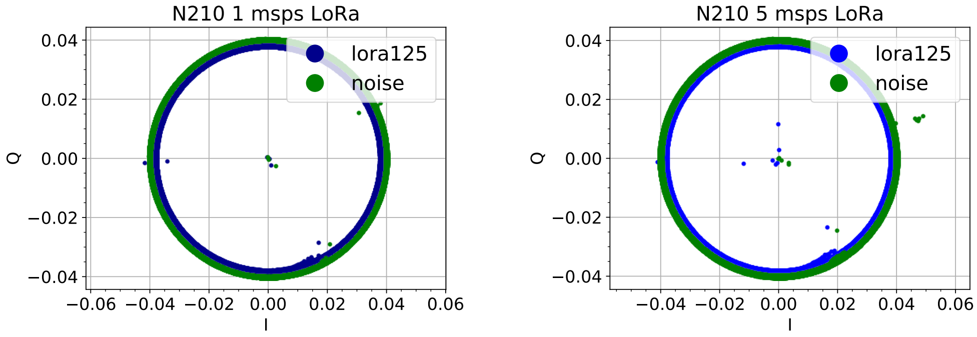
(c) 10 $msps$ (d) 25 $msps$

Figure 5.3: N210 LoRa 125 KHz IQ phase

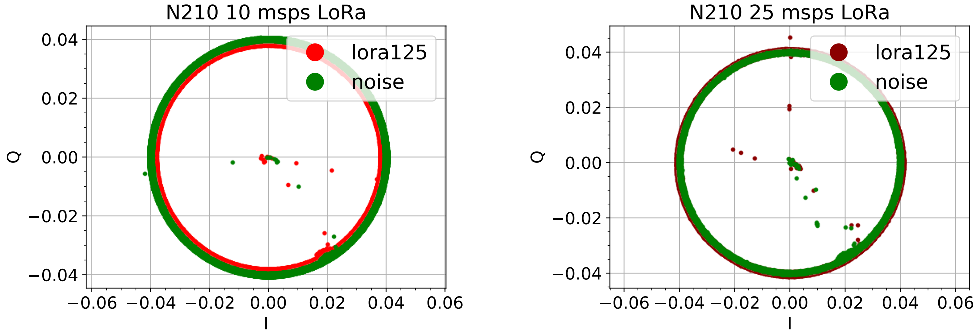
notably different from the LoRa^{TM} constellation and each constellation plot consists of only enough samples to depict a good constellation representation.

5.4 B205

The B205 SDR also features an automatic adjustable master clock. Similar to the B210, the phase plots in Fig.5.9 and 5.11 show a number of differences between (a)-(d). The time scale on all B205 plots was kept the same for comparison. The B205 constellation diagrams are shown in Figures 5.10 and 5.12, and each plot also contains a noise collect from the associated sample rate. As reported with the B210, the sawtooth constellation different from the LoRa^{TM} constellation and each constellation plot consists of only sufficient samples to produce a good constellation.

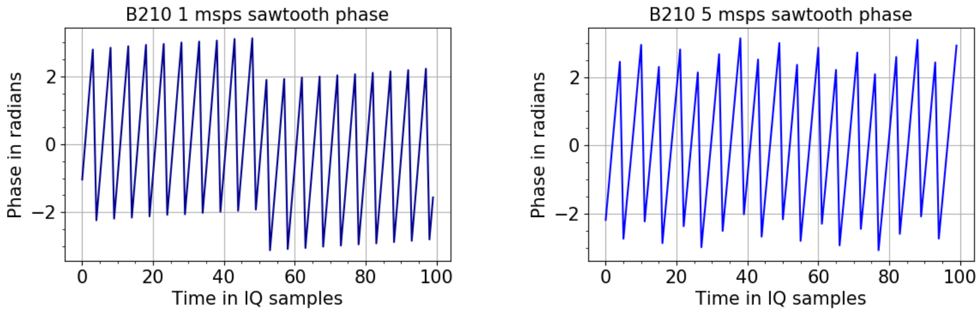


(a) $1\,msps$ (b) $5\,msps$



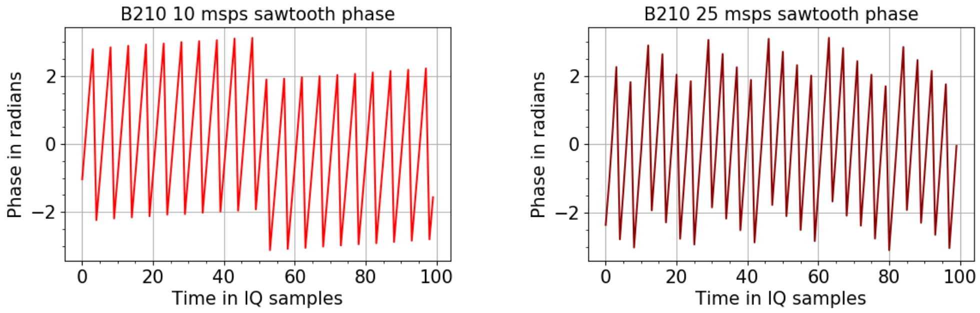
(c) $10\,msps$ (d) $25\,msps$

Figure 5.4: N210 LoRa 125 KHz IQ constellation



(a) $1\,msps$

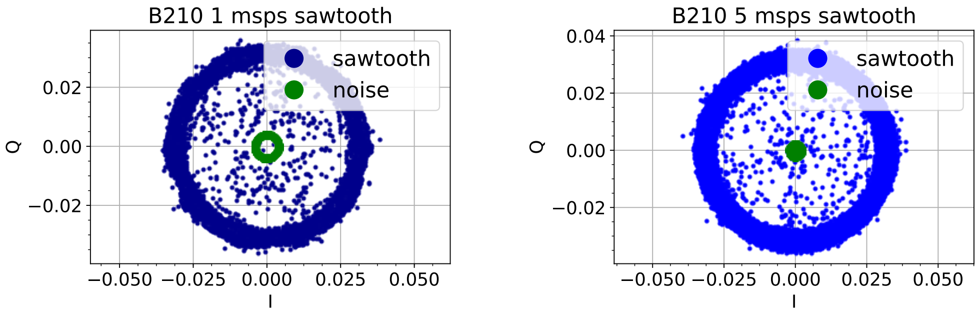
(b) $5\,msps$



(c) $10\,msps$

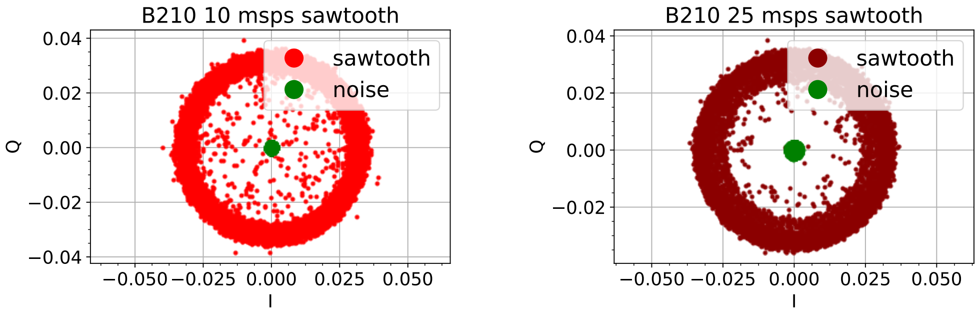
(d) $25\,msps$

Figure 5.5: B210 $1\,kHz$ sawtooth IQ phase



(a) $1\,msps$

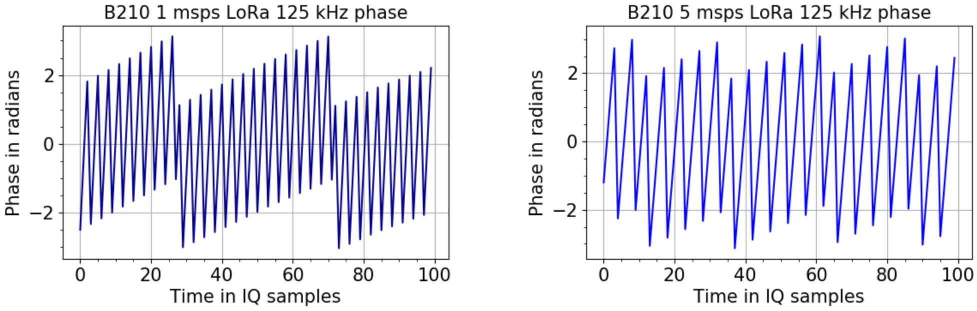
(b) $5\,msps$



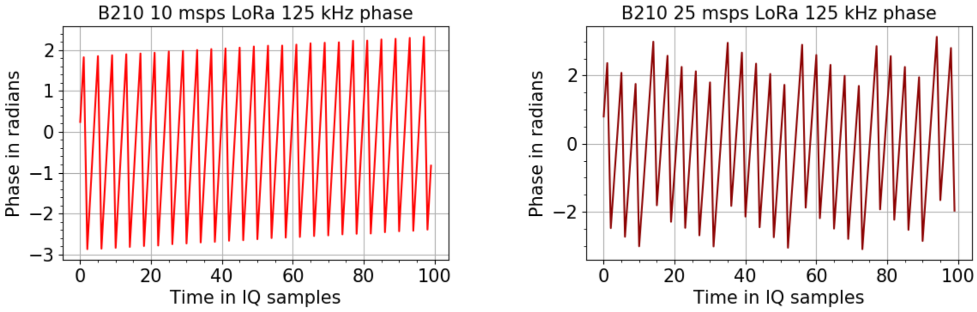
(c) $10\,msps$

(d) $25\,msps$

Figure 5.6: B210 $1\,kHz$ sawtooth IQ constellation

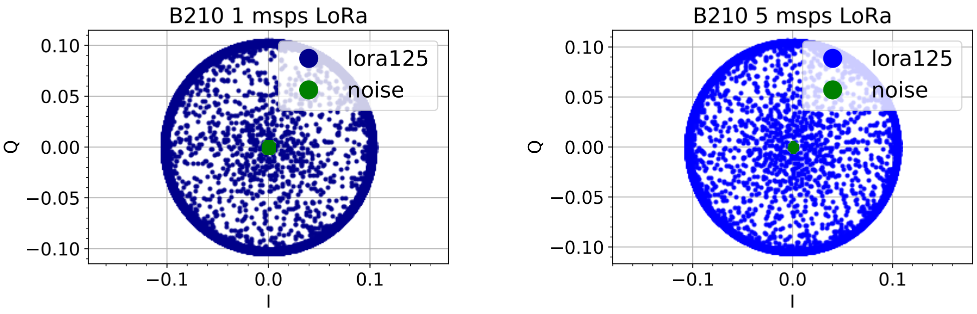


(a) $1\,msps$ (b) $5\,msps$

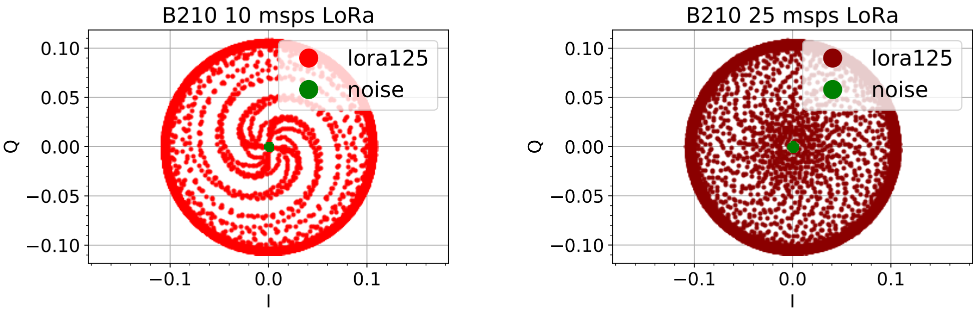


(c) $10\,msps$ (d) $25\,msps$

Figure 5.7: B210 LoRa $125\,KHz$ IQ phase

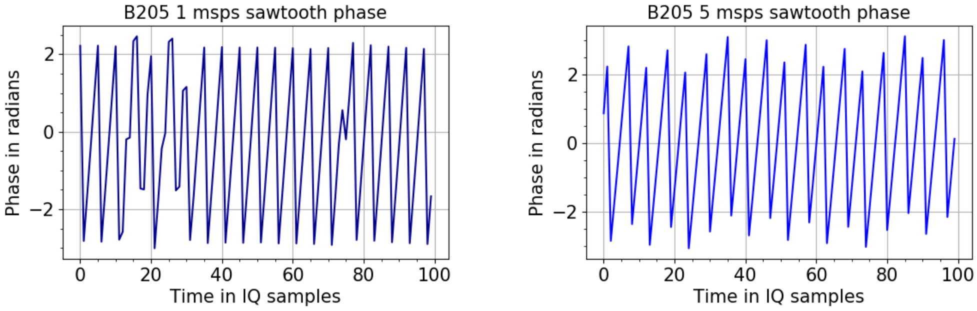


(a) $1\,msps$ (b) $5\,msps$

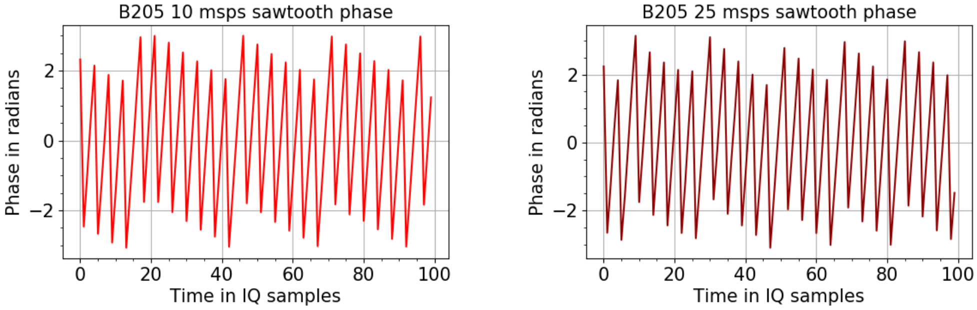


(c) $10\,msps$ (d) $25\,msps$

Figure 5.8: B210 LoRa $125\,KHz$ IQ constellation

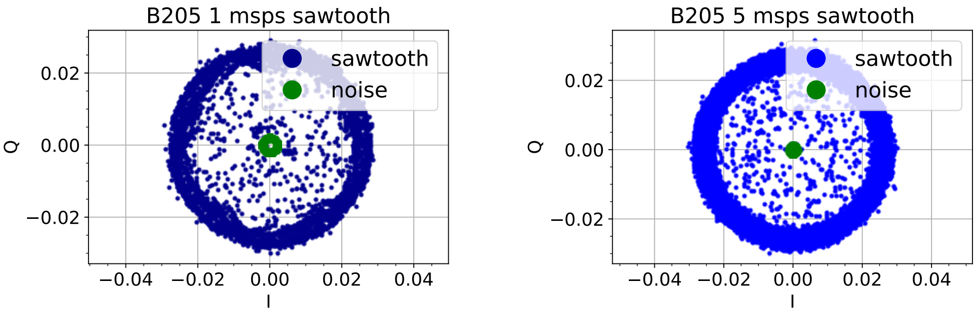


(a) $1\,msps$ (b) $5\,msps$

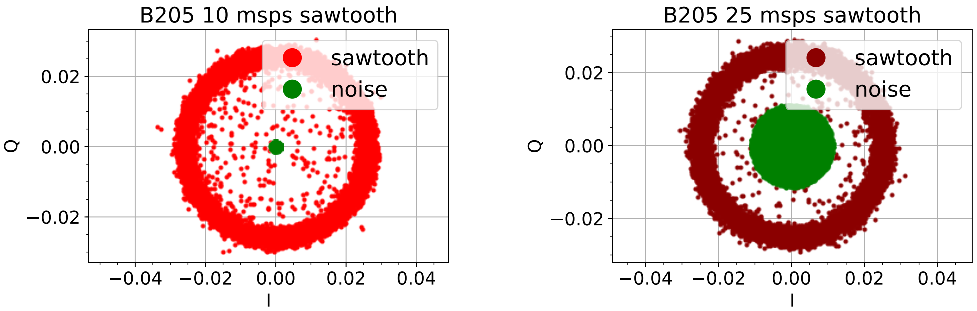


(c) $10\,msps$ (d) $25\,msps$

Figure 5.9: B205 $1\,kHz$ sawtooth IQ phase

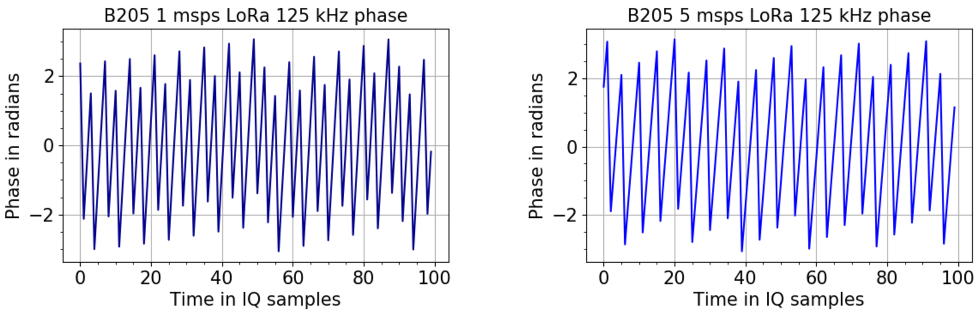


(a) $1\,msps$ (b) $5\,msps$

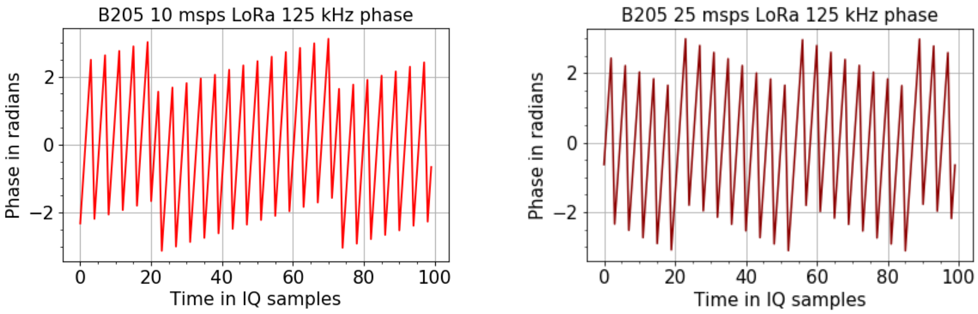


(c) $10\,msps$ (d) $25\,msps$

Figure 5.10: B205 LoRa $125\,KHz$ IQ constellation

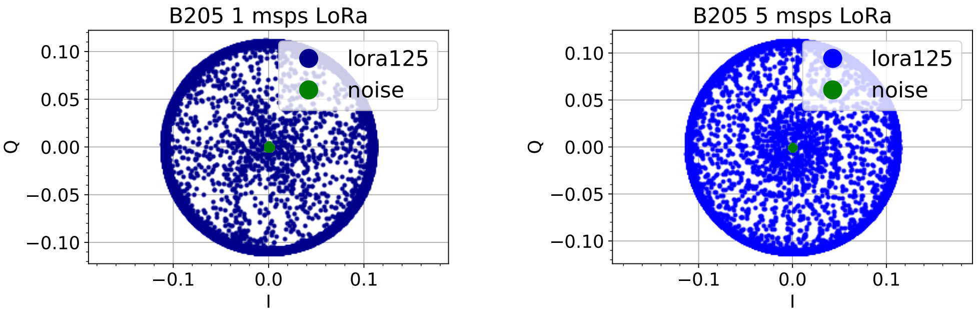


(a) 1 $msps$ (b) 5 $msps$

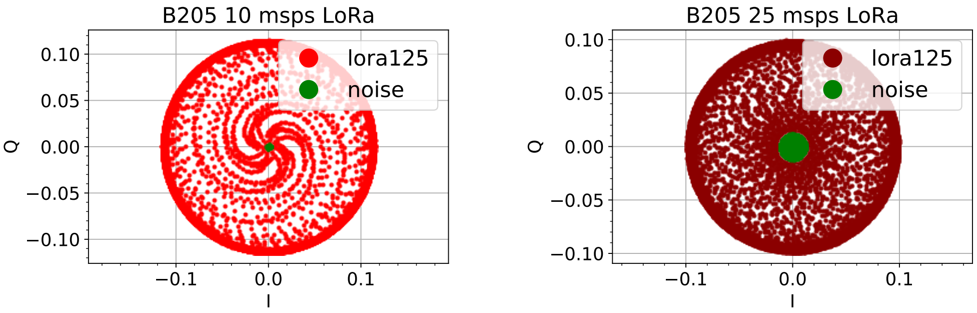


(c) 10 $msps$ (d) 25 $msps$

Figure 5.11: B205 LoRa 125 KHz IQ phase



(a) 1 $msps$ (b) 5 $msps$



(c) 10 $msps$ (d) 25 $msps$

Figure 5.12: B205 LoRa 125 KHz IQ constellation

5.5 SDR phase conclusion

A significant number of differences between the N210, B210, and B205 phase plots and constellation diagrams exist. The number of samples used to create the phase angles between the N210 and B2xx SDRs are significant. Between $5\,k$, and $200\,k$ samples are used to create a good sawtooth phase plot for the N210, while only about 100 are used for the B2xx radios. The differences in scale could affect the model's ability to discern differences between trained data from the N series SDR to validation and test data from the B series SDR and vice versa. Another issue is the difference in phase angles between sample rates of the B series SDRs. The B series phase angles are not always properly varying between -2π to 2π when compared to the N series SDR. These differences in phase angles could also affect the model's ability to properly discern the correct class when training, since validation and testing data are not always from the same sample rate or SDR model.

The constellations diagrams between the N series and B series SDRs are notable in a number of aspects. I expect the constellations to be different between the sawtooth and LoRa$^{\text{TM}}$, but the N series doesn't show much difference. The constellation diagrams between the N series and B series are quite striking. Although the B series constellations better depict my expectations, the N series indicate a tighter constellation structure. Their are also a notable difference in the constellations across different sample rates for the B series SDRs. These constellation differences could certainly affect the CNN classification and could be caused by IQ imbalance, in addition to differences in how the SDR manages the IQ data across different sample rates.

Chapter 6

Conclusion

In chapter 1 I presented the problems with modern RF allocation bands and how ISP can help bring order to this important communication domain. To harness the data in the RF domain, I established in chapter 2 how to use all the RF domain data and how RF data can be successfully manipulated using informatics techniques. Following this, in chapter 3, I validated the use of *all* the information contained in the four-channel tensor to provide better classification accuracy versus only using the spectrogram magnitude. In chapter 4, my RF classification method was used to train a CNN system to successfully classify RF signals from trained and un-trained sample rates using multiple SDR hardware models to improve classifier generalization. Next, in chapter 5, I showed that a robust classifier can benefit from multiple sample rates and multiple SDR hardware because of differences in hardware time domain jitter and frequency domain phase noise. In conclusion, I have demonstrated that an open-source RF domain classification system can provide robust physical layer multi-sample rate and multi-device classification.

Bibliography

[1] Spectrum management. National Telecommunications and Information Administration. [Online]. Available: https://www.ntia.doc.gov/category/spectrum-management

[2] "Industrial, Scientific, and Medical Equipment," June 2015, code of Federal Regulations. Title 47, Chapter I, Subchapter A, Part 18.

[3] S. Bhattarai, J. J. Park, B. Gao, K. Bian, and W. Lehr, "An Overview of Dynamic Spectrum Sharing: Ongoing Initiatives, Challenges, and a Roadmap for Future Research," *IEEE Transactions on Cognitive Communications and Networking*, vol. 2, no. 2, pp. 110–128, 2016.

[4] S. K. Sharma, T. E. Bogale, L. B. Le, S. Chatzinotas, X. Wang, and B. Ottersten, "Dynamic Spectrum Sharing in 5G Wireless Networks With Full-Duplex Technology: Recent Advances and Research Challenges," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 674–707, 2018.

[5] W. S. H. M. W. Ahmad, N. A. M. Radzi, F. S. Samidi, A. Ismail, F. Abdullah, M. Z. Jamaludin, and M. N. Zakaria, "5G Technology: Towards Dynamic Spectrum Sharing Using Cognitive Radio Networks," *IEEE Access*, vol. 8, pp. 14 460–14 488, 2020.

[6] A. Sharmila and P. Dananjayan, "Spectrum sharing techniques in cognitive radio networks – a survey," in *IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019, pp. 1–4.

[7] "Citizens Broadband Radio Serice (CBRS)," June 2015, code of Federal Regulations. Title 47, Chapter I, Subchapter D, Part 96.

[8] CBRS service. Select Spectrum. [Online]. Available: https://www.selectspectrum.com/available-spectrum/broadband/cbrs

[9] G. Frank, "Pulse code communication," U.S. Patent 2 632 058, March, 1953. [Online]. Available: https://www.freepatentsonline.com/2632058.html

[10] R. D. Badger, K. H. Jung, and M. Kim, "An Open-Sourced Time-Frequency Domain RF Classification Framework," in *29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1701–1705.

[11] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[12] C. Moy and J. Palicot, "Software radio: a catalyst for wireless innovation," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 24–30, 2015.

[13] C. Belisle, V. Kovarik, L. Pucker, and M. Turner, "The software communications architecture: two decades of software radio technology innovation," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 31–37, 2015.

[14] A. M. Wyglinski, D. P. Orofino, M. N. Ettus, and T. W. Rondeau, "Revolutionizing software defined radio: case studies in hardware, software, and education," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 68–75, 2016.

[15] W. H. Tuttlebee, *Software defined radio: enabling technologies.* John Wiley & Sons, 2003.

[16] A. M. Wyglinski, D. P. Orofino, M. N. Ettus, and T. W. Rondeau, "Revolutionizing software defined radio: case studies in hardware, software, and education," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 68–75, 2016.

[17] J. Lee, E. Hyun, and J.-Y. Jung, "A simple and efficient IQ data compression method based on latency, evm, and compression ratio analysis," *IEEE Access*, vol. PP, pp. 1–1, Aug. 2019.

[18] D. Muir, L. H. Crockett, and R. W. Stewart, "Generic compression of off-the-air radio frequency signals with grouped-bin FFT quantisation," in *28th European Signal Processing Conference (EUSIPCO)*, 2020, pp. 1767–1771.

[19] L. H. Nguyen and T. D. Tran, "RFI-radar signal separation via simultaneous low-rank and sparse recovery," in *2016 IEEE Radar Conference (RadarConf)*, 2016, pp. 1–5.

[20] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM Review*, vol. 35, no. 4, pp. 551–566, 1993. [Online]. Available: https://doi.org/10.1137/1035134

[21] A. Mitra, "Bit error analysis of new generation wireless transceivers," in *The 8th International Conference on Communication Systems*, vol. 2, 2002, pp. 636–640 vol.2.

[22] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977.

[23] H. Andrews and C. Patterson, "Singular value decomposition (SVD) image coding," *IEEE Transactions on Communications*, vol. 24, no. 4, pp. 425–432, 1976.

[24] K. Mishra, S. Kumar Singh, and P. Nagabhushan, "An improved SVD based image compression," in *2018 Conference on Information and Communication Technology (CICT)*, 2018, pp. 1–5.

[25] A. R., G. Xavier, H. V., N. Prasannan, R. Peter, and K. Soman, "GNU radio based control system," in *International Conference on Advances in Computing and Communications*, 2012, pp. 259–262.

[26] B. Hillburn. Datasets for radio deep learning. [Online]. Available: https://www.deepsig.io/new/2018/6/14/datasets-for-radio-deep-learning

[27] "The signal metadata format (SigMF), v0.0.2, jan. 8, 2019, https://sigmf.org."

[28] B. Hilburn, N. West, T. O'Shea, and T. Roy, "SigMF: The Signal Metadata Format," *Proceedings of the GNU Radio Conference*, vol. 3, no. 1, 2018. [Online]. Available: https://pubs.gnuradio.org/index.php/grcon/article/view/52

[29] Usrp hardware driver and usrp manual. Ettus Research. [Online]. Available: https://files.ettus.com/manual/page_configuration.html

[30] Invitation to the future. Yaesu. [Online]. Available: https://www.yaesu.com/pdf/System_Fusion_text.pdf

[31] M. Chiani and A. Elzanaty, "On the LoRa modulation for IoT: Waveform properties and spectral analysis," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8463–8470, Oct 2019.

[32] RF modules. HopeRF. [Online]. Available: https://www.hoperf.com/modules/lora/RFM98.html

[33] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4214–4217.

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[35] R. D. Badger and M. Kim, "Singular Value Decomposition for Compression of Large-Scale Radio Frequency Signals," in *29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1591–1595.

[36] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26–38, 1995.

[37] Z. Ji and K. R. Liu, “COGNITIVE RADIOS FOR DYNAMIC SPECTRUM ACCESS - Dynamic Spectrum Sharing: A Game Theoretical Overview,” *IEEE Communications Magazine*, vol. 45, no. 5, pp. 88–94, 2007.

[38] S. Bhattarai, J.-M. J. Park, B. Gao, K. Bian, and W. Lehr, “An Overview of Dynamic Spectrum Sharing: Ongoing Initiatives, Challenges, and a Roadmap for Future Research,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 2, no. 2, pp. 110–128, 2016.

[39] S. Chen, F. Xie, Y. Chen, H. Song, and H. Wen, “Identification of wireless transceiver devices using radio frequency (RF) fingerprinting based on STFT analysis to enhance authentication security,” in *IEEE 5th International Symposium on Electromagnetic Compatibility (EMC-Beijing)*, 2017, pp. 1–5.

[40] T. Jian, Y. Gong, Z. Zhan, R. Shi, N. Soltani, Z. Wang, J. G. Dy, K. R. Chowdhury, Y. Wang, and S. Ioannidis, “Radio frequency fingerprinting on the edge,” *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[41] 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.

[42] P. Pace, *Detecting and Classifying Low Probability of Intercept Radar, Second Edition.* Artech House, 2008.

[43] W. M. Lees, A. Wunderlich, P. J. Jeavons, P. D. Hale, and M. R. Souryal, “Deep Learning Classification of 3.5-GHz Band Spectrograms With Applications to Spectrum Sensing,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 224–236, 2019.

[44] US amateur radio service. FCC. [Online]. Available: https://www.fcc.gov/wireless/bureau-divisions/mobility-division/amateur-radio-service

[45] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[46] J. Cooley, P. Lewis, and P. Welch, "Historical notes on the fast Fourier transform," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 76–79, 1967.

[47] H. Zimmermann, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.

[48] G. Jajoo, Y. Kumar, A. Kumar, and S. Yadav, "Blind Signal Modulation Recognition through Density Spread of Constellation Signature," *Wireless Personal Communications*, vol. 114, 10 2020.

[49] S. Behura, S. Kedia, S. M. Hiremath, and S. K. Patra, "WiST ID—Deep Learning-Based Large Scale Wireless Standard Technology Identification," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1365–1377, 2020.

[50] H. Tamura, K. Yanagisawa, A. Shirane, and K. Okada, "Wireless Devices Identification with Light-Weight Convolutional Neural Network Operating on Quadrant IQ Transition Image," in *2020 18th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2020, pp. 106–109.

[51] Ettus N210 SDR. Ettus Research. [Online]. Available: https://files.ettus.com/manual/page_usrp2.html

[52] SBX daughterboard. Ettus. [Online]. Available: https://files.ettus.com/manual/page_dboards.html\protect\@normalcr\relax#dboards_sbx

[53] A. Paszke et. al, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.

[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).* Ieee, 2009, pp. 248–255.

[55] A. Luzzatto and M. Haridim, *Wireless transceiver design : Mastering the design of modern wireless equipment and systems.* John Wiley and Sons, Incorporated, 2016.

[56] J. Boccuzzi, *Signal processing for wireless communications.* McGraw-Hill Education, 2007.

[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[59] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[60] T. O'Shea, T. Roy, and T. C. Clancy, "Learning robust general radio signal detection using computer vision methods," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 829–832.

[61] T. J. O'Shea, T. Roy, and T. Erpek, "Spectral detection and localization of radio events with learned convolutional neural features," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 331–335.

[62] T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy. (2020) Deep Learning for Wireless Communications.

[63] C. Gravelle and R. Zhou, "SDR Demonstration of Signal Classification in Real-Time Using Deep Learning," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–5.

[64] T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, "Deep Learning for Wireless Communications," 2020.

[65] W. Gardner, *Cyclostationarity.* IEEE Press, 1994.

[66] J. L. Ziegler, R. T. Arn, and W. Chambers, "Modulation recognition with GNU radio, keras, and HackRF," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2017, pp. 1–3.

[67] 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 *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, 2007, pp. 212–215.

[68] 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.

[69] Y. Xia, Z. Ma, and Z. Huang, "Over-the-air radar emitter signal classification based on sdr," in *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2021, pp. 403–408.

[70] Ettus B210 SDR. Ettus Research. [Online]. Available: https://files.ettus.com/manual/page_usrp_b200.html

[71] A. M. Wyglinski, D. P. Orofino, M. N. Ettus, and T. W. Rondeau, "Revolutionizing software defined radio: case studies in hardware, software, and education," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 68–75, 2016.

[72] M. Patel, I. Darwazeh, and J. O'Reilly, "Bandpass sampling for software radio receivers, and the effect of oversampling on aperture jitter," in *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)*, vol. 4, 2002, pp. 1901–1905 vol.4.

[73] T. Lee and A. Hajimiri, "Oscillator phase noise: a tutorial," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 3, pp. 326–336, 2000.

[74] M. J. Underhill, "Time jitter and phase noise — Now and in the future?" in *IEEE International Frequency Control Symposium Proceedings*, 2012, pp. 1–8.

[75] R. P. Cutitta and C. R. Dietlein, "Phenomenology of signals degraded by phase noise," in *United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, 2018, pp. 1–2.

[76] C. Andrich, A. Ihlow, J. Bauer, N. Beuster, and G. Del Galdo, "High-precision measurement of sine and pulse reference signals using software-defined radio," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 5, pp. 1132–1141, 2018.

[77] S. Corum, J. D. Bonior, R. C. Qiu, N. Guo, and Z. Hu, "Evaluation of phase error in a software-defined radio network testbed," in *2012 Proceedings of IEEE Southeastcon*, 2012, pp. 1–4.

Curriculum Vitae

Robert David Badger

Researcher in the field of applied machine learning and signal processing for the Radio Frequency spectrum

Education:

- PhD Student in Intelligent Systems Engineering, June 2022, Luddy School of Informatics, Computing, and Engineering, Indiana University Bloomington, IN
- M.Sc. in Electrical and Computer Engineering, Aug. 2016, School of Engineering and Technology, Indiana University Purdue University Indianapolis
- BS in Electrical Engineering, Dec. 2003, University of Kentucky, Lexington, Kentucky

Positions Held:

- US Navy Electronic Warfare engineer, Aug. 2008 - present

Publications:

- Robert David Badger, Kristopher H. Jung, and Minje Kim, ”An Open-Sourced Time-Frequency Domain RF Classification Framework,” in Proceedings of the 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, Aug. 23-27, 2021, pp. 1701-1705
- Robert David Badger and Minje Kim, ”Singular Value Decomposition for Compression of Large-Scale Radio Frequency Signals,” in Proceedings of the 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, Aug. 23-27, 2021, pp. 1591-1595

Certifications:

- Engineer-In-Training (EIT) Dec 2003 EIT no. 12431

Skills:

- Deep Learning
- PyTorch
- Python
- Software Defined Radios
- Electronic Warfare (over 10 years of experience): USN lead CREW engineer for USMC Electronic Warfare Systems (EWS)