UNIVERSITY OF CALIFORNIA SAN DIEGO

Characterization of intended and unintended RF emissions

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Venkatesh Sathyanarayanan

Committee in charge:

Professor Peter Gerstoft, Chair
Professor Frederic Harris
Professor Florian Meyer
Professor Truong Nguyen

2024

The dissertation of Venkatesh Sathyanarayanan is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

To my guru aacharyaal Srimate Srivan Satakopa Sri Ranganatha Yatheendra Maha Desikan and family deity Oppilliappan,
To my country and temple that is Bhaaratham, the great saints who walked and blessed the land, and my forefathers,
To my parents for their love, patience, and the freedom they gave me,
To my wife's parents for their kindness and for treating me like their son,
To my wife for her love, patience, support, and being my best friend,
To my cousin Kanna for sowing the seeds of infatuation for research early in life

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I thank my advisor Prof. Peter Gerstoft for providing me with the wonderful opportunity of doing research in Noiselab that changed my life. I hope to emulate his kindness, cheerfulness, and work ethic in my academic career as a researcher and a teacher.

I thank my master's research advisor Prof. George Shoane for his encouragement through my Masters thesis and in pursuing a Ph.D. I thank Prof. Madhu Gupta for his guidance in the early stage of the transition from a corporate career to a Ph.D. I still remember his lectures on random processes and I wish I could emulate such a teaching environment in my career.

I thank my brother Bharath who has been kind and loyal as a brother, especially through the Ph.D. journey and I am indebted to him. I thank my brother-in-law Raj and his family for their kindness through my journey of Ph.D. I would especially like to thank shaanthaas Muntajibuddin and Santosh for their friendship and support. Their sense of joy and carefreeness in dealing with life has been a great practical learning experience for me, especially in the arduous Ph.D. journey.

I would like to thank all my wonderful colleagues at the UCSD Noiselab starting with Mark Wagner. He was very kind and supportive in the initial time of my PhD and helped me gain confidence. I would like to thank Mike Bianco, Byungjun, and Ankush who were wonderful collaborators in my research journey. Also, a special thanks to Aly-El Gamal for his guidance and kindness in my work on dataset generation for modulation classification. My sincere appreciation to my committee members Prof. Florian Meyer, Prof. Fred Harris, and Prof. Truong Nguyen.

This dissertation is a collection of papers that were published or have been prepared for publication. The text of Chapter 2, in full, is a reprint of the material as it appears in V. Sathyanarayanan, P. Gerstoft, "Anomalous activity detection using RF emanations", Submitted to IEEE Transactions on Electromagnetic Compatibility, 2024. The dissertation author was the primary researcher and author of this chapter. The co-authors listed in these publications directed and supervised the research.

The text of Chapter 3, in full, is a reprint of the material as it appears in V. Sathyanarayanan, P. Gerstoft, and A. E. Gamal, ”RML22: Realistic Dataset Generation for Wireless Modulation Classification,” in IEEE Transactions on Wireless Communications, vol. 22, no. 11, pp. 7663-7675, Nov. 2023. The dissertation author was the primary researcher and author of this chapter. The co-authors listed in these publications directed and supervised the research.

The text of Chapter 4, in full, is a reprint of the material as it appears in V. Sathyanarayanan, A. Jolly and P. Gerstoft, ”Novel Training Methodology to Enhance Deep Learning Based Modulation Classification,” 2021 55th Asilomar Conference on Signals, Systems, and Computers, pp. 356-360, 2021. The dissertation author was the primary researcher and author of this chapter. The co-authors listed in these publications directed and supervised the research.

The text of Chapter 5, in full, is a reprint of the material as it appears in V. Sathyanarayanan, M. Wagner and P. Gerstoft, ”Over The Air Performance of Deep Learning for Modulation Classification across Channel Conditions”, 2020 54th ASILOMAR Conference on Signals, Systems, and Computers, 2020, pp. 157-161. The dissertation author was the primary researcher and author of this chapter. The co-authors listed in these publications directed and supervised the research.

VITA

2008	B. Tech. in Instrumentation and Control Engineering, NIT Tiruchirappalli (India)
2011	M. S. in Electrical and Computer Engineering, Rutgers, the State University of New Jersey
2012-2016	Systems Engineer at Qualcomm Location Technologies, San Diego
2016-2019	Systems Engineer at Qualcomm Corporate Research, San Diego
2024	Ph. D. in Electrical Engineering (Signal and Image Processing), University of California San Diego

PUBLICATIONS

V. Sathyanarayanan, P. Gerstoft, ”Anomalous activity detection using RF emanations”, Submitted to IEEE Transactions on Electromagnetic Compatibility. (Under Review)

V. Sathyanarayanan, P. Gerstoft, ”Detection and characterization of unintended RF emissions on wideband real data” Submitted to International Conference on Signal Processing and Communications (SPCOM). (Under Review)

V. Sathyanarayanan, P. Gerstoft and A. E. Gamal, ”RML22: Realistic Dataset Generation for Wireless Modulation Classification,” in IEEE Transactions on Wireless Communications, vol. 22, no. 11, pp. 7663-7675, Nov. 2023.

V. Sathyanarayanan, P. Gerstoft and A. El Gamal, ”Data Centric Approach to Modulation Classification,” 2023 15th International Conference on COMmunication Systems and NETworkS (COMSNETS), pp. 340-344, 2023.

V. Sathyanarayanan, M. Wagner and P. Gerstoft, ”Over The Air Performance of Deep Learning for Modulation Classification across Channel Conditions,” 2020 54th Asilomar Conference on Signals, Systems, and Computers, pp. 157-161, 2020.

V. Sathyanarayanan, A. Jolly and P. Gerstoft, ”Novel Training Methodology to Enhance Deep Learning Based Modulation Classification,” 2021 55th Asilomar Conference on Signals, Systems, and Computers, pp. 356-360, 2021.

B. Kim, **V. Sathyanarayanan**, C. Mecklenbräuker and P. Gerstoft, ”Deep Learning-Based Modulation Classification for OFDM Systems Without Symbol-Level Synchronization,” IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), pp. 1-5, June 2023.

M. Wagner, H. Groll, A. Dormiani, **V. Sathyanarayanan**, C. Mecklenbräuker and P. Gerstoft, ”Phase Coherent EM Array Measurements in a Refractive Environment,” in IEEE Transactions on Antennas and Propagation, vol. 69, no. 10, pp. 6783-6796, Oct. 2021.

ABSTRACT OF THE DISSERTATION

Characterization of intended and unintended RF emissions

by

Venkatesh Sathyanarayanan

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California San Diego, 2024

Professor Peter Gerstoft, Chair

Spectrum sensing is essential for enabling optimal spectrum usage and ensuring data security, especially with the proliferation of IoT devices. Spectrum sensing involves detecting and characterizing intentional RF emissions called overt, and unintentional RF emissions called emanations. The thesis focuses on two pivotal aspects of spectrum sensing: characterization of overt specifically modulation classification and characterization of emanations.

Three distinct works are presented on modulation classification. DL has been successfully used recently. The focus, however, has been model-centric, with attempts to improve performance on the standard synthetic dataset RML16. The quality of the training dataset impacts model performance on real data. A hybrid approach is taken by leveraging wireless domain knowledge to improve dataset quality. The first two works respectively leverage domain knowledge of wireless channel conditions and SNR to improve dataset quality. Over-the-air (OTA) data captured using USRP radios are used in these works.

In the first work, studies are done to understand the performance impact due to the disparity of probability distribution between training and test data within the context of channel conditions. This is studied for OTA data collected in channels emulating LOS, NLOS, and AWGN. In the second work, signal processing advances in blind SNR estimation are leveraged to improve DL performance on modulation classification. A training methodology is introduced that partitions OTA data into subsets of different SNR levels. For the third work, shortcomings such as errors and ad-hoc choice of parameters are identified in RML16. A new realistic benchmark dataset RML22 is provided with the errors corrected and the choice of parameters justified. Thorough mathematical derivations are provided for the wireless models used to generate data. Performance impact due to artifacts and model parameterization is studied using the RML22 data generation framework.

For the second paradigm of detection and characterization of emanations, an HW agnostic solution is proposed. Prior work focussed on profiling specific HW but scalability led to the need for a HW-agnostic solution. Emanations are detected by scanning for the signature of harmonics from leakages of clock signals. A signal processing algorithm is provided to remove artifacts and estimate the pitch of harmonics that characterizes the emanation. IQ data is collected from the source of emanations placed inside a sanitized shield room using Signal Hound SDR. Results for anomaly detection using emanation patterns are presented for the use cases compromising data security: damaged electronic peripherals, and illegal copying of data to external storage devices.

Chapter 1

Introduction

Spectrum sensing refers to the detection and characterization of all RF signals. Detection refers to the ability to sense that there is an RF signal that is not noise. Characterization refers to finding metadata about the detected signal such as protocol, and modulation type. This is essential in enabling optimal spectral usage and detecting anomalous signals. Spectrum is a critical resource due to the increasing number of devices and data rate requirements. Developing intelligence in devices enables optimal spectral usage and thereby meets increasing spectral demands. It is also of interest for cellular carriers and defense organizations to police the spectrum. This helps detect illegal and malicious spectrum usage thereby ensuring robust communication links for critical applications.

Spectrum sensing typically refers to the detection of intentionally transmitted signals, called overt signals. Overt signals include cellular signals 4G, 5G, Wi-Fi, Bluetooth, GNSS, etc., that are transmitted for applications such as communication, and sensing. Another class of signals is unintentionally emitted RF signals, called emanations. Activity within electronic systems results in emanations [1]. Prior work [2, 3, 4, 5, 6, 7, 8] is focused on decoding information stealthily from emanations. This is done by mapping emanations to the data processed within the electronic system. Emanations are a risk for data security especially in sensitive military and civilian establishments. The IARPA developed the SCISRS program [9] to detect and characterize emanations.

The thesis focuses on two important cogs of spectrum sensing: modulation classification, detection, and characterization of emanations.

1.1 Dissertation overview

Chapter 2 presents the work on the detection and characterization of emanations. Sensing and understanding all signals in an RF-secure military or civilian setup is important, this includes unintended RF emissions called emanations. Prior work provide citations for prior work in detecting emanations involves profiling, which is hardware (HW) specific and hence not a scalable approach. Our technique looks for a generic signature of harmonics in the frequency domain without knowledge of HW. It detects emanation and characterizes it by estimating the pitch frequency of harmonic. Results are shown for detection of anomalous activity using emanation patterns for use cases emulating wear and tear of HW, and illegal data transfer to external storage devices.

Modulation classification has been an active area of research for the last few decades. Traditional approaches to modulation classification are broadly classified as likelihood-based and feature-based [10]. They typically work well for only a small subset of modulation types. In the last few years, researchers have borrowed tools from DL and applied them to the problem of modulation classification, demonstrating tremendous potential for universal success across a wide range of modulation types and wireless technologies. The focus, however, has been model-centric. Numerous architectures [11, 12, 13, 14, 15, 16] such as CNN, recurrent neural networks, generative adversarial networks and auto-encoders have been attempted on benchmark datasets RADIOML.2016.10A (RML16) [11] and RADIOML.2018.01A (RML18) [13]. Wireless communications and signal processing are mature fields. In this thesis, a hybrid approach is taken to leverage the efforts from these fields to improve the performance of deep learning on modulation classification.

Three distinct works are presented under modulation classification covered in Chapters 3, 4 and 5. In Chapter 3 provide citation for both journal and conference, we use a data-centric DL approach where the focus is on improving training dataset quality. Wireless systems knowledge is used to generate synthetic datasets such as RML16. However, DL model performance is only as good as dataset quality. RML16 has shortcomings such as errors and ad-hoc choices of parameters. We build upon RML16 and provide a realistic and correct methodology for generating datasets. A new benchmark dataset RML22 is generated. Going forward, we envision researchers improving model quality on RML22.

The work presented in Chapter 4 provide citation leverages work done in blind SNR estimation provide citation to improve modulation classification performance. A novel training methodology is introduced where different models are trained on signals belonging to specific subsets of SNRs. At inference, signals with specific SNRs are passed onto appropriate models showing performance improvement.

The work presented in Chapter 5 provide citation empirically studies the performance impact due to the disparity in probability distributions between training and test data. Software-defined radios (SDR) collect training and test data under channel conditions of additive white Gaussian noise (AWGN), line-of-sight (LOS), and non-line-of-sight (NLOS). Models are trained and tested on data collected in the three different channel conditions of AWGN, LOS, and NLOS. Results show that train and test data should belong to the same channel conditions for best performance.

1.2 Basics overview

1.2.1 Wireless signal representation

Wireless signals are transmitted on orthogonal carrier waves. The magnitudes of the orthogonal carrier waves are referred to as In phase and Quadrature phase (IQ) in baseband. IQ samples are the basic signal representation of a wireless signal. A single IQ sample can be represented as a two-dimensional real signal or a single complex number. An IQ sample s is

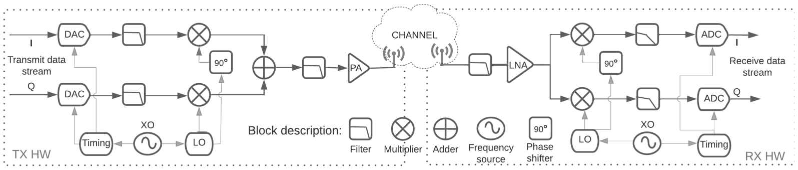


Figure 1.1. A homodyne architecture of the TX and RX HW. The HW model used follows the architecture presented.

represented as real numbers s_{I} and s_{Q}.

$$s[n] = s_{\text{I}}[n] + s_{\text{Q}}[n], \tag{1.1a}$$

$$s_{\text{p}}[n] = \Re(s[n]\mathrm{e}^{j2\pi f_c n T_{\text{s}}}), \tag{1.1b}$$

where s is baseband signal, s_p passband signal with center frequency f_c, s_{I} and s_{Q} are in phase and quadrature phase parts of an IQ sample, T_s is the sampling interval.

An IQ sample is a digital representation of a baseband signal. Baseband signal is typically shifted to a higher frequency before transmission by TX HW and is then referred to as a passband signal.

1.2.2 Transmit and receive hardware architecture

A TX and RX HW based on a homodyne architecture [17, pg. 337] is illustrated in Fig. 1.1. The HW artifacts models used in the thesis follow this architecture. The digital IQ stream is converted into an analog signal by the digital-to-analog converter (DAC). Further, it is shifted from baseband to a carrier frequency, boosted by a power amplifier and transmitted OTA via antennae. This signal undergoes channel effects also referred to as radio propagation effects, before reaching the RX. At RX, the antenna senses the electromagnetic energy, converts into an electric voltage. This electrical signal is amplified by a low-noise amplifier, shifted to baseband from a higher center frequency. It is then converted into a digital IQ stream by the analog-to-digital converter (ADC). A clock crystal is an important part of wireless system. It provides tone at requisite frequency via a phase-locked-loop (PLL) for the frequency shifting operations in the TX and RX systems. It also provides reference frequencies for generating sample timing to the ADC and DAC components.

1.2.3 Model of Channel and Hardware artifacts

The input-output model is presented where an input signal s is impacted by the transfer function constituted by channel and HW artifacts. The resulting signal is sensed by a spectrum-sensing HW as receive signal y. This model is used in Chapter 2 for the mathematical derivations demonstrating the emanation selection algorithm. This is also used in Chapter 3 for generating the RML22 dataset for modulation classification. The received IQ signal y for a wireless signal affected by channel and HW effects is [18, pg. 16]:

$$y[n] = e^{j2\pi f_{\text{err}} n T_s + \theta_{\text{err}}} \sum_{l} h[l] s[n - l - \zeta_{\text{err}}] + z[n], \tag{1.2}$$

where h is the channel impulse response, s is the modulated symbol that is up-sampled and pulse shaped by a RRC filter, T_s sampling rate, f_{err} frequency error, θ_{err} phase error, ζ_{err} timing error, z Additive White Gaussian Noise (AWGN). Note that y, s, h are the digital baseband equivalent terms.

The random variables $\zeta_{\text{err}}, \theta_{\text{err}}, f_{\text{err}}, z, h$ represent the set of artifacts that is imposed upon the clean transmit IQ stream.

Thermal Noise model

The thermal noise is modeled as follows in RML22.

$$SNR = 10\log_{10}\left[E(|s|^2)/E(|z|^2)\right], \tag{1.3a}$$

$$y[n] = s[n] + z[n], \tag{1.3b}$$

$$z[n] = z_I[n] + jz_Q[n] \sim \mathscr{CN}(0, \sigma_{\mathrm{z}}^2), \forall n, \tag{1.3c}$$

$$z_I[n], z_Q[n] \sim \mathscr{N}(0, \sigma_{\mathrm{z}}^2/2), \forall n, \tag{1.3d}$$

$$E(z(t)z^*(t+\tau)) = \delta(\tau)\sigma_{\mathrm{z}}^2,\ E(z_I z_Q) = E(z_I)E(z_Q) \tag{1.3e}$$

where $z[n]$ is the nth sample of complex thermal noise, z_I and z_Q are the real and imaginary parts, σ_{z} is the standard deviation of thermal noise.

Thermal noise z is modeled as zero mean AWGN, see (1.3c) and (1.3e), with the real and imaginary parts independently zero-mean AWGN random process[19, p 29], see (1.3d). SNR is calculated using (1.3a). The dominant source of thermal noise is assumed to be at the receiver, thus the additive assumption. In simulations, thermal noise z is assumed ergodic, and expectation calculated averaging samples over time. Although thermal noise simulated in baseband is a filtered and sampled version, the properties mentioned above are assumed to hold. The GNU radio block simulating thermal noise is in [20].

Phase, Frequency and Timing error

A model for simulating clock effects is presented, following the homodyne architecture in Fig. 1.1. The errors in the clock in TX and RX HW manifest as CFO, SRO and phase offset.

The cumulative CFO effects are:

$$f_{\mathrm{LO}}^{TX}[n] = (\hat{f}_{xo} + \Delta f_{xo}^{TX}[n])L_{LO}, \tag{1.4a}$$

$$f_{\mathrm{LO}}^{RX}[n] = (\hat{f}_{xo} + \Delta f_{xo}^{RX}[n])L_{LO}, \tag{1.4b}$$

$$f_{\mathrm{err}}^{\mathrm{XO}}[n] = \Delta f_{xo}^{TX}[n] - \Delta f_{xo}^{RX}[n], \tag{1.4c}$$

$$f_{\mathrm{err}}^{\mathrm{LO}}[n] = f_{\mathrm{LO}}^{TX}[n] - f_{\mathrm{LO}}^{RX}[n] = f_{\mathrm{err}}^{\mathrm{XO}}[n]L_{LO}, \tag{1.4d}$$

$$s'[n] = s[n]e^{j2\pi f_{\mathrm{err}}^{\mathrm{LO}} nT_S}, \tag{1.4e}$$

where $\Delta f_{xo}^{TX}[n], \Delta f_{xo}^{RX}[n]$ are crystal oscillator (XO) errors in TX and RX, $\hat{f}_{xo}$ is the needed reference tone frequency from XO, L_{LO} is the scaling factor to shift tone from XO frequency to center frequency, f_{LO}^{TX} and f_{LO}^{RX} are the LO signals in TX and RX, $f_{\mathrm{err}}^{\mathrm{XO}}$ and $f_{\mathrm{err}}^{\mathrm{LO}}$ are the XO and LO frequency errors in combined TX-RX system, s is the clean baseband signal, s' is the baseband signal with CFO.

The XO crystal of an RF system is prone to errors that manifest as CFO and SRO. Frequency source from XO feed the LO via a phase locked loop (PLL). This in turn is used for providing center frequencies for up-conversion and down-conversion in TX and RX systems, respectively. XO frequency errors cause a mismatch in the LO frequencies of TX and RX that manifests as CFO f_{err}, see (1.4) [21, pg. 360]. XO crystal is also used to generate accurate time ticks for the digital-to-analog-converter and analog-to-digital-converter via a timing PLL. XO frequency errors cause a mismatch in the sampling instants of DAC and ADC in TX and RX systems thereby causing SRO ζ_{err}, see (1.6) [21, pg. 436]. XO is assumed the only common source of errors for both CFO and SRO. In practice, the PLL leading into LO, DAC, ADC etc., also contribute to minor errors.

$$\hat{t} = 1/(\hat{f}_{\text{XO}} L_t) = 1/f_{\text{CR}}, \tag{1.5a}$$

$$t_{\text{DAC}} = \frac{1}{(\hat{f}_{\text{XO}} + \Delta f_{\text{XO}}^{\text{TX}}[n])L_t}, \tag{1.5b}$$

$$t_{\text{ADC}} = \frac{1}{(\hat{f}_{\text{XO}} + \Delta f_{\text{XO}}^{\text{RX}}[n])L_t}, \tag{1.5c}$$

$$\zeta_{\text{TX}}[n] = t_{\text{DAC}} - \hat{t} \approx \Delta f_{\text{XO}}^{\text{TX}}[n] L_t / f_{\text{CR}}^2, \tag{1.6a}$$

$$\zeta_{\text{RX}}[n] = t_{\text{ADC}} - \hat{t} \approx \Delta f_{\text{XO}}^{\text{RX}}[n] L_t / f_{\text{CR}}^2, \tag{1.6b}$$

$$\zeta_{err}[n] = \sum_{i=1}^{n} (\zeta_{\text{TX}}[i] - \zeta_{\text{RX}}[i]) = \sum_{i=1}^{n} \frac{f_{\text{err}}^{\text{XO}}[i] L_t}{f_{\text{CR}}^2}, \tag{1.6c}$$

where $\hat{t}$ is the requisite time tick interval to DAC and ADC, f_{CR} is the clock rate for DAC and ADC (equal to the analog bandwidth), L_t is the scaling factor in timing PLL, t_{DAC} and t_{ADC} are real time tick intervals going to DAC and ADC, ζ_{TX} and ζ_{RX} are the SRO in TX and RX, ζ_{err} is the cumulative SRO in combined TX-RX system. The scaling factor L_t in timing PLL is assumed the same for both DAC and ADC.

Phase changes occur due to factors such as frequency errors, Doppler, sampling errors, distance traveled and non-synchronous TX and RX LO. Artifacts as f_{err}, ζ_{err}, $h(.)$ capture these phase changes, except non-synchronous TX and RX LO. This is captured by θ_{err}:

$$x_{\text{LO}}^{\text{TX}}[n] = e^{j2\pi f_{\text{LO}}^{\text{TX}} nT_s + \theta_{err}^{\text{TX}}}, \tag{1.7a}$$

$$x_{\text{LO}}^{\text{RX}}[n] = e^{j2\pi f_{\text{LO}}^{\text{RX}} nT_s + \theta_{\text{err}}^{\text{RX}}}, \tag{1.7b}$$

$$x_{\text{LO}}^{\text{TX}}(x_{\text{LO}}^{\text{RX}})^* = e^{j2\pi f_{\text{err}}^{\text{LO}} nT_s + (\theta_{\text{err}}^{\text{TX}} - \theta_{\text{err}}^{\text{RX}})}, \tag{1.7c}$$

where $x_{\text{LO}}^{\text{TX}}$ and $x_{\text{LO}}^{\text{RX}}$ are the TX and RX LO signals, $\theta_{\text{err}}^{\text{TX}}$ and $\theta_{\text{err}}^{\text{RX}}$ are the phase errors in the TX and

RX respectively, θ_{err} is the net phase error from the combined TX and RX systems. The TX and RX LO signals used to upconvert and downconvert signals in TX and RX respectively. The net effect of TX and RX LO after downconversion at RX is in (1.7c).

The methodology to simulate f_{err}, ζ_{err} and θ_{err} is in (1.8). CFO is a clipped Gaussian process (1.8b). SRO is simulated via re-sampling through interpolation at time instants specified by t' (1.8e). Phase error θ_{err} is from a uniform distribution (1.8f).

$$f_{\text{bias}} \sim U(-f_{\text{max}}, f_{\text{max}}), \tag{1.8a}$$

$$f^{\text{XO}}_{\text{err}}[n] \sim N(f_{\text{bias}}, n\sigma^2), \quad |f^{\text{XO}}_{\text{err}}[n]| \leq f_{\text{max}} \tag{1.8b}$$

$$s'[n] = s[n]e^{j2\pi f^{\text{XO}}_{\text{err}}[n]L_{\text{LO}}nT_s} \tag{1.8c}$$

$$T_s^{\text{DAC}}[n] = n\hat{t}, \;\; T_s^{\text{ADC}}[n] = n\hat{t} + \zeta_{err}[n], \tag{1.8d}$$

$$t'[n] = \frac{f_{\text{CR}}}{f_s} T_s^{\text{ADC}}[n] = \frac{n}{f_s} + \frac{1}{f_s f_{\text{CR}}} \sum_{i=1}^{n} f^{\text{XO}}_{\text{err}}[i]L_t, \tag{1.8e}$$

$$\theta_{\text{err}} \sim U(0, 2\pi), \quad s'[n] = s[n]e^{j2\pi\theta_{\text{err}}}, \tag{1.8f}$$

where f_{bias} is the CFO at the start of frame, σ standard deviation per sample, f_{max} maximum frequency error bound, T_s^{DAC} and T_s^{ADC} sampling instants, t' new sampling instant.

Channel model

Channel effects are classified under small-scale and large-scale fading. Large-scale fading effect is due to path losses as a function of distance, shadowing effects, etc., and is considered stationary in time scales of a frame duration. It thus manifests as reduction in signal power at receiver, as captured by thermal noise model in Sec. 1.2.3. Small-scale fading occurs in the distance scale of carrier wavelengths, that cause rapid changes in phase and amplitude of received signal. The channel model considered here represents small-scale fading.

A wireless signal encounters numerous objects from TX to RX. The resultant signal at the RX is the sum of multiple reflected, scattered signal copies with delay. The magnitude of each multipath depends on path loss and material properties of reflector and scatters. The delay depends on path length of the multipath. Modeling the effect along each path via ray tracing needs complete knowledge of the channel, and not feasible. Instead, modeling via an input-output relation is used [19, pg. 26], where y is the output signal, h channel impulse response, x input signal. The passband is represented as:

$$\begin{aligned} y_{\text{p}}(t) &= \int h_{\text{p}}(\tau,t)x_{\text{p}}(t-\tau)\text{d}\tau, \\ &= \sum_i a_{\text{p}}(i,t)x_{\text{p}}(t-\tau(i,t)), && \text{(1.9a)} \\ h_{\text{p}}(\lambda,t) &= \sum_i a_{\text{p}}(i,t)\delta(\lambda-\tau(i,t)), && \text{(1.9b)} \end{aligned}$$

and baseband represented as:

$$\begin{aligned} y[m] &= \sum_l h[l,m]x[m-l] && \text{(1.10a)} \\ &= \sum_l x[m-l]\sum_i a[i,m]\text{sinc}[l-\frac{\tau[i,m]}{T_s}], && \text{(1.10b)} \\ h[l,m] &= \sum_i a_{\text{p}}[i,m]e^{-j2\pi f_c\tau[i,m]}\text{sinc}[l-\frac{\tau[i,m]}{T_s}], && \text{(1.10c)} \end{aligned}$$

where $a_{\text{p}}(i,t)$ is the passband magnitude response of ith path at time t, $a[i,m]$ baseband magnitude response of ith path at time mT_s, $\tau(i,t)$ is the delay of ith path at time t, $\delta(.)$ is an impulse signal.

The channel $h_p(\lambda,t)$ is represented as a slow, time varying system with memory. Most significant reflected and scattered paths are only considered and represented in the equation with an index i. The channel is assumed to be underspread where the timescale of channel variations is significantly longer than the delay spread of the channel. The continuous time passband representation of the input output relation and channel impulse response is presented in (1.9). The equivalent discrete baseband representation is in (1.10).

The radio propagation channel could be an office space, city downtown etc. To capture the propagation effects, detailed knowledge of the objects in the signal path and large computational resources are needed. An alternate approach is using statistical parametric model representing the channel.

$$f_X(x;\sigma_l) = \frac{2x}{\sigma_l^2}\exp\left\{\frac{-x^2}{\sigma_l^2}\right\}, \tag{1.11a}$$

$$f_X(x;K,\sigma_l) = \frac{x}{\sigma_l^2}\exp\left\{\frac{-(x^2+K^2)}{\sigma_l^2}\right\} I_0\left[\frac{Kx}{\lambda^2}\right], \tag{1.11b}$$

$$R[l,n] = E_m\left\{h^*[l,m]h[l,m+n]\right\}, \tag{1.11c}$$

$$S[v] = \sum_l \sum_n R[l,n]e^{-j2\pi vn}, \tag{1.11d}$$

$$S_{\text{Jakes}}[v] = \frac{1}{\pi f_d\sqrt{1-(v/f_d)^2}}, \; |v| \le f_d = \frac{\mathrm{v}_{\max}}{\lambda}, \tag{1.11e}$$

where f is a probability distribution, x and σ_l are the magnitude and standard deviation of the lth tap or path depending upon whether the channel taps $h[l,m]$ or path magnitudes $a[l,m]$ are modeled, K referred as K-factor is ratio of energies in direct and reflected paths, I_0 is the zeroth order modified Bessel function of the first kind, R is the auto-correlation function of channel impulse response, S is the Doppler spectrum, v Doppler frequency, f_d maximum Doppler frequency, $\mathrm{v}_{\max}$ maximum relative speed between TX and RX, λ carrier frequency wavelength.

In the statistical model, the magnitude of each path is assumed an aggregate of numerous paths of similar delay. By the central limit theorem, the real and imaginary components of the magnitudes $a(l,t)$ are zero-mean Gaussian. The amplitude of each path is thus a Rayleigh distribution, see (1.11a). An alternate model used is the Rician distribution, see (1.11b). Due to the independence between paths, the phase is uniformly [0, 2π] distributed. The channel tap magnitudes are equivalently modeled as a Rayleigh or Rician distribution.

The time varying nature of the channel is modeled via a tap gain auto-correlation function, see (1.11c). A measure of variance of channel in time for each tap l is conveniently captured by auto-correlation function, upon using the wide-sense stationarity assumption. The tap gain auto-correlation function is averaged across channel taps l and a Fourier transform in time is taken to obtain the Doppler spectrum, see (1.11d). The Doppler spectrum indicates the amount of spread in frequency due to time variation of the channel. Jakes model (1.11e) is commonly used to simulate the Doppler spread due to time varying nature of the channel.

1.3 References

[1] B. Yilmaz, E. Ugurlu, and M. Prvulovic, "Detecting Cellphone Camera Status at Distance by Exploiting Electromagnetic Emanations," in *IEEE Mil. Commun. Conf.*, pp. 1–6, 2019.

[2] M. Dey, A. Nazari, and A. Zajic, "EMPROF: Memory Profiling Via EM-Emanation in IoT and Hand-Held Devices," in *IEEE Int. Symp. Microarchit.*, pp. 881–893, 2018.

[3] A. Nazari, N. Sehatbakhsh, and M. Alam, "EDDIE: EM-based detection of deviations in program execution," in *IEEE Int. Symp. Comput. Archit.*, pp. 333–346, 2017.

[4] M. Bari, M. Chowdhury, and B. Chatterjee, "Detection of Rogue Devices using Unintended Near and Far-field Emanations with Spectral and Temporal Signatures," in *IEEE Int. Microw. Symp.*, pp. 591–594, 2022.

[5] M. Bari, M. Chowdhury, and S. Sen, "Is broken cable breaking your security?," in *IEEE Int. Symp. Circuits and Syst.*, pp. 1–5, 2023.

[6] J. Feng, T. Zhao, and S. Sarkar, "Fingerprinting IoT Devices Using Latent Physical Side-Channels," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, no. 2, pp. 1–26, 2023.

[7] P. Socha, V. Miškovský, and M. Novotný, "A Comprehensive Survey on the Non-Invasive Passive Side-Channel Analysis," *Sensors*, vol. 22, pp. 1–37, 2022.

[8] O. Sosa, Z. Dyka, and I. Kabin, "Simulation of Electromagnetic Emanation of Cryptographic ICs: Tools, Methods, Problems," in *IEEE East West Des. Test Symp.*, pp. 1–5, 2021.

[9] P. Kolb, "Securing compartmented information with smart radio systems."

[10] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Commun.*, vol. 1, pp. 137–156, 2007.

[11] T. J. O'Shea, J. Corgan, T. C. Clancy, C. Jayne, and L. Iliadis, "Convolutional radio modulation recognition networks," *Int. Conf. Eng. Appl. Neural Netw.*, pp. 213–226, 2016.

[12] X.Liu, D. Yang, and A. El-Gamal, "Deep neural network architectures for modulation classification," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, pp. 915–919, 2017.

[13] T. O'Shea, T.Roy, and T. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, 2018.

[14] M. Patel, X. Wang, and S. Mao, "Data Augmentation with Conditional GAN for Automatic Modulation Classification," in *ACM Workshop Wireless Secur. Mach. Learn.*, pp. 31–36, 2020.

[15] A. Ali, F. Yangyu, and S. Liu, "Automatic modulation classification of digital modulation signals with stacked autoencoders," *Digital Signal Process.*, vol. 71, pp. 108–116, 2017.

[16] R. Zhou, F. Liu, and C. Gravelle, “Deep Learning for Modulation Recognition: A Survey With a Demonstration,” *IEEE Access*, vol. 8, pp. 67366–67376, 2020.

[17] D. Pozar., *Microwave and Rf Design of Wireless Systems*. Wiley Publishing, 2000.

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

[19] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[20] M. Mueller, “GNU Radio thermal noise block.”

[21] M. Rice, *Digital Communications: A Discrete-Time Approach*. Pearson Education Inc, 2012.

Chapter 2

Anomalous Activity Detection using RF Emanations

Data security is important in corporate and military establishments. It is important to find anomalous activities that compromise data security. This is done by studying patterns in unintended radio frequency (RF) emissions called emanations. Prior work on emanation detection uses profiling on specific hardware (HW), however, this is not scalable across all types of HW. We propose a HW-agnostic solution for finding anomalous activity using emanations. Emanations are detected by scanning the signature of harmonics from leakages of clock signals. Harmonics undergo random modulation, channel and HW artifacts, and thermal noise. A signal processing algorithm is proposed to detect and characterize emanations, by estimating the pitch of harmonics. A preprocessing technique removes the effect of modulation and artifacts. Thorough mathematical derivations demonstrate the algorithm theoretically. In-phase and Quadrature-phase (IQ) data is collected from emanation sources placed in a shielded room from 0.1–1.1 GHz using a software-defined radio (SDR). Results are presented for use cases emulating anomalous activity such as a damaged mouse and keyboard, data transfer between a laptop, and external data storage peripherals.

2.1 Introduction

Digital data safety involves the protection of digital data stored and processed by electronic systems, against unauthorized access. Increasing digitization of daily life activities has increased the volume of data and thus complicated the life cycle of data. This complexity has increased the challenges of data security. Activities that compromise data security are termed anomalous. The focus of this work is detecting anomalous activities using unintended RF emissions called emanations.

Activity within electronic systems results in electromagnetic radiation [1]. If a mapping is found between the radiations and data processed within the electronic system, it makes digital data vulnerable to side-channel attacks. An unintentional medium of data leakage existing in electronic devices is called a side channel [2]. The technology related to these vulnerabilities, attacks, and protection against attacks is popularly called TEMPEST [3]. Other types of emanation signals include power consumption pattern [4], acoustic signals [5], and optical signals [6].

In any RF environment, ambient emanations from numerous sources form the baseline of the emanation pattern. It is essential to detect and characterize the emanation patterns across a wide bandwidth to learn this baseline. A change in the baseline is identified as a new emanation pattern symptomatic of anomalous activity.

Prior work [7, 8, 9, 10, 11, 12, 13] is focused on decoding information from side-channels. They profile a specific make and model of HW to learn emanation patterns. They assume knowledge of the specific device under question. The profiling uses traditional and deep learning approaches to map specific types of software activities to received emanations. This knowledge gained through profiling for a specific device is used to detect emanations and decode information surreptitiously. The profiling approach for anomalous activity detection requires learning patterns from all types of HW make and models, which is not feasible.

We propose to scan for a generic signature symptomatic of emanations. Intentionally

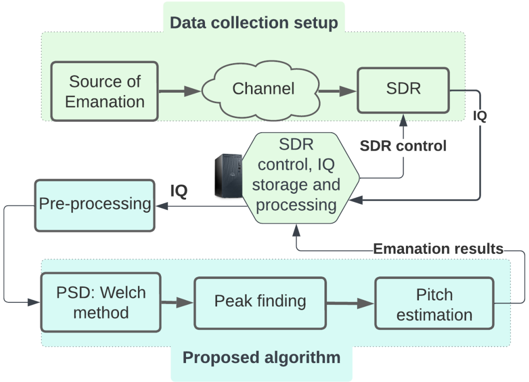


Figure 2.1. System block diagram illustrating the high-level modules of data collection setup and proposed algorithm.

transmitted signals such as Wi-Fi, Bluetooth, 4G, and 5G cellular signals, are human-constructed, with a regular structure. Spectrum sensing applications exploit this structure to detect and classify signals with known templates [14, 15]. Motivated by this, a profiling-free approach is approached by scanning for a template of harmonics due to leakages from clock signals, they are an essential component of electronic systems [16, 17, 18]. The generic HW-agnostic signature proposed is a harmonic series in the frequency domain at pitch frequencies spread across wide bandwidths, that are affected by artifacts.

The system block diagram in Fig. 2.1 summarizes the emanation detection algorithm and data collection. Emanations are transmitted from the source, where harmonic series at different pitch and center frequencies are obfuscated by modulation. They further undergo channel and HW artifacts, and thermal noise before being sensed by SDR and converted into IQ samples. Since this is the first work to showcase profiling-free emanation detection, the objective is to showcase performance on data collected in a controlled environment such as a shielded room for a limited set of use cases, backing the theoretical exposition.

An important preprocessing is developed to deal with the random frequency shifts at the source of emanation, channel, and HW artifacts. Derivations are presented describing the removal of artifacts, harmonic structure extraction, finding peaks, and estimating the pitch of harmonics. This is shown for single-harmonic, and multi-harmonic cases in the presence of interferences.

Detection of anomalous activity is shown for the following use cases. Due to wear and tear of electronic equipment such as a mouse or keyboard, there could be leakages compromising data security [18]. This is emulated by exposing a portion of the cable from a mouse and keyboard connected to a laptop and collecting IQ data from resulting emanations. The 0.1–1.1 GHz data is split into 25 MHz slices processed separately. The resulting emanation pattern consists of emanations detected in each frequency slice across. Emanation patterns for IQ from damaged peripherals are compared with the baseline of an idle laptop only, to detect anomalous activity.

Data security is compromised by copying data without permission onto external storage peripherals, these data transfer results in emanations [19]. An organization could use the proposed algorithm to detect these anomalous activities.

The major contributions of this paper are as follows:

- Profiling-free emanation detection is provided by scanning for a harmonic signature. Prior work did profiling of specific HW make and model that is not a feasible solution.

- Thorough mathematical derivations are provided showcasing the theoretical performance of the proposed algorithm - removal of artifacts using the preprocessing technique, finding peaks, and estimating pitch frequency.

- A preprocessing technique is introduced that removes modulation, channel, and HW artifacts, and retains harmonic structure.

- Anomalous activity detection is showcased on wideband IQ data collected in a shielded room from 0.1–1.1 GHz.

2.2 Method

The high-level details of the method are in the flow chart in Fig. 2.2. Models are presented for signal, modulation of signal, and artifacts it undergoes in Sec. 2.2.2, 2.2.3 and 2.2.4. The

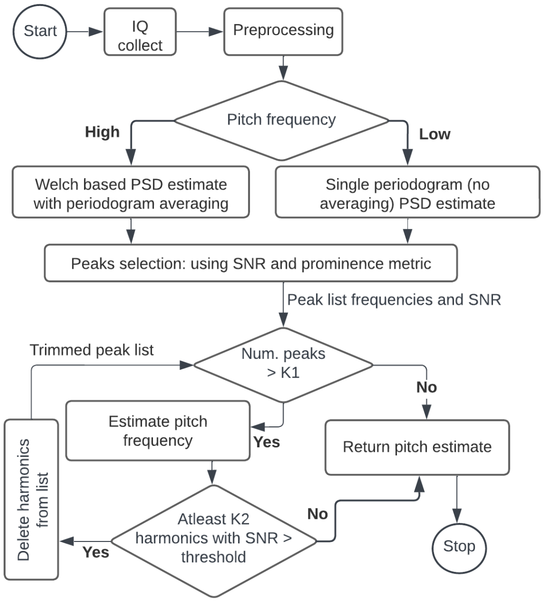

Start
IQ collect
Preprocessing
Pitch frequency
High
Low
Welch based PSD estimate with periodogram averaging
Single periodogram (no averaging) PSD estimate
Peaks selection: using SNR and prominence metric
Peak list frequencies and SNR
Trimmed peak list
Num. peaks > K1
No
Estimate pitch frequency
Yes
Return pitch estimate
Delete harmonics from list
Atleast K2 harmonics with SNR > threshold
No
Yes
Stop

Figure 2.2. Emanation detection flow chart of the proposed algorithm detecting emanations from raw IQ samples.

preprocessing technique is an important contribution in this paper that removes the effect of modulation, artifacts due to HW, and channel. Theoretical derivations are done using the models introduced, to show how preprocessing achieves this for emanation from a single source in Sec. 2.2.5.

The derivations are extended to a use case where multiple emanations are present resulting in multi-harmonics in the received signal, see Sec. 2.2.6. When multi-harmonics are present, there are additional cross terms present in the derivations, compared to single harmonic cases. In Sec. 2.2.6, it is shown how the additional cross-terms in the multi-harmonics use case are reduced, resulting in the removal of artifacts. In Sec. 2.2.7 it is shown how the algorithm deals with overt signals interfering emanations. Overt signals are intentionally transmitted signals such as Wi-Fi, cellular, and Bluetooth.

The preprocessed signal has artifacts removed and has a harmonic structure in the frequency domain. The Welch averaging is used to estimate the Power Spectral Density (PSD) of the preprocessed signal, see Sec. 2.2.8. Periodogram averaging reduces signal variance and improves the performance of the subsequent peak-finding module. However, this reduces the lowest frequency resolution, placing a lower threshold on the detected pitch frequency. Below this threshold, low-pass filtering was applied, and no averaging was performed. Dominant peaks are identified in the frequency domain in both cases using SNR thresholding and prominence-based parameterless distance separation, see Sec. 2.2.9. SNR thresholds are computed based on the signal variance estimates.

The frequency and SNR of dominant peaks in the frequency domain are used to estimate the pitch frequency of each source, see Sec. 2.2.10. This is motivated by [20] where they demonstrate pitch detection on acoustic signals, with a low sample rate and a single set of harmonics. We improve upon this to detect multiple series of harmonics using an iterative procedure. The pitch frequency and an SNR for each harmonic series are reported. Equations are in a discrete-time domain, consistent with code implementation. Frequency domain representation supports the plots in Sec. 2.4.

2.2.1 Justification of choice of algorithm

In this section, the shortcomings of prior work in satisfying the requirements of our application are addressed. The details are provided for the computational load imposed by wireless applications compared to audio.

Pitch detection has been extensively studied for audio signals [21, 22, 23, 24]. Audio applications have been varied such as source separation, enhancement of audio effects, biomedicine, and mechanics. Techniques such as auto-correlation [21], cross-correlation [22], and average square difference [23] use a similarity measure to estimate the pitch frequency. Frequency domain techniques include cepstrum [24] use a dual transform for estimation. Techniques based on parametric estimation theory such as maximum likelihood estimate [25] and maximum a-posteriori [26] have been explored. Techniques using notch filters at frequencies of harmonics to suppress them and minimize the overall output power are in [27, 28], subspace approaches such as Multiple Signal Classification (MUSIC) and Estimation of Signal Parameters by Rotational Invariance Techniques (ESPRIT) are in [29, 30].

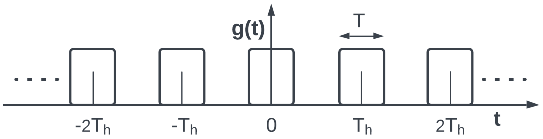


Figure 2.3. Illustration of unmodulated emanation signature: a periodic pulse function with duration T and period T_h.

These approaches assume some of the following in their signal model that limits their application: assumption of white or colored Gaussian noise only and no consideration of channel and clock artifacts, knowledge of the number of sources and number of harmonics in each source, also the number of harmonics for each source are assumed the same. Wireless systems have high sample rates. Here, each IQ capture is processed with 25 MHz bandwidth and 100 ms duration, giving an IQ length is $N = 2.5 \times 10^6$. This is compared to the 44.1 kHz sample rate in audio and speech applications. In [27] that showcase techniques such as non-linear square, subspace-based MUSIC, and filtering-based capon method, dimensions of the matrices operated upon are $O(N)$, where N is IQ length. Operating on matrices with dimensions in millions for wireless applications is computationally not feasible.

Our application requires detecting and characterizing emanations from multiple sources simultaneously. Further, the HW, channel artifacts are different for the wireless systems compared to the audio applications. Further, there could be interferences due to overt signals. The proposed algorithm addresses these requirements.

2.2.2 Harmonic signal model

Models are provided for the signal, modulation, and artifacts to establish algorithm performance through mathematical derivations. The signal model for the harmonics is provided in this subsection. Devices, with microprocessors, generate periodic carrier and clock signals. These have sharp transitions and are approximated as periodic pulse train [16, 18, 17]. Periodic pulse train x, see Fig. 2.3 with period T_h, square pulse p of duration T, is given by [31, Eq.

(3.41)]:

$$x(t) = \sum_k p(t - kT_h), \tag{2.1}$$

$$p(t) = \begin{cases} 1 & |t| < \frac{T}{2} \\ 0 & \frac{T}{2} < |t| < \frac{T_h}{2}. \end{cases} \tag{2.2}$$

The Fourier series representation [31, Eq. (3.44)] containing harmonics at a multiple of pitch frequency $f_h = 1/T_h$ is:

$$x(t) = \frac{T}{T_h} \sum_m \operatorname{sinc}\left(\frac{mT}{T_h}\right) \exp\left(\frac{j2\pi mt}{T_h}\right). \tag{2.3}$$

The signal model for a single harmonic series x_{sh} [27, Eq. (1.1)] with complex amplitude $\alpha_m \in \mathbb{C}$:

$$x_{\mathrm{sh}}[n] = \sum_{m=1}^{M} \alpha_m \exp(jw_h mn), n \in [0, \ldots, N-1], \tag{2.4}$$

$$X_{\mathrm{sh}}(w) = \sum_{m=1}^{M} \alpha_m \delta(w - mw_h), \tag{2.5}$$

where X_{sh} is the Fourier transform of x_{sh}, and M is the number of harmonics in the harmonic series. The signal model x_{sh} is generic and captures the harmonic pattern of the pulse train in (2.3).

The pitch frequency in radians w_h is related to physical frequency in Hz f_h as [27, Eq. (1.4)]:

$$w_h = 2\pi f_h / f_s, \tag{2.6}$$

where f_s is the sampling rate. Frequency in radians w_h is used in equations for conciseness, and frequency in Hz f_h is used in plots for ease of interpretation.

The single harmonic series x_{sh} in (2.4) undergo unintentional modulation at emanation source, channel, HW artifacts, and AWGN noise, resulting in received signal $\boldsymbol{y}$. For multi-harmonic cases, each harmonic series $\boldsymbol{y}_k$ separately undergo unintentional modulation at emanation source, channel, and HW artifacts. They are combined at the receiver with thermal noise $\boldsymbol{w}_{\text{mh}}$ to get receive IQ $\boldsymbol{y}_{\text{mh}}$ as:

$$\boldsymbol{y}_{\text{mh}} = \sum_{k=1}^{K} \boldsymbol{y}_k + \boldsymbol{w}_{\text{mh}}, \tag{2.7}$$

where $\boldsymbol{w}_{\text{mh}}$ is the AWGN noise for multiharmonic usecase. In the following sections, the emanation model is presented. This is followed by showing how preprocessing helps remove artifacts for single and multiharmonic series.

2.2.3 Transmit emanation model

The harmonic signature undergoes unintended modulation that is characteristic of the physics of the HW, and unknown to us. This is captured in the transmit emanation model presented in this subsection, where the harmonics are modulated. In a digital communication system signal modulation is defined as follows. The binary sequence to be transmitted is parsed into subsequences of length K_m, indexed by i. Each subsequence is mapped into a waveform x_i. In a frequency modulation scheme, a frequency shift w_i is applied to a carrier frequency w_c as follows: [32, Eq. (4.8)]:

$$x_i[n] = \exp\left(j\left(w_c + w_i\right)n\right), \quad 1 \leq i \leq 2^{K_m}, \tag{2.8}$$

where 2^{K_m} is the number of waveforms, $n \in [0, \ldots, N-1]$. In contrast to modulation in a digital communication system, unintended modulation occurs on emanation signals. Signal generation circuits of a processor generate periodic clock signals that act as a carrier. Due to space constraints, signal generation and data processing circuits are in proximity, leading to unintentional modulation of the carrier signals [16]. Activities in desktops and laptops modulate the clock signals and apply specific frequency shifts unknown to the user [17, 33]. Examples of activities are based on processor and memory activities. A program is run at an alternating period T_d to achieve a frequency modulation of w_d. Such unintentionally modulated emanation signal with unknown frequency shifts w_d of the harmonics x_{sh} is modeled as x_{tx}:

$$x_{\text{tx}}[n] = x_{\text{sh}}[n] \sum_{d=1}^{D} \alpha_d \exp(jw_d n), \tag{2.9}$$

where w_d, α_d are the d^{th} frequency and complex amplitude shifts applied due to unintended modulation, D number of frequency shifts whose value is unknown due to nature of emanation modulation [17]. Preprocessing operation in Sec. 2.2.5 is shown to remove the effect of frequency modulation w_d, see (2.26).

Inserting (2.4) into (2.9) gives:

$$x_{\text{tx}}[n] = \sum_m \alpha_m \sum_d \alpha_d \exp(j(mw_h + w_d)n). \tag{2.10}$$

2.2.4 Receive emanation model

The transmitted emanation further undergoes HW, channel artifacts, and thermal noise as described in this subsection. Transmit source of emanation and receive HW are considered static, the channel is time-invariant. Transmit emanation x_{tx} impacted by channel impulse response h becomes x_{ch}:

$$x_{\text{ch}}[n] = (x_{\text{tx}} \circledast h)[n], \tag{2.11}$$

Taking Fourier transform upon inserting (2.10) into (2.11):

$$\begin{aligned}X_{\text{ch}}(w) &= X_{\text{tx}}(w)H(w)\\ &= H(w)\sum_m\sum_d \alpha_m\alpha_d\delta(w-mw_h-w_d)\\ &= \sum_m\sum_d h_{m,d}\alpha_m\alpha_d\delta(w-mw_h-w_d),\end{aligned} \tag{2.12}$$

where $h_{m,d} = H(mw_h + w_d)$, $X_{\text{tx}}(w)$ is the Fourier transform of x_{tx} and contains pure tones, represented as Dirac delta functions at $mw_h + w_d$. The effect of the channel artifact is an amplitude and phase shift $h_{m,d}$ that does not impact the frequency of tones. Thus, the channel does not impact the pitch estimation. Inverse Fourier transform of $X_{\text{ch}}(w)$ (2.12) is taken to get $x_{\text{ch}}[n]$ as follows:

$$x_{\text{ch}}[n] = \sum_m\sum_d h_{m,d}\alpha_m\alpha_d \exp(j(mw_h+w_d)n). \tag{2.13}$$

In addition to the channel effect, time-variant clock artifact and thermal noise [34, Eqs. (1.11), (8.5)], impact transmit emanation resulting in received IQ samples as follows:

$$\boldsymbol{y} = \boldsymbol{v} + \boldsymbol{w}. \tag{2.14}$$

where $\boldsymbol{y} = [y[0],\dots,y[N-1]]^T$ is emanations impacted by channel and clock artifacts $\boldsymbol{v} = [v[0],\dots,v[N-1]]^T$, and thermal noise $\boldsymbol{w} = [w[0],\dots,w[N-1]]^T$, such that $\boldsymbol{y},\boldsymbol{v},\boldsymbol{w} \in \mathbb{C}^N$.

Emanation impacted by channel and clock artifacts $\boldsymbol{v}$ result in the Hadamard product ($\odot$) between clock artifacts $\boldsymbol{c}$ and emanation impacted by the channel $\boldsymbol{x}_{\text{ch}}$:

$$\begin{aligned}&\boldsymbol{v} = \boldsymbol{c}\odot\boldsymbol{x}_{\text{ch}},\\ &\boldsymbol{x}_{\text{ch}} = [x_{\text{ch}}[0],\dots,x_{\text{ch}}[N-1]]^T,\\ &\boldsymbol{c} = [\exp(j\beta[0]),\dots,\exp(j\beta[N-1])]^T,\end{aligned} \tag{2.15}$$

where $\beta[n] = w_e[n]n + \theta_e[n]$ [35, pg. 360], w_e, θ_e frequency and phase errors due to imperfect time-variant clocks. Grouping the summation over d gives:

$$\boldsymbol{x}_{\text{ch}} = \sum_d \boldsymbol{x}_d, \tag{2.16}$$

where $\boldsymbol{x}_d = [x_d[0], \dots, x_d[N-1]]^T$ such that $x_d[n]$ is:

$$x_d[n] = \alpha_d \sum_m h_{m,d} \alpha_m \exp(j(w_h + w_d)mn). \tag{2.17}$$

2.2.5 Preprocessing

Thus a mathematical representation has been provided on how the harmonics undergo modulation and artifacts to result in received IQ samples y. The received samples y are processed to extract one or more harmonic series. In this subsection, it is shown how preprocessing removes the effect of modulation, channel, and clock artifacts and helps extract the harmonics. Preprocessing [36, 37, 38, 39] is used in audio applications to reduce artifacts due to reverberations and background noise. For electromagnetic signals in this work, the preprocessing technique is introduced as the Hadamard product of the signal with its complex conjugate. This is computationally simple and equivalent to auto-correlation in the frequency domain of the FFT of the signal with its frequency-reversed copy, as explained below. This is motivated by the time domain auto-correlation used in pitch estimation [40] for audio.

Preprocessing is done by taking the product of each IQ sample y with its complex conjugate y^*:

$$s[n] = y^*[n]y[n]. \tag{2.18}$$

This operation is equivalent to a matched filter [32, Eq. (3.56)] in the frequency domain:

$$\begin{aligned} S(w) = Y^*(w) \circledast Y(w) &= \sum_{w_1} Y^*(w_1) Y(w - w_1) \\ &= \sum_{w_1} Y^*(w_1) Y(-(w_1 - w)), \end{aligned} \tag{2.19}$$

where $Y(w)$ the Fourier transform of y is correlated against its frequency-reversed copy $Y(-w)$.

The harmonic signal x_{sh} is frequency modulated by the source of emanation see (2.9), and affected by the clock, channel artifacts, and thermal noise. Combining (2.14) and (2.18), the feature extracted sample vector $\boldsymbol{s} = [s[0], \ldots, s[N-1]]^T$ is:

$$\boldsymbol{s} = \boldsymbol{y}^* \odot \boldsymbol{y} = \boldsymbol{v}^* \odot \boldsymbol{v} + \boldsymbol{z}_1, \tag{2.20}$$

where reduction is due to the distributive property of the complex conjugate, and the distributive property of Hadamard product. Variants of notation z represent cross terms that do not contain harmonic patterns. The cross-terms $\boldsymbol{z}_1$ in (2.20) is:

$$\boldsymbol{z}_1 = 2\mathfrak{Re}\{\boldsymbol{v}^* \odot \boldsymbol{w}\} + \boldsymbol{w}^* \odot \boldsymbol{w}. \tag{2.21}$$

Emanations impacted by channel and clock $\boldsymbol{v}$ from (2.15) is inserted into (2.20) as follows:

$$\boldsymbol{v}^* \odot \boldsymbol{v} = (\boldsymbol{c} \odot \boldsymbol{x}_{\text{ch}})^* \odot (\boldsymbol{c} \odot \boldsymbol{x}_{\text{ch}}) \tag{2.22a}$$

$$= \boldsymbol{x}_{\text{ch}}^* \odot ((c^* \odot c) \odot \boldsymbol{x}_{\text{ch}}) \tag{2.22b}$$

$$= \boldsymbol{x}_{\text{ch}}^* \odot \boldsymbol{x}_{\text{ch}}, \tag{2.22c}$$

where commutative and associative properties of Hadamard product, and the distributive property of complex conjugate operator over Hadamard product gives (2.22b). Further, the properties $\boldsymbol{c}^* \odot \boldsymbol{c} = J_N$ and $J_N \odot \boldsymbol{x}_{\text{ch}} = \boldsymbol{x}_{\text{ch}}$, where $J_N \in \mathbb{C}^N$ is an all ones vector, gives(2.22c). This removes the time-varying clock artifacts and reduces signal variance. Inserting $\boldsymbol{x}_{\text{ch}}$ from (2.16) into (2.22):

$$\begin{aligned}\boldsymbol{x}_{\text{ch}}^{*} \odot \boldsymbol{x}_{\text{ch}} &= \big(\sum_{d_1} \boldsymbol{x}_{d_1}\big)^{*} \odot \big(\sum_{d_2} \boldsymbol{x}_{d_2}\big) \\ &= \sum_{d=1} \boldsymbol{x}_{d}^{*} \odot \boldsymbol{x}_{d} + \boldsymbol{z}_{2}, \end{aligned} \tag{2.23}$$

where the distributive property of Hadamard product and complex conjugate operator over addition gives (2.23), and $\boldsymbol{z}_2$ contains cross terms as follows:

$$\boldsymbol{z}_2 = \sum_{\substack{\forall d_1, d_2 \\ d_1 \neq d_2}} \boldsymbol{x}_{d_1} \odot \boldsymbol{x}_{d_2}. \tag{2.24}$$

Combining (2.20), (2.22), (2.23), $\boldsymbol{s}$ becomes ($\boldsymbol{z} = \boldsymbol{z}_1 + \boldsymbol{z}_2$):

$$\boldsymbol{s} = \sum_{d} \boldsymbol{x}_{d}^{*} \odot \boldsymbol{x}_{d} + \boldsymbol{z} = \sum_{d} \boldsymbol{s}_{d} + \boldsymbol{z}. \tag{2.25}$$

$s_d[n]$ is computed using $x_d[n]$ from (2.17) as follows:

$$s_d[n] = \sum_{m=-M}^{M} \gamma_{m,d} \exp(jw_h mn), \tag{2.26}$$

where $s_d[n] \in \mathbb{R}$, $\gamma_{m,d} = |\alpha_d|^2 \left|h_{m,d}\right|^2 |\alpha_m|^2 (M - |m|)$ such that $\gamma_{m,d} \in \mathbb{C}$. Preprocessing removes the effect of unknown frequency modulation. Frequency modulation term $\exp(jw_d n)$ in x_d (2.17), is removed in s_d. Inserting (2.26) into (2.25):

$$\begin{aligned} s[n] &= \sum_{m} \exp(jw_h mn) \sum_{d} \gamma_{m,d} + z[n] \\ &= \sum_{m} \gamma_m \exp(jw_h mn) + z[n], \end{aligned} \tag{2.27}$$

whose Fourier transform is:

$$S(w) = \sum_{m} \gamma_m \delta\left(w - m w_h\right) + Z(w), \tag{2.28}$$

where Z is the Fourier transform of $\boldsymbol{z}$. Matrix notation of (2.27) becomes:

$$\boldsymbol{s} = \boldsymbol{E}\boldsymbol{a} + \boldsymbol{z}, \tag{2.29}$$

where matrices $\boldsymbol{E} \in \mathbb{C}^{N\times M}$ the harmonic components, $\boldsymbol{a} \in \mathbb{C}^{N\times 1}$ the complex amplitude, are:

$$\begin{aligned} \boldsymbol{E} &= (e_{nm}), \quad e_{nm} = \exp\left(j w_h m n\right), \\ \boldsymbol{a} &= [\gamma_1, \ldots, \gamma_M]^T. \end{aligned} \tag{2.30}$$

where $m \in [1, \ldots, M]$, $n \in [0, \ldots, N-1]$.

2.2.6 Preprocessing for multi harmonics

The derivations are extended to a multi-harmonic use case where a receiver contains emanation signatures from multiple sources. There are additional cross-terms when compared to a single harmonic use case. The results from Sec. 2.2.5 are used to reduce the terms and extract the harmonic structure in the derivations. Using (2.14) and (2.15), an emanation from each of the K sources impacted by channel and clock artifacts is:

$$\boldsymbol{y}_k = \boldsymbol{c}_k \odot \boldsymbol{x}_{\text{ch}}^k, \tag{2.31}$$

where x_{ch}^k is transmit emanation impacted by channel, clock artifacts $\boldsymbol{c}_k = [\exp(j\beta_k[0]), \ldots, \exp(j\beta_k[N-1])]^T$, $\beta_k[n] = w_e^k[n]n + \theta_e^k[n]$ such that $w_e^k[n]$, $\theta_e^k[n]$ represent frequency and phase errors of k^{th} source due to imperfect clocks. Preprocessing is applied on receive IQ $\boldsymbol{y}_{\text{mh}}$, inserting (2.7) into

(2.20) as follows:

$$\boldsymbol{s}_{\text{mh}} = \boldsymbol{y}^*_{\text{mh}} \odot \boldsymbol{y}_{\text{mh}} = \big(\sum_{k_1} \boldsymbol{y}_{k_1} + \boldsymbol{w}\big)^* \odot \big(\sum_{k_2} \boldsymbol{y}_{k_2} + \boldsymbol{w}\big). \tag{2.32}$$

The properties of commutative, distributive over-addition of the Hadamard product, distributive over-addition, and distributive over Hadamard product of the complex conjugate operator are used to reduce (2.32) into:

$$\boldsymbol{s}_{\text{mh}} = \sum_k \boldsymbol{y}^*_k \odot \boldsymbol{y}_k + \boldsymbol{z}_1, \tag{2.33}$$

$$\boldsymbol{z}_1 = \boldsymbol{w}^* \odot \boldsymbol{w} + 2\sum_k \mathfrak{Re}\left\{\boldsymbol{y}^*_k \odot \boldsymbol{w}\right\} + \sum_{\substack{\forall k_1, k_2 \\ k_1 \neq k_2}} \boldsymbol{y}^*_{k_1} \odot \boldsymbol{y}_{k_2}, \tag{2.34}$$

where $\boldsymbol{z}_1$ represents cross-terms. Using (2.31), $\boldsymbol{y}^*_k \odot \boldsymbol{y}_k$ becomes:

$$\begin{aligned} \boldsymbol{y}^*_k \odot \boldsymbol{y}_k &= \left(\boldsymbol{c}_k \odot \boldsymbol{x}^k_{\text{ch}}\right)^* \odot \left(\boldsymbol{c}_k \odot \boldsymbol{x}^k_{\text{ch}}\right) \\ &= \left(\boldsymbol{x}^k_{\text{ch}}\right)^* \odot \left(\boldsymbol{x}^k_{\text{ch}}\right) \\ &= \big(\sum_{d_1} \boldsymbol{x}^k_{d1}\big)^* \odot \big(\sum_{d_2} \boldsymbol{x}^k_{d2}\big), \end{aligned} \tag{2.35}$$

where (2.22), and (2.23) are used. Further, using (2.23) and (2.24):

$$\boldsymbol{y}^*_k \odot \boldsymbol{y}_k = \sum_d \left(\boldsymbol{x}^k_d\right)^* \odot \left(\boldsymbol{x}^k_d\right) + \boldsymbol{z}_k, \tag{2.36}$$

$$\boldsymbol{z}_k = \sum_{\substack{\forall d_1, d_2 \\ d_1 \neq d_2}} (\boldsymbol{x}^k_{d_1})^* \odot (\boldsymbol{x}^k_{d_2}) \tag{2.37}$$

Using (2.33) and (2.36), $\boldsymbol{s}_{\mathrm{mh}}$ becomes:

$$\boldsymbol{s}_{\mathrm{mh}} = \sum_k \sum_d \boldsymbol{s}_d^k + \boldsymbol{z}_{\mathrm{mh}}, \tag{2.38}$$

$$\boldsymbol{z}_{\mathrm{mh}} = \sum_k \boldsymbol{z}_k + \boldsymbol{z}_1, \tag{2.39}$$

where $\boldsymbol{s}_d^k = \left(\boldsymbol{x}_d^k\right)^* \odot \left(\boldsymbol{x}_d^k\right)$, $\boldsymbol{z}_{\mathrm{mh}} = [z_{\mathrm{mh}}[0], \ldots, z_{\mathrm{mh}}[N-1]]^T$. Using (2.26), $\boldsymbol{s}_d^k = [s_d^k[0], \ldots, s_d^k[N-1]]^T$ becomes:

$$s_d^k[n] = \sum_m \gamma_{m,d}^k \exp(jw_h^k mn), \tag{2.40}$$

Inserting (2.40) into (2.38), s_{mh} gives:

$$\begin{aligned} s_{\mathrm{mh}}[n] &= \sum_k \sum_m \exp(jw_h^k mn) \sum_d \gamma_{m,d}^k + z_{\mathrm{mh}}[n] \\ &= \sum_k \sum_m \gamma_m^k \exp(jw_h^k mn) + z_{\mathrm{mh}}[n], \end{aligned} \tag{2.41}$$

where $\gamma_m^k = \sum_d \gamma_{m,d}^k$. Fourier transform of s_{mh} gives:

$$S_{\mathrm{mh}}(w) = \sum_k \sum_m \gamma_m^k \delta(w - mw_h^k) + Z_{\mathrm{mh}}(w), \tag{2.42}$$

where Z_{mh} is the Fourier transform of z_{mh}. Matrix notation of (2.41) is:

$$\boldsymbol{s}_{\mathrm{mh}} = \boldsymbol{E}_k \boldsymbol{a}_k + \boldsymbol{z}_{\mathrm{mh}}, \tag{2.43}$$

where $\boldsymbol{E}_k \in \mathbb{C}^{N \times M}$ is harmonic components, $\boldsymbol{a}_k \in \mathbb{C}^N$ the complex amplitude, are:

$$[\boldsymbol{E}_k]_{nm} = \exp\left(jw_h^k mn\right), \quad \boldsymbol{a}_k = [\gamma_1^k, \ldots, \gamma_M^k]^T,$$

where $n \in [0, \ldots, N-1]$, $m \in [1, \ldots, M]$. Thus it has been theoretically shown for multi-harmonics

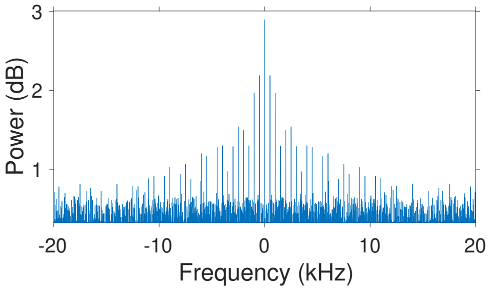


Figure 2.4. FFT over absolute of OOK signal which is an overt signal with patterns similar to emanation patterns with harmonics at multiples of the symbol rate. The signal has return-to-zero (RZ) signaling and 500 Hz symbol rate.

use-case, the removal of artifacts, and extraction of harmonic structure, see (2.41) and (2.42).

2.2.7 Overt signals

Intentionally transmitted signals such as Bluetooth, cellular, and Wi-Fi signals are overt signals x_o. These are included in the signal model in (2.7) to discuss the impact of overt on pitch estimation. Overt signal that undergoes channel and clock artifacts is expanded using (2.11), (2.14):

$$y_o[n] = \exp\left(j\beta[n]\right)\left(x_o \circledast h\right)[n]. \tag{2.44}$$

Combining $\boldsymbol{y}_o$=$[y_o[0],\ldots,y_o[N-1]]^T$ with received emanations $\boldsymbol{y}_k$ and thermal noise $\boldsymbol{w}_{\text{mh}}$ (2.7), the received IQ $\boldsymbol{y}$ is:

$$\boldsymbol{y} = \sum_{k=1}^{K} \boldsymbol{y}_k + \boldsymbol{y}_o + \boldsymbol{w}. \tag{2.45}$$

Applying preprocessing on receive IQ $\boldsymbol{y}$ following (2.32):

$$\boldsymbol{s}_o = \boldsymbol{y}^* \odot \boldsymbol{y} = \boldsymbol{s}_{\text{mh}} + \boldsymbol{z}_o, \tag{2.46}$$

where $\boldsymbol{s}_{\mathrm{mh}}$ is the preprocessing applied to the received signal from multiple emanation sources from (2.32), $\boldsymbol{z}_o$ containing additional terms is:

$$\boldsymbol{z}_o = 2\mathfrak{Re}\left\{\boldsymbol{y}_o^* \odot \boldsymbol{w}\right\} + 2\sum_k \mathfrak{Re}\left\{\boldsymbol{y}_k^* \odot \boldsymbol{y}_o\right\} + \boldsymbol{y}_o^* \odot \boldsymbol{y}_o. \tag{2.47}$$

The term $\boldsymbol{y}_o^* \odot \boldsymbol{w}$ represents cross-correlation between overt and thermal noise and $\boldsymbol{y}_k^* \odot \boldsymbol{y}_o$ between emanation source and overt, $\boldsymbol{y}_o^* \odot \boldsymbol{y}_o$ auto-correlation of overt. The first term is uncorrelated and does not result in a harmonic pattern. The second and third terms result in a harmonic pattern only if the overt signal has a harmonic pattern.

An On-Off-Keying (OOK) modulation signal is an example of an overt signal that could be confused with an emanation, the harmonics in the OOK signal are in Fig. 2.4. In this illustration, the OOK modulated signal is synthetically generated over random information bits, preprocessing, and FFT applied. OOK is a binary amplitude shift keying modulation type [32, Eq. (4.9)], see below:

$$x(t) = b\left(u(t+T) - u(t)\right), \quad b \in \{0,1\}, \tag{2.48}$$

where u is the unit impulse function, T duration of symbol, b is binary information bit. OOK is not widely used in digital communication systems [32, pg. 175]. Thus it is shown overt signals do not impact algorithm performance, except for corner cases as detailed.

2.2.8 Power spectral density estimation

The preprocessed signal has artifacts removed and harmonic structure retained. It is necessary to reduce the signal variance of preprocessed signal and transform it into the frequency domain for peak finding. This is achieved using the Welch method [41] discussed in this section. Dominant peaks are identified in the PSD of $s_{\mathrm{o}}[n]$. The frequency and SNR of the dominant peaks are used to estimate the pitch in Sec. 2.2.10. Large signal variance results in picking false peaks and missing true peaks. This impacts the performance of pitch estimation. The Welch method is used to estimate the PSD to reduce the signal variance. The description of this method in this subsection closely follows [42, pg. 730]. It involves splitting the time series into smaller overlapping segments, that are uncorrelated, and their modified periodograms are averaged to reduce variance.

The direct current component at zero frequency could leak due to windowing and obscure the low-frequency components and therefore the mean is subtracted as follows:

$$\bar{s}_{\mathrm{o}}[n] = s_{\mathrm{o}}[n] - \frac{1}{N}\sum_{n=0}^{N-1} s[n], \tag{2.49}$$

Further, $\bar{s}_{\mathrm{o}}[n]$ of length N is broken into smaller overlapping segments $s_i[n]$ of length L, where $i \in [0, N_s - 1]$, the number of segments $N_s = \lfloor \frac{N}{L_o} \rfloor$ [42, Eq. (10.67)]:

$$s_i[n] = \bar{s}_{\mathrm{o}}[(i-w)L_o + n]v[n], n \in [0, L-1], \tag{2.50}$$

and overlap length is $L - L_o$ samples, $v[n]$ is the Kaiser window of length L [42, Eq. (7.59)]:

$$v[n] = \frac{I_o\left(\beta\sqrt{1 - (\frac{2n}{L-1} - 1)^2}\right)}{I_o(\beta)}, \tag{2.51}$$

where I_o zeroth-order modified Bessel function of the first kind parameterized by β. The modified periodogram $P_i(w_l)$ is for each segment at frequencies, $w_l = \frac{2\pi l}{L}$, $l \in [0, L-1]$:

$$P_i(w_l) = \frac{1}{LV}\left|\sum_{n=0}^{L-1} s_i[n]\exp(jw_l n)\right|^2, \tag{2.52}$$

$$V = \frac{1}{L}\sum_{l} |v[l]|^2. \tag{2.53}$$

Averaging the periodograms $P_i(w_l)$ gives the PSD estimate:

$$P(w_l) = \frac{1}{N_s}\sum_{i=1}^{N_s} P_i(w_l). \tag{2.54}$$

Further, the variance is shown to reduce by a factor of N_s [42, Eq. (10.75)]:

$$Var(P(w_l)) = \frac{1}{N_s} Var(P_s(w_l)) \tag{2.55}$$

where $P_s(w_l)$ is the PSD of $s_{\text{o}}[n]$ without periodogram averaging. The PSD estimated in (2.54), is used as input by the subsequent peak finding block.

For fixed-length IQ samples, there is a tradeoff between frequency resolution and signal variance. The frequency resolution, which is equal to the main lobe width of the Kaiser window is $w_{\text{res}} = \left(\sqrt{1+\frac{\beta}{\pi}^2}\right)\frac{2\pi}{L}$ [43]. Consider adjacent peaks that are part of harmonic series as $(m+1)w_h$ and mw_h for $\forall m$, their separation is w_h. To resolve two adjacent peaks, the frequency separation should be greater than the frequency resolution w_{res}:

$$w_h \geq w_{\text{res}} = \left(\sqrt{1+\left(\frac{\beta}{\pi}\right)^2}\right)\frac{2\pi}{L}. \tag{2.56}$$

Thus the window length L restricts the pitch frequency w_h that can be detected. Periodogram averaging increases frequency resolution w_{res} by a factor $\frac{N}{L}$. To detect pitch frequency less than w_{res}, a modified periodogram (2.52) without averaging, is computed on s_{mh} at frequencies $w_n = \frac{2\pi n}{N}$, $n \in [0, N-1]$. Inserting s_{mh} from (2.41) into (2.46), and computing Welch based PSD that includes periodogram averaging:

$$P(w_l) = \sum_k \sum_m \eta_m^k \delta(w_l - m w_h^k) + P_z(w_l), \tag{2.57}$$

where $P_z(w_l)$ represents terms not containing harmonics, η_m^k is the power of the harmonic peaks as follows:

$$\eta_m^k = \left|\gamma_m^k\right|^2 + 2\left|Z_{\text{mh}}(m w_h^k)\right| + 2\left|Z_{\text{o}}(m w_h^k)\right| \tag{2.58}$$

where $Z_{\text{mh}}(w)$, $Z_{\text{o}}(w)$ are the Fourier transforms of $z_{\text{mh}}[n]$ from (2.43) and $z_o[n]$ from (2.47).

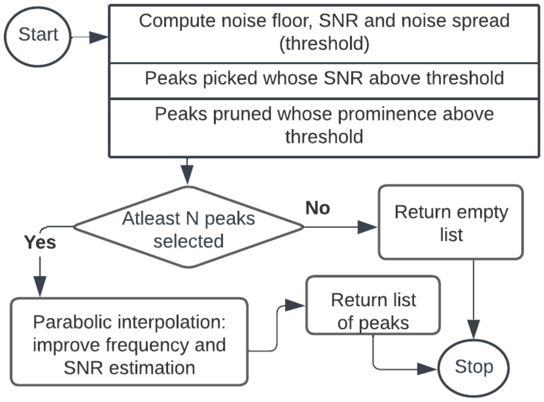

Start
Compute noise floor, SNR and noise spread (threshold)
Peaks picked whose SNR above threshold
Peaks pruned whose prominence above threshold
Atleast N peaks selected
No
Return empty list
Yes
Parabolic interpolation: improve frequency and SNR estimation
Return list of peaks
Stop

Figure 2.5. Peak detection flow chart. Input to the peak detection block is the PSD of the preprocessed signal. It outputs a list of dominant peaks with frequency and SNR values to the subsequent pitch estimation block.

2.2.9 Peak finding

This section describes the identification of peaks in the PSD of the preprocessed signal. The peak detection flowchart is in Fig. 2.5. Peak detection is extensively used in biomedical signal processing [44, 45]. Peaks are commonly identified by searching local maxima whose SNR exceeds a threshold. Biomedical signal peaks have specific patterns that are utilized to estimate the noise floor and SNR accurately [45]. A robust percentile-based approach is used to estimate the noise floor and threshold [44, 45, 46], which does not assume a specific model for signal peaks.

The frequency and SNR of the detected peaks are passed onto the subsequent pitch estimation block. The peaks with SNR exceeding a given threshold are picked. They are pruned further using the prominence metric. The noise floor is the median of the signal [47]. To handle non-flat noise floor, the spectrum is split into N_f narrow frequency slices of length $L_f = L/N_f$, where L number of samples in PSD. The PSD in dB of *i*th frequency slice is:

$$P^i_{\text{dB}}(w_l) = P_{\text{dB}}(w_{(i-1)L_f+l}), l \in [0, L_f - 1], \tag{2.59}$$

where $P_{\text{dB}}(w_l) = 10\log_{10} P(w_l)$. The noise floor of i^{th} slice NF$_i$ is the median of $P^i_{\text{dB}}(w_l)$. The threshold is calculated as twice the estimated standard deviation σ of signal power around the mean. Assuming a Gaussian distribution for the noise, 68% of data is contained within σ around the mean. The estimated standard deviation for the i^{th} frequency slice is calculated as follows:

$$\sigma_i = \frac{\mathrm{PCT}(P_{\mathrm{dB}}^i, 84\%) - \mathrm{PCT}(P_{\mathrm{dB}}^i, 16\%)}{2}, \tag{2.60}$$

where PCT is the percentile function. There could be parts of the spectrum with stronger interferences and overt signals. A median is taken for estimated standard deviations, across frequency slices:

$$\sigma_t = \mathrm{PCT}([\sigma_1, \ldots, \sigma_{N_f}], 50\%). \tag{2.61}$$

The presence of overt signals and interferences would bias the noise floor and standard deviation calculation. Preprocessing removes the effect of overt signals occupying large frequency bands and thus aids in an accurate estimate.

Points of local maxima are identified at frequencies w_p and their SNR is computed as:

$$\mathrm{SNR}(w_p) = P_{\mathrm{dB}}(w_p) - \mathrm{NF}_i, \tag{2.62}$$

where i is the frequency slice containing w_p. These peaks at w_p are trimmed based on SNR exceeding the threshold as $\mathrm{SNR}(w_p) > 2n_\sigma\sigma_t$, where n_σ is a hyper-parameter chosen empirically.

Peaks filtered by the SNR threshold might have false peaks close to a true peak. Explicitly specifying a distance separation is not robust for detecting emanations as the pitch frequency could vary across frequencies. The prominence metric is a parameterless minimum distance separation to filter false peaks close to a true peak. This metric is motivated by topographical prominence in geology [48]. The peak prominence is its height relative to the lowest contour line:

$$\mathrm{Prominence}(w_p) = P_{\mathrm{dB}}(w_p) - P_{\mathrm{dB}}(w_{\mathrm{prom}}) \tag{2.63}$$

where w_{prom} is calculated as:

$$w_{\mathrm{prom}} = \underset{w_m}{\operatorname{argmax}}\{P_{\mathrm{dB}}(w_m) : P_{\mathrm{dB}}(w_m) \leq P_{\mathrm{dB}}(w_p)\}, \tag{2.64}$$

and w belongs to a set of points of local minima:

$$w_m \in \{w : P_{\mathrm{dB}}(w - w_l) \geq P_{\mathrm{dB}}(w) \leq P_{\mathrm{dB}}(w + w_l)\} \ . \tag{2.65}$$

This metric has been used in the fields of biomedical [49] and speech signal processing [50].

Due to discrete sampling instants not coinciding with the true peaks, there is an error in the frequency w_p and power $P_{\mathrm{dB}}(w_p)$ estimates of the identified peaks. This error in each peak δw_p is bound by $\left|\delta w_p\right| \leq \frac{w_{\mathrm{res}}}{2}$, where w_{res} is the frequency resolution, see (2.56). Parabolic interpolation [51] is used to get a better estimate of frequency and power. For a peak at w_p, the points at w_{p-1}, w_p and w_{p+1} are fitted to a parabola in the coordinate system centered at $(w_p, 0)$:

$$P_{\mathrm{dB}}(\overline{w}) = a(\overline{w} - \overline{w}_{\mathrm{p}})^2 + b, \tag{2.66}$$

where $\overline{w}$ is the frequency, $\overline{w}_p$ peak frequency in the new coordinate system. The estimated peak frequency w_{p} is [51]:

$$\overline{w}_{\mathrm{p}} = \frac{\pi}{L}\left(\frac{P_{\mathrm{dB}}\left(\frac{-2\pi}{L}\right) - P_{\mathrm{dB}}\left(\frac{2\pi}{L}\right)}{P_{\mathrm{dB}}\left(\frac{-2\pi}{L}\right) + P_{\mathrm{dB}}\left(\frac{2\pi}{L}\right) - 2P_{\mathrm{dB}}(0)}\right), \tag{2.67}$$

where $\frac{-2\pi}{L}$, 0, $\frac{2\pi}{L}$ are the frequency of samples interpolated in the new coordinate system. The power at w_{p} is:

$$P_{\mathrm{dB}}(w_{\mathrm{p}}) = P_{\mathrm{dB}}\left(0\right) - \frac{1}{4}\left(P_{\mathrm{dB}}\left(\frac{-2\pi}{L}\right) - P_{\mathrm{dB}}\left(\frac{2\pi}{L}\right)\right). \tag{2.68}$$

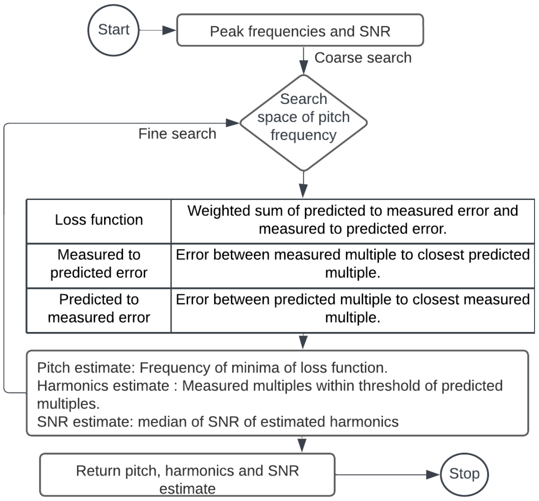

Start
Peak frequencies and SNR
Coarse search
Search space of pitch frequency
Fine search
Loss function
Weighted sum of predicted to measured error and measured to predicted error.
Measured to predicted error
Error between measured multiple to closest predicted multiple.
Predicted to measured error
Error between predicted multiple to closest measured multiple.
Pitch estimate: Frequency of minima of loss function.
Harmonics estimate : Measured multiples within threshold of predicted multiples.
SNR estimate: median of SNR of estimated harmonics
Return pitch, harmonics and SNR estimate
Stop

Figure 2.6. Pitch estimation block. Input: list of peak frequencies and SNR. Output: provides pitch frequencies, corresponding multiples, and SNR.

2.2.10 Pitch Estimation

The frequency and SNR of dominant peaks are used to find the pitch frequency. There could be one or more harmonic series. The intention is to detect each of them and estimate the corresponding pitch. The pitch estimation procedure in [20] is improved upon to detect multiple harmonic series, for the wireless signals in this work. The flowchart for pitch estimation is in Fig. 2.6.

The measured multiples are the peaks w_p identified in the PSD of preprocessed IQ in section 2.2.9. PSD is assumed to contain peaks at harmonics that are integral multiple of the pitch at w_h^k. These harmonics are referred to as predicted multiples. The set of measured multiples $\mathscr{M} = w_p : \forall p$ are assumed to include K series of harmonics from K source of emanations:

$$\mathscr{M} = F_1 \bigcup F_2 \bigcup \ldots \bigcup F_K \bigcup F_p, \tag{2.69}$$

where $F_k = m w_h^k : m \in [1, \ldots, M]$ is the set of harmonics for k^{th} source, F_p is the set of false peaks due to noise, interferences, spectral leakage, etc.

An iterative process estimate pitch w_h^k of k^{th} emanation source as:

$$w_h^k = \underset{w}{\operatorname{argmin}}\, \mathscr{L}(w), w \in \mathscr{S}_w. \tag{2.70}$$

where w is the frequency, $\mathscr{S}_w$ the frequency search space. The loss function $\mathscr{L}(w)$ considers the following factors: higher loss for a larger difference in frequency between predicted and measured multiple, higher loss for lower SNR valued measured multiple. A measured multiple is considered part of the harmonic series if within a threshold percentage error pt of the predicted multiple:

$$F_k = \{w_p : \left| w_p - m w_h^k \right| \leq \frac{\text{pt}}{100} w_h^k\}. \tag{2.71}$$

Set of measured multiples is updated as $\mathscr{M} \equiv \mathscr{M} - F_k \equiv \{w \in \mathscr{M} : w \notin F_k\}$. Pitch estimation for a new harmonic series w_h^{k+1}, is thus iteratively attempted on the updated set $\mathscr{M}$.

The list of multiples F_k is removed from $\mathscr{M}$ and pitch estimation of w_h^{k+1} is again attempted on the set of peaks $\mathscr{M} - F_k \equiv \{w \in \mathscr{M} : w \notin F_k\}$. This process is recursive until the pitch corresponding to each of the K sources is estimated.

The loss function is calculated as follows:

$$\mathscr{L}(w) = L_{\text{pm}}(w) + \alpha_w L_{\text{mp}}(w), \tag{2.72}$$

where L_{pm} is the cumulative error in matching predicted to measured multiples, L_{mp} measured to predicted multiples, α_w weightage given to L_{mp}.

The loss L_{pm} is computed iterating over predicted multiples penalizing the mismatch to its closest measured multiple:

$$L_{\text{pm}}(w) = \sum_{n=1}^{N_f} \frac{\left| nw - w_{p_n} \right|}{(nw)^{q_1}} \left(1 + q_2 (A_{p_n})^{q_3}\right), \tag{2.73}$$

where q_1, q_2, q_3 are hyper-parameters chosen empirically, A_{p_n} is SNR of peak w_{p_n}, $N_f = \frac{max(\mathscr{M})}{w}$.

The index of measured multiple p_n, closest to a predicted multiple at nw is:

$$p_n = \operatorname*{argmin}_{p} \left| nw - w_p \right|. \tag{2.74}$$

Since iteration is over predicted multiples, there is no penalty for unaccounted measured multiples $\mathscr{M} - F_k \equiv F_{k+1} \bigcup \ldots \bigcup F_p$ in loss function L_{pm}. These are instead considered in L_{mp}. The unaccounted measured multiples are due to the presence of multiple sources of emanations and false peaks. Similarly L_{mp} does not penalize for unaccounted predicted multiples. The unaccounted predicted multiples occur due to the low SNR of corresponding measured multiples. These measured multiples are not picked in the peak finding block described in Sec. 2.2.9 due to low SNR. The low SNR could be due to factors such as the nature of emanation, high noise, and interferences. The loss functions are combined in (2.72) to estimate the pitch.

The loss function L_{mp} iterates over the measured multiple, penalizing the mismatch with the closest predicted multiple:

$$L_{\text{mp}}(w) = \sum_{p=1}^{P} \frac{\left| n_p w - w_p \right|}{(w_p)^{q_1}} \left(1 + q_2 (A_p)^{q_3}\right), \tag{2.75}$$

where A_p is SNR of peak w_p, the index of predicted multiple n_p closest to the measured multiple w_p is:

$$n_p = \operatorname*{argmin}_{n} \left| nw - w_p \right|. \tag{2.76}$$

Thus mathematical derivations have been provided demonstrating the performance of the algorithm. In the following section, the emanation detection performance of the algorithm on real IQ data is shown.

2.3 Experimental Setup

The signal hound SDR captures IQ samples from sources of emanation inside a shielded room, see Fig. 2.7. An antenna is placed inside the room 2.5 meters from the source of emanation.

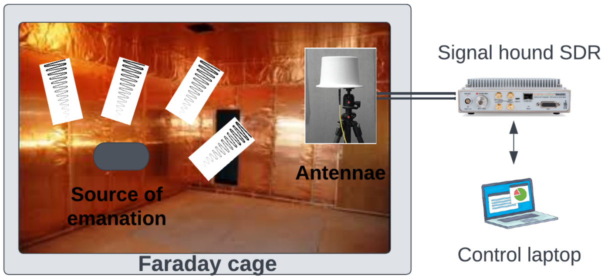


Figure 2.7. Shielded room acting as a Faraday cage. Used to capture data in a controlled RF environment, that blocks RF signals from external sources.

Table 2.1. Parameters used

Type	Details
IQ	Freq. captured: 0.1–1.1 GHz, Bandwidth and duration of IQ capture: 200 MHz (max bandwidth of SDR) and 100 ms, Bandwidth and duration of input to the algorithm: 25 MHz and 100 ms.
PSD	Ensemble duration: 1 ms, Percentage overlap: 75, Window: Kaiser with beta of 10.
Pitch estimation	Error threshold: 10% for pitch freq. ¡ 500 Hz and 2% otherwise, q_1 = 0.5, q_2 = 1, q_3 = -1, $\alpha_w = 10^{-3}$ for high pitch, 100 for low pitch, n_σ = 2.

To ensure there are no emanations due to the SDR, both SDR and SDR-controlling laptop are outside the room. The room is sanitized by collecting IQ from an empty room with no emanations. No emanations are detected when the IQ is passed through the emanation detection algorithm.

Sources of emanation of interest are placed inside a shielded room, with parameters of IQ capture in Table. 2.1. The 200 MHz is the maximum bandwidth capture for the Signalhound SDR and is split into 25 MHz slices that are processed. The choice of 25 MHz processing bandwidth and 100 ms capture duration is a balance between computational load and algorithm

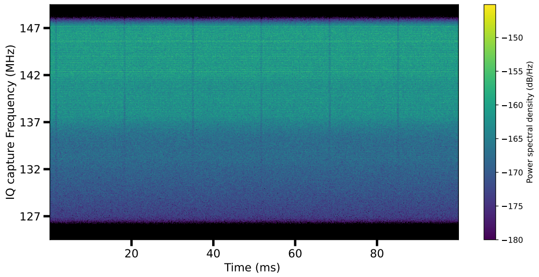
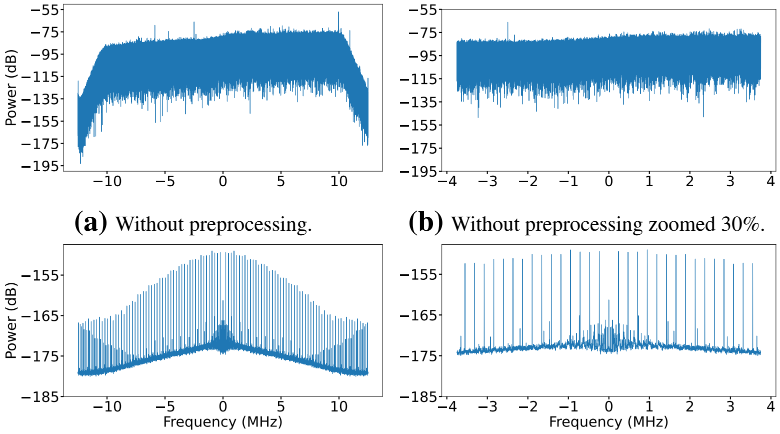


Figure 2.8. Spectrogram of emanations from a laptop connected to a monitor via an HDMI to USB-C adaptor.



(a) Without preprocessing.

(b) Without preprocessing zoomed 30%.

(c) With preprocessing.

(d) With preprocessing zoomed 30%.

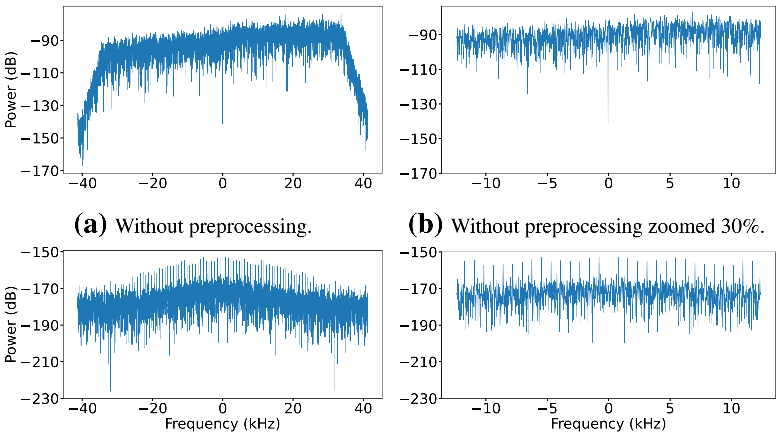
Figure 2.9. High pitch detection: PSD over IQ with and without preprocessing. It captures emanations from a laptop connected to a monitor via an HDMI to USB-C adaptor, the center frequency is 137.5 MHz. (a) A noise-like signal is observed without preprocessing. (c) A clear harmonic pattern was observed with prominent peaks, after preprocessing. Subplots (b), and (d) are zoomed versions of (a), and (c) and are provided for details.

performance: Fixing the capture duration, a larger processing bandwidth means more samples and computation but restricts the highest pitch frequency of the harmonics that can be estimated. Similarly, fixing processing bandwidth, and increasing the duration of capture increases SNR gains obtained from periodogram averaging, see Sec. 2.2.8 but increases computational load.

2.4 Results

The focus is finding anomalous activities by learning and tracking emanation patterns. Emanations identified from narrow frequency slices are grouped to build the emanation pattern. A change in the pattern from baseline is a potential anomalous activity. The algorithm in Sec. 2.2 is verified on real IQ data in this section. The performance is shown first on a single frequency slice for the case of a laptop connected to a monitor, highlighting algorithm performance in a step-by-step manner. This is followed by learning emanation patterns across 1 GHz bandwidth for a laptop and a desktop connected to a monitor and for IQ collected from cases emulating anomalous activity.

An emanation corresponds to a harmonic series with a pitch at f_1. There could be



(a) Without preprocessing. **(b)** Without preprocessing zoomed 30%.

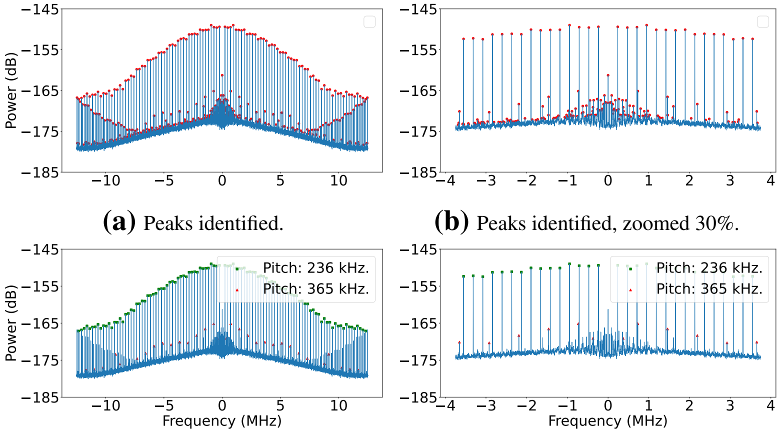
(c) With preprocessing. **(d)** With preprocessing zoomed 30%.

Figure 2.10. Caption same as Fig. 2.9 but for low pitch detection.

multiple emanations corresponding to multiple harmonic series with pitch at f_1, f_2, etc. Physical frequency f (2.6) is more intuitive and used in illustrations, compared to the frequency in radians in Sec. 2.2.

The algorithm is demonstrated step-by-step for a laptop connected to a monitor via an adaptor USB-C to HDMI. The spectrogram is in Fig. 2.8. Mildly visible periodic lines are seen in the spectrogram, across both time and frequency axes. It is interesting to detect the periodicity of these signatures and those not visible in the spectrogram. The center frequency used in the spectrogram and estimated pitch plots is the actual IQ capture frequency. All other plots use a baseband frequency where the center frequency is zero, for ease of illustration.

The PSD with and without preprocessing is presented. This illustrates the effect of preprocessing in removing artifacts. This is done for both low and high-pitch detection. Estimation of pitch frequencies is done separately for low and high pitch frequencies. This is to accommodate the tradeoff between frequency resolution and signal variance, see Sec. 2.2. In high pitch detection, the PSD of the raw IQ is in Fig. 2.9a with no visible signatures. In Sec. 2.2.5, preprocessing is shown to deal with artifacts. PSD on preprocessed signal is in Fig. 2.9c. Notice the peaks with harmonic patterns post preprocessing, similar to the form in (2.42). There are one or more harmonic series in the PSD which can be estimated.



(a) Peaks identified.

(b) Peaks identified, zoomed 30%.

(c) Harmonics identified amongst peaks.

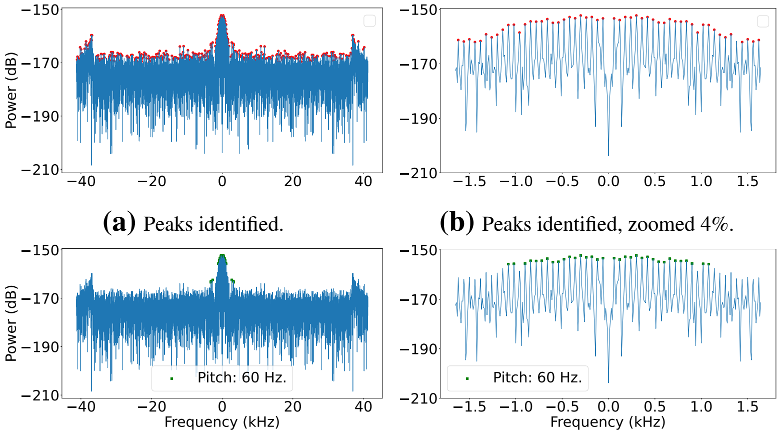
(d) Harmonics identified, zoomed 30%.

Figure 2.11. High pitch detection: Identified peaks are overlayed on PSD on the left side of the diagram. The pitches estimated, and peaks identified as belonging to the harmonics are on the right side. Use-case: IQ from a laptop connected to a monitor via an HDMI to USB-C adaptor, the center frequency is 137.5 MHz. (a) Peaks identified. (c) Harmonics of estimated pitches. Estimated pitch frequencies are 236 kHz and 365 kHz. Subplots (b) and (d) are zoomed versions of (a), and (c) and are provided for details.

Similarly, for low pitch detection, PSD with and without preprocessing is in Fig. 2.10a and 2.10c. Notice the larger signal variance in the PSD, compared to the PSD for high pitch detection in Fig. 2.9a, 2.9c. Detection of low-frequency pitch necessitates lower-frequency resolution. Therefore no periodogram averaging is done in computing PSD which results in high signal variance.

In the PSD, peaks are detected using the algorithm in Sec. 2.2.9, the challenge is to detect peaks in the presence of noise. This is done by finding local maxima in the PSD whose SNR exceeds a threshold, these are calculated from PSD using (2.62) and (2.61). Further prominence as a distance-based metric removes false peaks. Peaks thus identified are shown in Fig. 2.11a for high pitch, and in Fig. 2.12a for low pitch.

The detected peak frequency and SNR are fed to the subsequent pitch estimation block in Sec. 2.2.10. The loss function is computed for every candidate pitch frequency from frequency search space, see (2.73), (2.75) and (2.72). Pitch is estimated as the frequency where the loss function is minimum. Harmonics of the estimated pitch are estimated using (2.71). The pitch



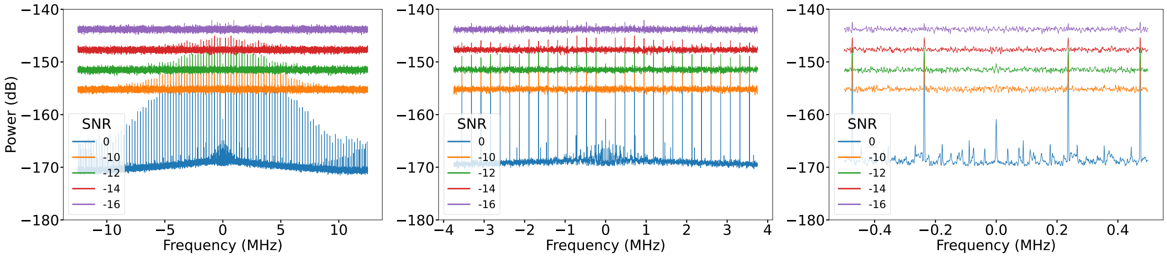
(a) Peaks identified.

(b) Peaks identified, zoomed 4%.

(c) Harmonics identified amongst peaks.

(d) Harmonics identified, zoomed 4%.

Figure 2.12. Caption same as Fig. 2.11 but for low pitch detection.



(a) PSD at various noise SNRs.

(b) PSD at various noise SNRs, zoomed 30%.

(c) PSD at various noise SNRs, zoomed 4%.

Figure 2.13. PSD of preprocessed IQ, highlighting the impact of noise on peaks. Thermal noise is synthetically added at specified SNR levels to IQ from emanations of a laptop connected to a Monitor. Emanations are detected up to SNR as low as -14 dB.

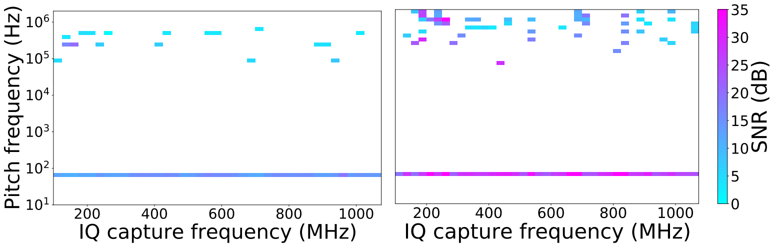
is valid only if at least 5 harmonics are identified using (2.71). The pitches and corresponding harmonics for high pitch detection are in Fig. 2.11c and for low pitch detection in Fig. 2.12c. The IQ capture from 125 to 150 MHz contains pitch at frequencies 60 Hz, 236 kHz, and 365 kHz.

The algorithm is stress tested by synthetically adding AWGN noise w_s at various SNRs, to received signal y:

$$y_s = y + w_s. \tag{2.77}$$

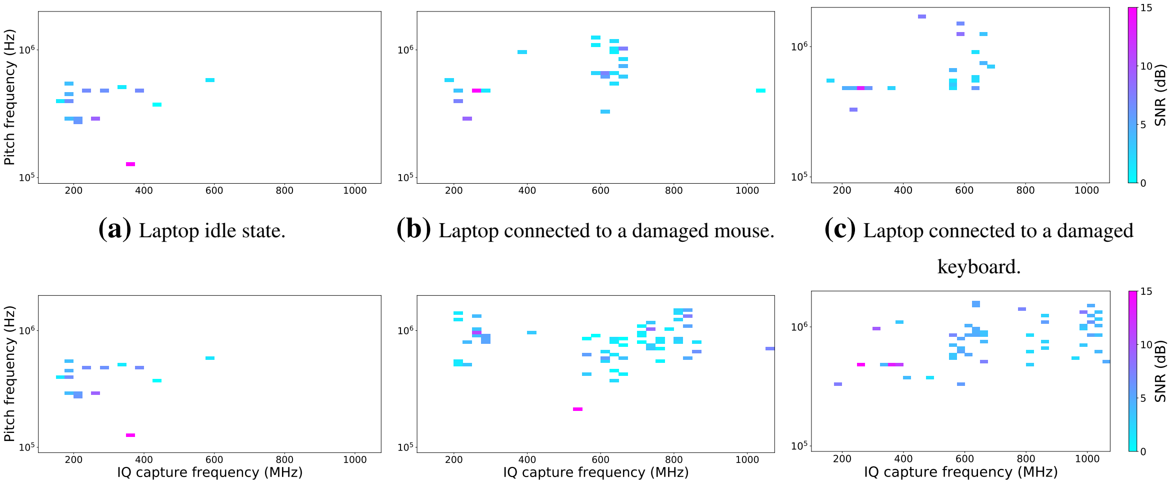
The SNR is defined as,

$$SNR = 10\log_{10}\left(\frac{||y||^2}{||w_s||^2}\right), \tag{2.78}$$



(a) Laptop connected to a Monitor. **(b)** Desktop connected to a Monitor.

Figure 2.14. Emanation patterns for (a) a laptop and (b) a desktop connected to a monitor via an HDMI to USB-C adaptor. Emanations detected in each of the 25 MHz slices of IQ data are plotted. The desktop has CPU-intensive processes running, resulting in more emanations with higher SNR spread across wider capture frequencies, compared to a laptop.



(a) Laptop idle state. **(b)** Laptop connected to a damaged mouse. **(c)** Laptop connected to a damaged keyboard.

(d) Laptop idle state. **(e)** Laptop to SD card, active data transfer. **(f)** Laptop to Samsung pen drive, active data transfer.

Figure 2.15. Use case demonstrating detection of anomalous activity. Emulating wear and tear of electronics: emanation pattern of Laptop (a) in idle state, (b) connected to a damaged mouse, (c) connected to a damaged keyboard. Emulating illegal copying of secure data: emanation pattern of Laptop (d) in idle state, (e) connected to SD card with active data transfer, (f) connected to pen drive with active data transfer. Emanations patterns in (b), and (c) differ from baseline (a), similarly for (e), and (f) over (d), indicating detection of anomalous activity.

where $||y||^2$ is computed empirically as the power of y:

$$||y||^2 = \frac{1}{N}\sum_i \left(y_I^2[i] + y_Q^2[i]\right), \tag{2.79}$$

where y_I and y_Q are the I and Q of the complex receive sample y as follows: $y = y_I + jy_Q$. The

AWGN noise w_s is a complex Gaussian distribution [52, A.1.3]: $w_s \sim \mathscr{C}\mathscr{N}(0, \sigma_{\mathrm{w_s}})$, where variance $\sigma_{\mathrm{w_s}}$ is:

$$\sigma_{\mathrm{w_s}} = ||y||^2 10^{-\frac{SNR}{10}}. \tag{2.80}$$

w_I, w_Q are sampled from $N(0, \frac{\sigma_{\mathrm{w_s}}}{2})$. Synthetic AWGN noise is added to IQ data of emanations of a laptop connected to a monitor. The PSD over preprocessed IQ is shown in Fig. 2.13, for various levels of SNR. The algorithm detects the fundamental harmonic series at 236 kHz until around -14 dB SNR. This highlights the robustness of the algorithm. For detection of the fundamental frequency at SNRs below -10 dB, $q1 = 0.9$ is used compared to default in Table 2.1.

Further emanation patterns are presented for a laptop, and a desktop connected to a monitor, see Fig. 2.14a and 2.14b. The x-axis represents the center frequency at which IQ samples are captured, y-axis represents the pitch frequency of detected emanations. The laptop is in an idle state, desktop has CPU CPU-intensive processes running. This causes crowded emanations in the plot with stronger SNR, compared to the emanation pattern of a laptop. The wideband pitch at 60 Hz corresponds to the leakage from the monitor with a 60 Hz refresh rate.

Detection of anomalous activity is illustrated using two use cases as described below. Wear and tear of the mouse and keyboard are emulated by exposing the copper wire by removing cable shielding in a small area. IQ data is collected from the damaged mouse and keyboard that are actively used. Emanation patterns detected on the data are in Fig. 2.15. In the baseline of an idle laptop only, the emanation pattern has pitches detected more in the lower IQ capture frequencies. For damaged peripherals, the plot has more emanations at higher frequency regions compared to the baseline, indicating potential anomalous activity.

For the second use case, IQ data is collected from an SD card and a pendrive with active data transfer with a laptop. The emanation patterns detected are in the bottom row of Fig. 2.15. Notice more number of emanations detected at higher IQ capture frequencies when there is active data transfer to external storage devices compared to an idle laptop, indicating anomalous activity. This is the first effort towards a generic HW agnostic solution in detecting anomalous activity using emanations. Therefore the focus is on detecting emanations and establishing that emanation patterns can be used to detect anomalous activity. The emanation pattern plots in Fig. 2.15 are a proof of concept demonstrating this.

This work is the first effort toward using emanations in an HW-agnostic manner to detect anomalous activity. The efforts taken in this paper make assumptions and have limitations that are stated as follows. Theoretically, it has been shown that the algorithm can deal with channel and clock artifacts, interferences, and thermal noise. Practically this has been shown via simulations only for varying thermal noise levels. Future work is to attempt various channel and clock artifacts via simulations. Also, careful hyperparameter tuning was needed to estimate the pitch. This hand-crafting approach could be replaced by a more robust algorithm. Deep learning approaches as applied to audio pitch estimation could be explored. Also, there are limitations in multi-pitch estimation, currently more than a few pitches cannot be estimated. This could be improved by using a longer duration emanation capture. Concerning data collection and HW experiments, we could attempt data collection from a wider set of HWs and also do repeatable captures. Further, emanation needs to be studied concerning different software activities on a HW. Also, intentional clock artifacts could be introduced in the receive SDR by having clock offset and performance tested. Further, data collection in an RF environment outside the shielded room needs to be studied. Further, overt signals could be introduced and algorithm performance checked in their presence practically.

2.5 Conclusion

A profiling-free HW agnostic technique is presented to detect RF emanations. Harmonics from leakages of clock signals is identified as a generic signature symptomatic of emanations. A model for emanations as harmonics modulated by random frequency shifts is used. The important preprocessing helps unmodulate the harmonics, remove HW artifacts, and retain the harmonic structure. Thorough mathematical derivations highlight the performance of the algorithm theoretically. Derivations are shown for single-harmonic, and multi-harmonic cases with intentionally transmitted signals.

Algorithm performance is shown on IQ data collected in a shielded room. Emanations detected across the 1 GHz bandwidth are shown as emanation pattern plots of detected pitch frequencies vs IQ capture frequency. Emanation patterns detected the anomalous activity of damaged electronic peripherals and illegal data transfer. Damaged electronic peripherals are emulated by exposing cables of a mouse and keyboard, and data transfer is emulated by active data transfer between an SD card and a pen drive with a laptop. Emanation patterns for both use cases showed different emanation patterns compared to the baseline of an idle laptop. Thus we have shown HW-agnostic anomalous activity detection using emanations.

2.6 Ackowledgements

The text of this chapter is in the full reprint of the material as it may appear in V. Sathyanarayanan, P. Gerstoft, ”Anomalous activity detection using RF emanations”, Submitted to IEEE transactions on Electromagnetic compatibility, 2024. The dissertation author was the primary researcher and author of this chapter. The co-authors listed in these publications directed and supervised the research.

2.7 References

[1] B. Yilmaz, E. Ugurlu, and M. Prvulovic, “Detecting Cellphone Camera Status at Distance by Exploiting Electromagnetic Emanations,” in *IEEE Mil. Commun. Conf.*, pp. 1–6, 2019.

[2] R. Spreitzer, V. Moonsamy, and T. Korak, “Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices,” *IEEE Commun. Surv. Tuts.*, vol. 20, no. 1, pp. 465–488, 2018.

[3] NSA, “NACSIM 5000: TEMPEST Fundamentals..” https://cryptome.org/nacsim-5000.zip. Partially declassified document.

[4] R. Spolaor, L. Abudahi, and V. Moonsamy, “No Free Charge Theorem: A Covert Channel via USB Charging Cable on Mobile Devices,” in *Appl. Crypto. Netw. Secur.*, pp. 83–102, 2017.

[5] S. Anand and N.Saxena, “Keyboard Emanations in Remote Voice Calls: Password Leakage and Noise (Less) Masking Defenses,” in *ACM Conf. Data Appl. Secur. Privacy*, pp. 103–110, 2018.

[6] M. Guri, B. Zadov, and E. Atias, “LED-it-GO: Leaking (a lot of) data from air-gapped computers via the (small) hard drive LED,” in *Detect. Intrusions Malware Vulnerability Assess.*, pp. 161–184, 2017.

[7] M. Dey, A. Nazari, and A. Zajic, “EMPROF: Memory Profiling Via EM-Emanation in IoT and Hand-Held Devices,” in *IEEE Int. Symp. Microarchit.*, pp. 881–893, 2018.

[8] A. Nazari, N. Sehatbakhsh, and M. Alam, “EDDIE: EM-based detection of deviations in program execution,” in *IEEE Int. Symp. Comput. Archit.*, pp. 333–346, 2017.

[9] M. Bari, M. Chowdhury, and B. Chatterjee, “Detection of Rogue Devices using Unintended Near and Far-field Emanations with Spectral and Temporal Signatures,” in *IEEE Int. Microw. Symp.*, pp. 591–594, 2022.

[10] M. Bari, M. Chowdhury, and S. Sen, “Is broken cable breaking your security?,” in *IEEE Int. Symp. Circuits and Syst.*, pp. 1–5, 2023.

[11] J. Feng, T. Zhao, and S. Sarkar, “Fingerprinting IoT Devices Using Latent Physical Side-Channels,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, no. 2, pp. 1–26, 2023.

[12] P. Socha, V. Miškovský, and M. Novotný, “A Comprehensive Survey on the Non-Invasive Passive Side-Channel Analysis,” *Sensors*, vol. 22, pp. 1–37, 2022.

[13] O. Sosa, Z. Dyka, and I. Kabin, “Simulation of Electromagnetic Emanation of Cryptographic ICs: Tools, Methods, Problems,” in *IEEE East West Des. Test Symp.*, pp. 1–5, 2021.

[14] V. Sathyanarayanan, P. Gerstoft, and A. El-Gamal, "RML22: Realistic Dataset Generation for Wireless Modulation Classification," *IEEE Trans. Wireless Commun.*, pp. 1–12, 2023. doi: 10.1109/TWC.2023.3254490.

[15] H. Xia, K. Alshathri, and V. Lawrence, "Cellular Signal Identification Using Convolutional Neural Networks: AWGN and Rayleigh Fading Channels," in *IEEE Int. Symp. Dyn. Spectr. Access Netw.*, pp. 1–5, 2019.

[16] D. Agrawal, B. Archambeault, R. Rao, and P. Rohatgi, "The EM Side-Channel(s)," in *Crypto. Hardw. Embed. Syst.*, pp. 29–45, 2003.

[17] M. Prvulovic, A. Zajić, and R. Callan, "A Method for Finding Frequency-Modulated and Amplitude-Modulated Electromagnetic Emanations in Computer Systems," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 1, pp. 34–42, 2017.

[18] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," in *USENIX Secur. Symp.*, pp. 1–16, 2009.

[19] O. A. Ibrahim, S. Sciancalepore, G. Oligeri, and R. D. Pietro, "Magneto: Fingerprinting usb flash drives via unintentional magnetic emissions," *ACM Trans. Embed. Comput. Syst.*, vol. 20, no. 1, p. 26, 2020.

[20] C. Maher and J. Beauchamp, "Fundamental frequency estimation of musical signals using a two-way mismatch procedure," *J. Acoust. Soc. Am.*, vol. 95, no. 4, pp. 2254–2263, 1994.

[21] L. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 25, no. 1, pp. 24–33, 1977.

[22] E. Azarov, M. Vashkevich, and A. Petrovsky, "Instantaneous pitch estimation based on RAPT framework," in *Proc. Eur. Signal Process. Conf.*, pp. 2787–2791, 2012.

[23] M. Ross, H. Shaffer, and A. Cohen, "Average magnitude difference function pitch extractor," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 22, no. 5, pp. 353–362, 1974.

[24] S. Abeysekera, "Multiple pitch estimation of polyphonic audio signals in a frequency-lag domain using the bispectrum," in *IEEE Int. Symp. Circuits Syst.*, pp. 469–472, 2004.

[25] M. Christensen and S. Jensen, "Variable order harmonic sinusoidal parameter estimation for speech and audio signals," in *Proc. Asilomar Conf. Signals Syst. Comput.*, pp. 1126–1130, 2006.

[26] J. Tabrikian, S. Dubnov, and Y. Dickalov, "Maximum a-posteriori probability pitch tracking in noisy environments using harmonic model," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 1, pp. 76–87, 2004.

[27] M. Christensen and A. Jakobsson, *Multi-pitch estimation.* Springer, 2009.

[28] A. Nehorai and B. Porat, "Adaptive comb filtering for harmonic signal enhancement," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, no. 5, pp. 1124–1138, 1986.

[29] M. Christensen, A. Jakobsson, and S. Jensen, “Joint high-resolution fundamental frequency and order estimation,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 5, pp. 1635–1644, 2007.

[30] Y. Wu, L. Amir, and J. Jensen, “Joint Pitch and DOA Estimation Using the ESPRIT Method,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 23, no. 1, pp. 32–45, 2015.

[31] A. Oppenheim, A. Willsky, and S. Nawab, *Signals and Systems*. Prentice Hall Inc., 1996.

[32] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice Hall, 2001.

[33] R. Callan, A. Zajić, and M. Prvulovic, “FASE: Finding amplitude-modulated side-channel emanations,” in *Int. Symp. Comput. Archit.*, pp. 592–603, 2015.

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

[35] M. Rice, *Digital Communications: A Discrete-Time Approach*. Pearson Education Inc, 2012.

[36] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 27, no. 2, pp. 113–120, 1979.

[37] Y. Ephraim, D. Malah, and B. Juang, “On the application of hidden Markov models for enhancing noisy speech,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 12, pp. 1846–1856, 1989.

[38] Y. Ephraim and H. V. Trees, “A signal subspace approach for speech enhancement,” *IEEE Trans. Speech Audio Process.*, vol. 3, no. 4, pp. 251–266, 1995.

[39] J. Hansen and M. Clements, “Constrained iterative speech enhancement with application to speech recognition,” *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 795–805, 1991.

[40] L. Rabiner, “On the use of autocorrelation analysis for pitch detection,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 25, no. 1, pp. 24–33, 1977.

[41] P. Welch, “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, 1967.

[42] A. Oppenheim, R. Schafer, and J. Buck, *Discrete-Time Signal Processing*. Prentice Hall Inc., 1998.

[43] J. Kaiser and R. Schafer, “On the use of the I0-sinh window for spectrum analysis,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 28, no. 1, pp. 105–107, 1980.

[44] C. Yang, Z. He, and W. Yu, “Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis,” *BMC Bioinf.*, vol. 10, no. 4, pp. 1–4, 2009.

[45] P. Du, W. Kibbe, and S. Lin, “Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching,” *Bioinf.*, vol. 22, no. 17, pp. 2059–2065, 2006.

[46] K. Coombes, H. A. Fritsche, and C. H. Clarke, “Quality control and peak finding for proteomics data collected from nipple aspirate fluid by surface-enhanced laser desorption and ionization,” *Clin. chem.*, vol. 49, no. 10, pp. 1615–1623, 2003.

[47] S. Mukherjee, “Median-based noise floor tracker (MNFT): Robust estimation of noise floor drifts in interferometric data,” *Clas. Quantum Gravity*, vol. 20, no. 17, pp. S925–S937, 2003.

[48] M. Llobera, “Building past landscape perception with GIS: Understanding topographic prominence,” *J. Archaeol. Sci.*, vol. 28, no. 9, pp. 1005–1014, 2001.

[49] B. Kashyap, M. Horne, and P. Pathirana, “Automated topographic prominence based quantitative assessment of speech timing in Cerebellar Ataxia,” *Biomedical Signal Processing and Control*, vol. 57, no. 101759, 2020.

[50] Y. Song and N. Madhu, “Improved CEM for Speech Harmonic Enhancement in Single Channel Noise Suppression,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 30, pp. 2492–2503, 2022.

[51] J. O. Smith and X. Serra, “PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation,” in *Int. Comput. music Assoc.*, 1987.

[52] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

Chapter 3

RML22: Realistic Dataset Generation for Wireless Modulation Classification

Application of Deep learning (DL) to modulation classification has shown significant performance improvements. The focus has been model centric, where newer architectures are attempted on benchmark dataset RADIOML.2016.10A (RML16). RML16 is a high impact effort that laid the foundation for generating a synthetic dataset for applying DL models to wireless problems. This encouraged development of newer architectures to RML16. We use a data centric DL approach where focus moves from model architectures to data quality. RML16 has shortcomings such as errors and ad-hoc choices of parameters. We build upon RML16 and provide realistic and correct methodology of generating dataset. A new benchmark dataset RML22 is generated. Going forward, we envision researchers to improve model quality on RML22. We attempt to improve data quality by studying the impact of information sources. Further, the choices of artifacts and signal model parameterization are analyzed carefully. The Python source code used to generate RML22 is shared to enable researchers to further improve dataset quality.

3.1 Introduction

Wireless spectrum awareness is essential in enabling optimal spectral usage and detecting anomalous signals. Spectrum is a critical resource due to increasing number of devices and data rate requirements. Developing intelligence in devices enables optimal spectral usage and thereby meet increasing spectral demands. It is also of interest for cellular carriers and defense organizations to police the spectrum. This helps detect illegal and malicious spectrum usage, thereby ensuring robust communication link for critical applications. Modulation classification is an important cog of spectrum awareness. Traditional approaches to modulation classification are broadly classified as likelihood based and feature based [1]. They typically work well for only a small subset of modulation types, channel and hardware (HW) artifacts.

DL as a field grew due to significant improvements in big data [2], algorithms [3, 4], computational capabilities [5] and open source software platforms [6, 7]. In the last few years, researchers have borrowed tools from DL and applied them to the problem of modulation classification, demonstrating tremendous potential for universal success across a wide range of modulation types and wireless technologies. The focus, however, has been model centric. Numerous architectures [8, 9, 10, 11, 12, 13] such as CNN, recurrent neural networks, generative adversarial networks and auto-encoders have been attempted on benchmark datasets RADIOML.2016.10A (RML16) [8] and RADIOML.2018.01A (RML18) [10].

It is important that the model architecture and dataset quality improve hand-in-hand for best deployment performance. We use a data centric [14, 15] approach where the focus moves from modeling to data quality. Here, the models are fixed, and efforts are taken to improve data quality. It is expected that the deployment performance improves with improved data quality.

In contrast to traditional estimation theoretic methods, DL models learn from data. Their performance is as good as the data. A quality real dataset involves captures across a wide range of channel conditions, and HW artifacts. Obtaining such a real dataset is expensive. Wireless is a mature field and accurate signal and artifact models exist. We leverage upon this, to provide a framework towards generating a synthetic dataset for modulation classification.

A high impact effort to successfully leverage accurate wireless models to generate synthetic dataset is in [16], which laid the foundation for generating a synthetic dataset for applying DL models to solve wireless problems. The benchmark datasets for modulation classification RML16 and RML18, are generated using this methodology. GNU radio [17] Python libraries are used to simulate the datasets and RML16 simulation software is available in [18]. In RML18, the methodology of RML16 is expanded to include more modulation types and a wider set of channel conditions. In our work, we focus on RML16, since it is the main prior work that has focused on providing a framework for generating synthetic dataset. The software for RML18 is not available and therefore not included for analysis.

We build upon [18] and provide a realistic methodology of generating dataset. The errors in RML16 are identified, corrected, and a new benchmark dataset is generated referred to as RML22. Eventual goal is to train a model on a comprehensive synthetic dataset and deploy it with excellent performance in the real world. We have taken a first step towards building such a dataset in RML22. Future iteration is to compare and improve the performance of model trained on RML22 and tested on real collect. The software used to generate RML22 is shared, for researchers to build upon the code to further improve dataset quality. The mathematical details behind each block in dataset generation is presented.

The impact of artifacts and information sources are studied. Performance impact of different information sources are studied. The choices of artifacts and signal model parameterization are analyzed carefully. The results provide guidelines on approaches to improving dataset quality. It is intended that continuous improvement in quality leads to building a complete dataset, on which models can be trained and deployed real time. The simulation of blocks used in dataset generation is representative of a generic wireless system. The dataset generation methodology can be extended to any wireless communications problem that can be posed as a data-driven learning problem such as classification of technology [19], modality, modulation [8], modulation and coding scheme [20], coarse signal-to-noise-ratio (SNR) estimation [21] and coarse center frequency offset (CFO) estimation [22].

The major contributions of this paper are as follows:

- A benchmark dataset RML22 is provided, generated after addressing shortcomings of

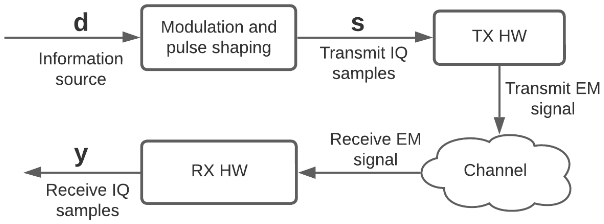


Figure 3.1. Wireless system block diagram.

RML16. We showcase that model trained on RML22 outperforms RML16 by 23% when tested on RML22.

- RML22 generation software is shared to enable further improve dataset quality.
- Impact of information sources is studied. DL model trained and tested on direct English text outperforms that of randomly generated data by 12% due to DL model learning the intrinsic structure of English language.
- A careful mathematical treatment is provided to help reduce errors in future progression of dataset generation.

A quality dataset needs accurate models, realistic simulation and parameterization of models. In Sec. 3.2, we present details of models used and simulation approach with appropriate justifications. In Sec. 3.3, we present details on realistic parameterization of models used in simulation along with simulation call flow. In Sec. 3.4, we demonstrate shortcomings in RML16 and solutions that are incorporated in our dataset RML22.

3.2 System model

We provide the details of signal model, HW and channel artifacts below in sections 3.2.2, 3.2.4. Most literature on DL applied to modulation classification do not investigate the underlying mathematics for dataset generation and focus only on DL architecture. A careful mathematical exposition helps reduce errors in future progression of dataset generation. Further, it helps non-wireless researchers using new DL architectures to modulation classification datasets. The shortcomings in RML16 dataset in Sec. 3.4 were identified through such effort.

A typical wireless system block diagram is in Fig. 3.1. An information source d is modulated into a symbol space and shaped to generate a digital transmit sequence s. The transmit (TX) HW transmits the digital sequence into a channel to a receive (RX) HW, which outputs the raw received IQ samples y. The intention is to learn the underlying signature of each modulation type and correctly classify it. The TX signal s undergoes HW and channel imperfections. In this work, blind modulation classification [23] is performed where no prior information about the signal or system is assumed.

The end goal is to generate a complete dataset through which models can be trained and deployed to process over the air (OTA) signals. OTA signals could belong to modalities such as single or multi-carrier, single input single output (SISO) or multiple input multiple output (MIMO), spread spectrum, etc. This goal entails considering all modalities with accurate models for channel and HW artifacts. The first step towards building a complete dataset is solving the simpler case of single carrier and SISO correctly. Blind modulation classification even for this simpler case is a hard problem. Channel modeling is very hard and there is a trade-off between accuracy and complexity. In this work, a variation of sum of sinusoids is used for the channel model [24, 25, 26]. Sophisticated and more accurate channel models such as QUADRIGA [27] and NYUSIM [28] exist. We see models as an approximation of a real system that are computationally tractable. The final goal is to get rid of models and learn more accurate representations directly from real data. Simpler choice of channel models also help reproducibility and ease of adoption of dataset generation methodology by other researchers.

3.2.1 Wireless signal representation

Wireless signals are transmitted on orthogonal carrier waves, whose magnitudes are referred to as In phase and Quadrature phase (IQ). An IQ sample s is represented as real numbers

Table 3.1. Abbreviations used and their expansions.

Abbreviation	**Expansion**	**Abbreviation**	**Expansion**
ADC	Analog to Digital Converter	AWGN	Additive White Gaussian Noise
CFO	Center Frequency Offset	CNN	Convolutional Neural Network
DAC	Digital to Analog Converter	DL	Deep Learning
ETU	Extended Typical Urban model	IQ	In phase and Quadrature phase
LO	Local Oscillator	LOS	Line Of Sight
LTE	Long Term Evolution	OTA	Over the air
PA	Power Amplifier	PLL	Phase Locked Loop
RRC	Root Raised Cosine	RX	Receive
ReLU	Rectified Linear Unit activation	SNR	Signal to Noise Ratio
SRO	Sample Rate Offset	sps	Samples Per Symbol
TX	Transmit	USRP	Universal Software Radio Peripheral
XO	Crystal Oscillator	BPSK	Binary Phase Shift Keying
QPSK	Quadrature Phase Shift Keying	8PSK	8 Phase Shift Keying
16QAM	16 Quadrature and Amplitude Modulation	64QAM	64 Quadrature and Amplitude Modulation
PAM4	Pulse Amplitude Modulation 4	WBFM	Wide Band Frequency Modulation
CPFSK	Continuous Phase Frequency Shift Keying	GFSK	Gaussian Frequency Shift Keying
AM-DSB	Amplitude Modulation Dual Side Band		

s_{I} and s_{Q}.

$$s[n] = s_{\mathrm{I}}[n] + js_{\mathrm{Q}}[n], \quad s_{\mathrm{p}}[n] = \Re(s[n]\mathrm{e}^{j2\pi f_c n T_{\mathrm{s}}}), \tag{3.1}$$

where s is baseband signal, s_p passband signal with center frequency f_c, s_{I} and s_{Q} are in phase and quadrature phase parts of an IQ sample, T_s is the sampling interval. A sequence of received IQ samples of specified length is passed as input to a DL model. This sequence is referred to as a frame.

3.2.2 Modulation

The modulation types used are BPSK, QPSK, 8PSK, 16QAM, 64QAM, PAM4, WBFM, CPFSK, AM-DSB, and GFSK. These are types found in popular wireless applications and consistent with prior research [8, 29, 30]. WBFM and AM-DSB are analog modulation types whose information source is analog, the rest are digital. The equations governing the simulation of modulation and pulse shaping are presented below.

Modulation alters one of three characteristics of a carrier signal, namely amplitude, frequency, or phase to embed an information source. The analog signal is sampled and represented digitally such as $s[n] = s(t)|_{t=nT_s}$, $n = [1....N]$.

$$s[n] = a[n]\cos\left(2\pi f[n]nT_s + \theta[n]\right), \tag{3.2}$$

where a is the amplitude, f frequency, θ phase of the signal, s modulated symbol, n sample index. One or more of the three characteristics of an electromagnetic wave a, f, θ are varied to generate a requisite signal. Fading effects in wireless communication occur at passband frequencies. However, simulation at the passband is computationally inefficient since center frequency is significantly higher than signal bandwidth and sampling instants shorter. Therefore, a baseband approach is used that is mathematically equivalent and computationally efficient. The equations provided in this work follow digital baseband representation consistent with the simulation.

The equations illustrating the modulation types used are in (3.3) to (3.5). Frequency modulations CPFSK, GFSK, and WBFM [31] are:

$$s_{\text{CPFSK}}[n] = \exp\left(jK_m\pi d[n]\right), \tag{3.3a}$$

$$s_{\text{GFSK}}[n] = \exp\left(jK_m\pi\sum_k d[k]g[n-k]\right) \tag{3.3b}$$

$$s_{\text{WBFM}}[n] = \exp\left(j2\pi\frac{f_d}{f_{out}}\widetilde{d}[n]\right), \tag{3.3c}$$

$$g_{\text{GFSK}}[n] = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\left(\frac{nT_s}{\sqrt{2}\sigma}\right)^2\right], \quad \sigma = \frac{\sqrt{\ln 2}}{2\pi\alpha_g} \tag{3.3d}$$

where $d \in \pm 1$ is the digital information source, $\widetilde{d} \in [-1, +1]$ is the analog information source, s represent modulated symbols, K_m is the modulation index, $g(.)$ is the Gaussian filter, α_g is the roll-off factor for Gaussian filter used in GFSK, f_d is the maximum frequency deviation, f_{out} is the analog modulation sample rate. CPFSK is a frequency modulation technique with continuity of phase during transition between symbols. GFSK is a frequency modulation type with a Gaussian filter applied for pulse shaping. The two amplitude modulation types are PAM4 and AM-DSB, see (3.4). The modulation order M is equal to 4 for PAM4.

$$s_{\text{AMDSB}}[n] = \widetilde{d}[n], \;\; \widetilde{d} \in [-1, +1] \tag{3.4a}$$

$$s_{\text{PAM}}[n] = d_{\text{PAM}}[n], \;\; d_{\text{PAM}} \in \{1/M ... M/M\}, \tag{3.4b}$$

The modulations BPSK, QPSK, 8PSK are types of M-PSK see (3.5a). The M-QAM modulation can be thought of as amplitude modulation taking place in quadrature, see (3.5b). For more details on M-PSK and M-QAM see [32, pg. 175].

$$s_{\text{MPSK}}[n] = \exp(j2\pi \frac{i[n]-1}{M}), \;\; i \in \{1, 2, ...M\}, \tag{3.5a}$$

$$s_{\text{MQAM}} = A_I[n] + jA_q[n], \tag{3.5b}$$

$$A_I, A_q \in [-(\sqrt{M}-1)d, -(\sqrt{M}-3)d, ..., +(\sqrt{M}-3)d, +(\sqrt{M}-1)d], \tag{3.5c}$$

where M is the modulation order in M-PSK or M-QAM, A_I and A_q are in-phase and quadrature phase components, d is the signal distance. All modulation types are up-sampled using interpolation filters. A root-raised-cosine (RRC) pulse shaping filter (3.6) [32, pg. 140], is applied via a convolution operation to PAM, PSK and QAM modulation types. RRC filter optimizes the trade-off between maximal symbol rate and low inter-symbol-interference.

$$g_{\text{RRC}}(t) = \frac{1}{T_s} \text{sinc}\,(t/T_s) \frac{\cos(\pi \alpha_r t/T_s)}{1-(2\alpha_r t/T_s)^2}, \tag{3.6}$$

where α_r is the roll-off factor for RRC filter, sinc(t) is defined as $\sin(\pi t)/\pi t$.

3.2.3 Transmit and receive HW architecture

A TX and RX HW based on a homodyne architecture [33, pg. 337] is illustrated in Fig. 3.2. The HW artifacts models presented in Sec. 3.2.4 follow this architecture. The digital IQ stream is converted into an analog signal by the digital-to-analog converter (DAC). Further, it

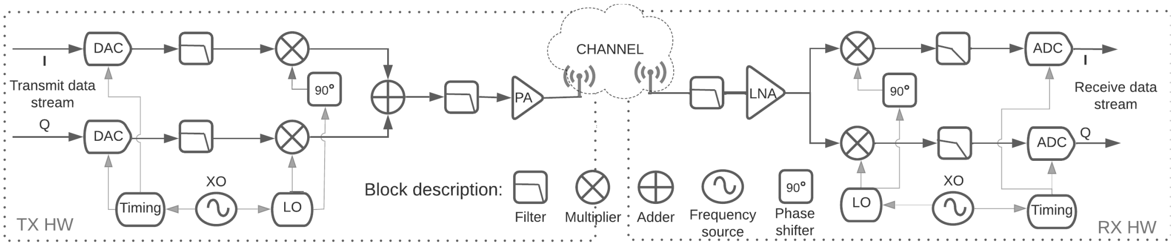


Figure 3.2. A homodyne architecture of the TX and RX HW. The HW model used follows the architecture presented.

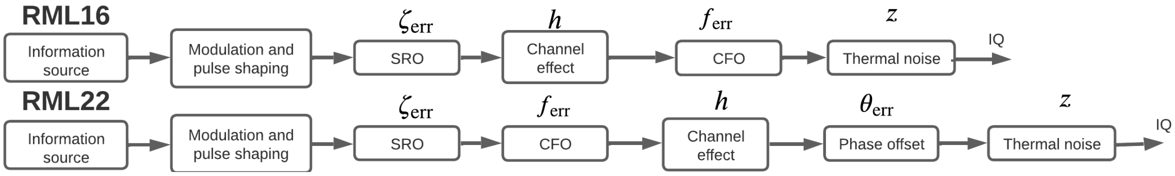


Figure 3.3. Dataset simulation block diagram for RML16 and RML22. Note that RML22 dataset follows the block diagram in the sequence presented for simulation. RML16 dataset does not include the phase offset block. The sequence it follows is also different. All blocks in RML16 has errors that are analyzed and corrected in Sec. 3.4.

is shifted from baseband to a carrier frequency, boosted by a power amplifier and transmitted OTA via antennae. This signal undergoes channel effects also referred to as radio propagation effects, before reaching the RX. At RX, the antenna senses the electromagnetic energy, converts into an electric voltage. This electrical signal is amplified by a low-noise amplifier, shifted to baseband from a higher center frequency. It is then converted into a digital IQ stream by the analog-to-digital converter (ADC). A clock crystal is an important part of a wireless system. It provides tone at requisite frequency via a phase-locked-loop (PLL) for the frequency shifting operations in the TX and RX systems. It also provides reference frequencies for generating sample timing to the ADC and DAC components.

3.2.4 Channel and HW artifacts

The problem of blind modulation classification is solved using DL techniques. To simulate dataset for the DL approach, a simplified yet accurate model is presented. The received

IQ signal y for a wireless signal affected by channel and HW effects is [34, pg. 16]:

$$y[n] = e^{j2\pi f_{\text{err}} n T_s + \theta_{\text{err}}} \sum_{l} h[l] s[n - l - \zeta_{\text{err}}] + z[n], \tag{3.7}$$

where h is the channel impulse response, s is the modulated symbol up-sampled and pulse shaped by an RRC filter, T_s sampling rate, f_{err} frequency error, θ_{err} phase error, ζ_{err} timing error, z Additive White Gaussian Noise (AWGN). The y, s, h are the digital baseband equivalent terms.

The random variables $\zeta_{\text{err}}, \theta_{\text{err}}, f_{\text{err}}, z, h$ represent the set of artifacts that is imposed upon the clean transmit IQ stream, see block diagram in Fig. 3.3. Several HW artifacts that are not considered here such as IQ imbalance, local oscillator (LO) leakage causing spur at DC, power amplifier (PA) non-linearities etc. The model (3.7) is a simplified, yet accurate representation of the artifacts considered.

Thermal Noise model

The thermal noise is modeled as follows in RML22.

$$SNR = 10\log_{10}\left[E(|s|^2)/E(|z|^2)\right], \tag{3.8a}$$

$$y[n] = s[n] + z[n], \tag{3.8b}$$

$$z[n] = z_I[n] + jz_Q[n] \sim \mathscr{CN}(0, \sigma_z^2), \forall n, \tag{3.8c}$$

$$z_I[n], z_Q[n] \sim \mathscr{N}(0, \sigma_z^2/2), \forall n, \tag{3.8d}$$

$$E(z(t)z^*(t+\tau)) = \delta(\tau)\sigma_z^2, \; E(z_I z_Q) = E(z_I)E(z_Q) \tag{3.8e}$$

where $z[n]$ is the nth sample of complex thermal noise, z_I and z_Q are the real and imaginary parts, σ_z is the standard deviation of thermal noise.

Thermal noise z is modeled as zero mean AWGN, see (3.8c) and (3.8e), with the real and imaginary parts independently zero-mean AWGN random process[35, p 29], see (3.8d). SNR is calculated using (3.8a). The dominant source of thermal noise is assumed to be at the receiver, thus the additive assumption. In simulations, thermal noise z is assumed ergodic, and expectation calculated averaging samples over time. Although thermal noise simulated in baseband is a filtered and sampled version, the properties mentioned above are assumed to hold. The GNU radio block simulating thermal noise is in [36].

Phase, Frequency and Timing error

The CFO and SRO in RML16 are negligible, see Sec. 3.4.1. A new model for simulating clock effects is presented, following the homodyne architecture in Fig. 3.2. The errors in the clock in TX and RX HW manifest as CFO, SRO and phase offset. The cumulative CFO effects are:

$$f_{\mathrm{LO}}^{TX}[n] = (\hat{f}_{XO} + \Delta f_{XO}^{TX}[n])L_{LO}, \tag{3.9a}$$

$$f_{\mathrm{LO}}^{RX}[n] = (\hat{f}_{XO} + \Delta f_{XO}^{RX}[n])L_{LO}, \tag{3.9b}$$

$$f_{\mathrm{err}}^{\mathrm{XO}}[n] = \Delta f_{XO}^{TX}[n] - \Delta f_{XO}^{RX}[n], \tag{3.9c}$$

$$f_{\mathrm{err}}^{\mathrm{LO}}[n] = f_{\mathrm{LO}}^{TX}[n] - f_{\mathrm{LO}}^{RX}[n] = f_{\mathrm{err}}^{\mathrm{XO}}[n]L_{LO}, \tag{3.9d}$$

$$s'[n] = s[n]e^{j2\pi f_{\mathrm{err}}^{\mathrm{LO}} nT_s}, \tag{3.9e}$$

where $\Delta f_{XO}^{TX}[n], \Delta f_{XO}^{RX}[n]$ are crystal oscillator (XO) errors in TX and RX, $\hat{f}_{XO}$ is the needed reference tone frequency from XO, L_{LO} is the scaling factor to shift tone from XO frequency to center frequency, f_{LO}^{TX} and f_{LO}^{RX} are the LO signals in TX and RX, $f_{\mathrm{err}}^{\mathrm{XO}}$ and $f_{\mathrm{err}}^{\mathrm{LO}}$ are the XO and LO frequency errors in combined TX-RX system, s is the clean baseband signal, s' is the baseband signal with CFO.

The XO crystal of an RF system is prone to errors that manifest as CFO and SRO. Frequency source from XO feed the LO via a phase locked loop (PLL). This in turn is used for providing center frequencies for up-conversion and down-conversion in TX and RX systems, respectively. XO frequency errors cause a mismatch in the LO frequencies of TX and RX that manifests as CFO f_{err}, see (3.9) [37, pg. 360]. XO crystal is also used to generate accurate time ticks for the digital-to-analog-converter and analog-to-digital-converter via a timing PLL. XO frequency errors cause a mismatch in the sampling instants of DAC and ADC in TX and RX systems thereby causing SRO ζ_{err}, see (3.11) [37, pg. 436]. XO is assumed the only common source of errors for both CFO and SRO. In practice, the PLL leading into LO, DAC, ADC etc., also contribute to minor errors.

$$\hat{t} = 1/(\hat{f}_{\text{XO}} L_t) = 1/f_{\text{CR}}, \tag{3.10a}$$

$$t_{\text{DAC}} = \frac{1}{(\hat{f}_{\text{XO}} + \Delta f^{\text{TX}}_{\text{XO}}[n]) L_t}, \tag{3.10b}$$

$$t_{\text{ADC}} = \frac{1}{(\hat{f}_{\text{XO}} + \Delta f^{\text{RX}}_{\text{XO}}[n]) L_t}, \tag{3.10c}$$

$$\zeta_{\text{TX}}[n] = t_{\text{DAC}} - \hat{t} \approx \Delta f^{\text{TX}}_{\text{XO}}[n] L_t / f^2_{\text{CR}}, \tag{3.11a}$$

$$\zeta_{\text{RX}}[n] = t_{\text{ADC}} - \hat{t} \approx \Delta f^{\text{RX}}_{\text{XO}}[n] L_t / f^2_{\text{CR}}, \tag{3.11b}$$

$$\zeta_{err}[n] = \sum_{i=1}^{n} (\zeta_{\text{TX}}[i] - \zeta_{\text{RX}}[i]) = \sum_{i=1}^{n} \frac{f^{\text{XO}}_{\text{err}}[i] L_t}{f^2_{\text{CR}}}, \tag{3.11c}$$

where $\hat{t}$ is the requisite time tick interval to DAC and ADC, f_{CR} is the clock rate for DAC and ADC (equal to the analog bandwidth), L_t is the scaling factor in timing PLL, t_{DAC} and t_{ADC} are real time tick intervals going to DAC and ADC, ζ_{TX} and ζ_{RX} are the SRO in TX and RX, ζ_{err} is the cumulative SRO in combined TX-RX system. The scaling factor L_t in timing PLL is assumed the same for both DAC and ADC.

Phase changes occur due to factors such as frequency errors, Doppler, sampling errors, distance traveled and non-synchronous TX and RX LO. Artifacts as f_{err}, ζ_{err}, $h(.)$ capture these phase changes, except non-synchronous TX and RX LO. This is captured by θ_{err}:

$$x_{\text{LO}}^{\text{TX}}[n] = e^{j2\pi f_{\text{LO}}^{\text{TX}} nT_s + \theta_{err}^{\text{TX}}}, \tag{3.12a}$$

$$x_{\text{LO}}^{\text{RX}}[n] = e^{j2\pi f_{\text{LO}}^{\text{RX}} nT_s + \theta_{\text{err}}^{\text{RX}}}, \tag{3.12b}$$

$$x_{\text{LO}}^{\text{TX}}(x_{\text{LO}}^{\text{RX}})^* = e^{j2\pi f_{\text{err}}^{\text{LO}} nT_s + (\theta_{\text{err}}^{\text{TX}} - \theta_{\text{err}}^{\text{RX}})}, \tag{3.12c}$$

where $x_{\text{LO}}^{\text{TX}}$ and $x_{\text{LO}}^{\text{RX}}$ are the TX and RX LO signals, $\theta_{\text{err}}^{\text{TX}}$ and $\theta_{\text{err}}^{\text{RX}}$ are the phase errors in the TX and RX respectively, θ_{err} is the net phase error from the combined TX and RX systems. The TX and RX LO signals used to upconvert and downconvert signals in TX and RX respectively. The net effect of TX and RX LO after downconversion at RX is in (3.12c).

The methodology to simulate f_{err}, ζ_{err} and θ_{err} is in (3.13). CFO is a clipped Gaussian process (3.13b). SRO is simulated via re-sampling through interpolation at time instants specified by t' (3.13e). Phase error θ_{err} is from a uniform distribution (3.13f).

$$f_{\text{bias}} \sim U(-f_{\text{max}}, f_{\text{max}}), \tag{3.13a}$$

$$f_{\text{err}}^{\text{XO}}[n] \sim N(f_{\text{bias}}, n\sigma^2), \quad |f_{\text{err}}^{\text{XO}}[n]| \leq f_{\text{max}} \tag{3.13b}$$

$$s'[n] = s[n]e^{j2\pi f_{\text{err}}^{\text{XO}}[n] L_{\text{LO}} nT_s} \tag{3.13c}$$

$$T_s^{\text{DAC}}[n] = n\hat{t}, \;\; T_s^{\text{ADC}}[n] = n\hat{t} + \zeta_{err}[n], \tag{3.13d}$$

$$t'[n] = \frac{f_{\text{CR}}}{f_s} T_s^{\text{ADC}}[n] = \frac{n}{f_s} + \frac{1}{f_s f_{\text{CR}}} \sum_{i=1}^{n} f_{\text{err}}^{\text{XO}}[i] L_t, \tag{3.13e}$$

$$\theta_{\text{err}} \sim U(0, 2\pi), \quad s'[n] = s[n]e^{j2\pi\theta_{\text{err}}}, \tag{3.13f}$$

where f_{bias} is the CFO at the start of frame, σ standard deviation per sample, f_{max} maximum frequency error bound, T_s^{DAC} and T_s^{ADC} sampling instants, t' new sampling instant.

Channel model

Channel effects are classified under small-scale and large-scale fading. Large-scale fading effect is due to path losses as a function of distance, shadowing effects etc., and is considered stationary in time scales of a frame duration. It thus manifests as reduction in signal power at receiver, as captured by thermal noise model in Sec. 3.2.4. Small-scale fading occurs in the distance scale of carrier wavelengths, that cause rapid changes in phase and amplitude of received signal. The channel model considered here represents small-scale fading.

A wireless signal encounters numerous objects from TX to RX. The resultant signal at the RX is the sum of multiple reflected, scattered signal copies with delay. The magnitude of each multipath depends on path loss and material properties of reflector and scatters. The delay depends on path length of the multipath. Modeling the effect along each path via ray tracing needs complete knowledge of the channel, and not feasible. Instead, modeling via an input-output relation is used [35, pg. 26], where y is the output signal, h channel impulse response, x input signal. The passband is represented as:

$$\begin{aligned} y_{\mathrm{p}}(t) &= \int h_{\mathrm{p}}(\tau,t)x_{\mathrm{p}}(t-\tau)\mathrm{d}\tau, \\ &= \sum_i a_{\mathrm{p}}(i,t)x_{\mathrm{p}}(t-\tau(i,t)), \qquad (3.14a) \\ h_{\mathrm{p}}(\lambda,t) &= \sum_i a_{\mathrm{p}}(i,t)\delta(\lambda-\tau(i,t)), \qquad (3.14b) \end{aligned}$$

and baseband represented as:

$$\begin{aligned} y[m] &= \sum_l h[l,m]x[m-l] \qquad (3.15a) \\ &= \sum_l x[m-l]\sum_i a[i,m]\mathrm{sinc}[l-\frac{\tau[i,m]}{T_s}], \qquad (3.15b) \\ h[l,m] &= \sum_i a_{\mathrm{p}}[i,m]e^{-j2\pi f_c\tau[i,m]}\mathrm{sinc}[l-\frac{\tau[i,m]}{T_s}], \qquad (3.15c) \end{aligned}$$

where $a_{\mathrm{p}}(i,t)$ is the passband magnitude response of ith path at time t, $a[i,m]$ baseband magnitude response of ith path at time mT_s, $\tau(i,t)$ is the delay of ith path at time t, $\delta(.)$ is an impulse signal.

The channel $h_p(\lambda,t)$ is represented as a slow, time varying system with memory. Most significant reflected and scattered paths are only considered and represented in the equation with an index i. The channel is assumed to be underspread where the timescale of channel variations is significantly longer than the delay spread of the channel. The continuous time passband representation of the input output relation and channel impulse response is presented in (3.14). The equivalent discrete baseband representation is in (3.15).

The radio propagation channel could be an office space, city downtown etc. To capture the propagation effects, detailed knowledge of the objects in the signal path and large computational resources are needed. An alternate approach is using statistical parametric model representing the channel.

$$f_X(x;\sigma_l) = \frac{2x}{\sigma_l^2}\exp\left\{\frac{-x^2}{\sigma_l^2}\right\}, \tag{3.16a}$$

$$f_X(x;K,\sigma_l) = \frac{x}{\sigma_l^2}\exp\left\{\frac{-(x^2+K^2)}{\sigma_l^2}\right\} I_0\left[\frac{Kx}{\lambda^2}\right], \tag{3.16b}$$

$$R[l,n] = E_m\left\{h^*[l,m]h[l,m+n]\right\}, \tag{3.16c}$$

$$S[v] = \sum_l \sum_n R[l,n]e^{-j2\pi vn}, \tag{3.16d}$$

$$S_{\text{Jakes}}[v] = \frac{1}{\pi f_d\sqrt{1-(v/f_d)^2}}, \; |v| \le f_d = \frac{\mathrm{v}_{\max}}{\lambda}, \tag{3.16e}$$

where f is a probability distribution, x and σ_l are the magnitude and standard deviation of the lth tap or path depending upon whether the channel taps $h[l,m]$ or path magnitudes $a[l,m]$ are modeled, K referred as K-factor is ratio of energies in direct and reflected paths, I_0 is the zeroth order modified Bessel function of the first kind, R is the auto-correlation function of channel impulse response, S is the Doppler spectrum, v Doppler frequency, f_d maximum Doppler frequency, $\mathrm{v}_{\max}$ maximum relative speed between TX and RX, λ carrier frequency wavelength.

Table 3.2. Parameters used in RML16 and RML22 dataset generation.

Description	RML16 Values	RML22 Values
Signal	Samples per symbol = 8, Sample rate = 200 kHz	Samples per symbol = 2, Sample rate = 30 kHz Center frequency = 1 GHz, Clock rate = 100MHz
Modulation	Roll-off factor = 0.35, CPFSK modulation index = 0.5, GFSK BW time product = 0.3, GFSK sensitivity = 0.03, WBFM max. freq. deviation = 75 kHz	Roll-off factor = 0.35, CPFSK modulation index = 0.5, GFSK BW time product = 0.3, GFSK sensitivity = 1.57, WBFM max. freq. deviation = 75 kHz
Dataset	Num. frames per mod. per SNR = 1000, Frame length = 128, Frame duration = 0.64 ms	Num. frames per mod. per SNR = 2000, Frame length = 128, Frame duration = 4.5 ms
Fading	Rician fading model, Filter tap magnitudes = $[0,-0.97,-5.23]$ dB, Filter tap delays = [0, 4.5, 8.5] ns, Num. of taps = 8, Max. freq. dev (Doppler) = 1 Hz, Num. of sinusoids = 8, K-factor = 4	3GPP fading model ETU70, Filter tap magnitudes = $[-1, -1,-1,0,0,0,-3,-5,-7]$ dB, Filter tap delays = [0, .05, .12, .2, .23, .5, 1.6, 2.3, 5] ns, Num. of taps = 8, Max. freq. dev (Doppler) = 70 Hz, Num. of sinusoids = 8
Clock effect	LO and SRO max. freq. deviation: 500 Hz, 500 Hz LO and SRO standard dev. per sample = 10^{-2}, 10^{-2}	XO, LO and SRO max. freq. dev.: 5 Hz, 500 Hz, 50 Hz XO, LO and SRO std dev. per sample = 10^{-4}, 10^{-2}, 10^{-3} XO to LO scaling = 100, XO to clock rate scaling = 10
AWGN	-20 to 20 dB in steps of 2 dB	-20 to 20 dB in steps of 2 dB

In the statistical model, the magnitude of each path is assumed an aggregate of numerous paths of similar delay. By central limit theorem, the real and imaginary components of the magnitudes $a(l,t)$ are zero-mean Gaussian. The amplitude of each path is thus a Rayleigh distribution, see (3.16a). An alternate model used is the Rician distribution, see (3.16b). Due to the independence between paths, the phase is uniformly $[0, 2\pi]$ distributed. The channel tap magnitudes are equivalently modeled as a Rayleigh or Rician distribution.

The time varying nature of the channel is modeled via a tap gain auto-correlation function, see (3.16c). A measure of variance of channel in time for each tap l is conveniently captured by auto-correlation function, upon using the wide-sense stationarity assumption. The tap gain auto-correlation function is averaged across channel taps l and a Fourier transform in time is taken to obtain the Doppler spectrum, see (3.16d). The Doppler spectrum indicates the amount of spread in frequency due to time variation of the channel. Jakes model (3.16e) is commonly used to simulate the Doppler spread due to time varying nature of the channel.

Channel effects are simulated using the power delay profile and maximum Doppler frequency f_d for a specific propagation environment [25, 26]. Power delay profile contains average path gains and delays for a specified number of multipaths. User provides the maximum relative speed v and then $f_d = v/\lambda$, where λ is the center frequency wavelength.

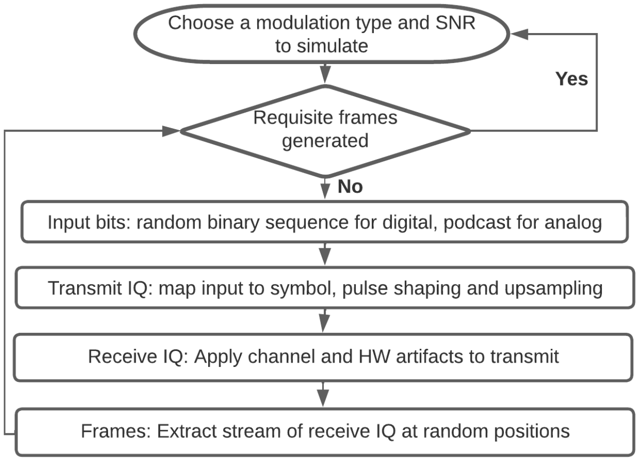

Choose a modulation type and SNR to simulate
Requisite frames generated
Yes
No
Input bits: random binary sequence for digital, podcast for analog
Transmit IQ: map input to symbol, pulse shaping and upsampling
Receive IQ: Apply channel and HW artifacts to transmit
Frames: Extract stream of receive IQ at random positions

Figure 3.4. Call flow of RML22 dataset generation methodology. Software implementation of dataset generation follows this call flow.

3.3 Dataset parameterization

In this section, RML22 dataset parameterization is detailed. The dataset generation parameters for RML22 and RML16 are in Table 3.2, with call flow of dataset simulation in Fig. 3.4. A spectrum sensing system could sense signals of varying signal bandwidths and operate at varying sample rate. Bandwidth is inversely proportional to symbol duration. Symbol duration is chosen as that of long term evolution (LTE) symbol time of 66.67 μs [38], whose bandwidth is 15 kHz. Sample rate is product of bandwidth and sps. In spectrum sensing applications, computational load is a major challenge [39], due to the wide bandwidth sensed. Therefore, samples per symbol (sps) greater than 2 may not be feasible, as used here. The choice of sps is studied in detail in [40] that indicates minimal performance improvements with higher sps. The sample rate for sps=2, is $f_s = 2/66.671 \times 10^{-6} = 30$ kHz. A study of impact of signal bandwidth and sps is in Sec. 4.5. The choice of number of frames and frame length are extensively studied [10], RML22 use the same as RML16. For SNR, performance values flatten around the points of -20 and 20 dB [9]. Therefore, it is not useful to train on data beyond this range. The specific SNR points in this range that are pertinent for training are studied in [41] for RML16. To simulate a fading channel, the inputs should be the maximum Doppler frequency and power delay profile

Table 3.3. Shortcomings in RML16 dataset.

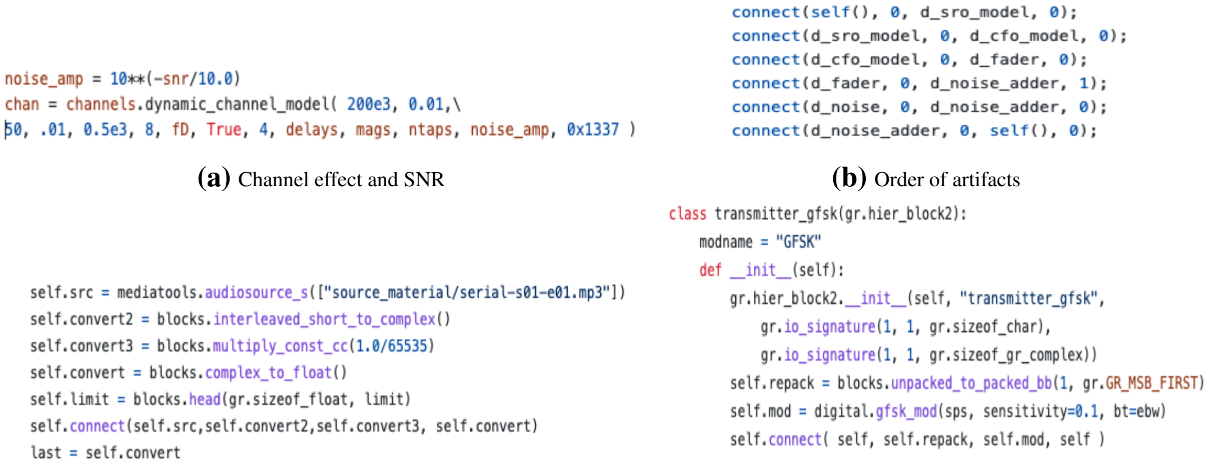
Item	Details
Clock effect	CFO & SRO applied is negligible
Channel effect	Static LOS channel simulated instead of Rician fading
SNR	SNR is from -40 to 40 dB, instead of -20 to 20 dB
Order of artifacts	Incorrect order of artifacts introduces wrong frequency shifts
Information source	Analog modulation types WBFM and DSB-AM use incorrect information source
GFSK modulation	Incorrect modulation index value used

path gains and delays. In this work, a 3rd generation partnership project (3GPP) channel model LTE Extended Typical Urban model 70 (ETU70) [42] is used, this represents maximum 70 Hz Doppler frequency. A study of impact of maximum Doppler frequency is in Sec. 4.5.

Frequency errors are generated for XO based on a homodyne architecture. It is scaled accordingly for LO, ADC and DAC clocks. A clock rate of 100 MHz is chosen based upon the maximum sampling rate of software defined radio USRP N310 [43]. Timing jitters due to imperfect sample instants in DAC and ADC introduces SRO. Therefore, choice of DAC and ADC clock rate impacts the SRO. The center frequency is derived by scaling XO frequency. However, the errors in XO are also scaled accordingly into CFO, which depend on center frequency. In this work, center frequency of 1 GHz is chosen. The frequency error does not change appreciably within a frame. For standard deviation per sample of LO frequency error of 0.01, expected frequency error is 0.34 Hz at the end of frame length of 128 samples, see Sec. 3.4.1 for details. This frequency error is not appreciable, and thus the standard deviation likely does not impact classification performance. The optimal choice of clock error simulation parameters is studied in [44], the impact of XO frequency deviation is in Sec. 4.5.

3.4 RML16 error analysis and correction

In this section, we present the error analysis for RML16 and corresponding corrections that are incorporated in RML22. See summary of errors in Table 3.3. The permanent GitHub link to snapshots of code in Fig. 3.5 is in [24, 45].



```
noise_amp = 10**(-snr/10.0)
chan = channels.dynamic_channel_model( 200e3, 0.01,\
50, .01, 0.5e3, 8, fD, True, 4, delays, mags, ntaps, noise_amp, 0x1337 )
```

(a) Channel effect and SNR

```
connect(self(), 0, d_sro_model, 0);
connect(d_sro_model, 0, d_cfo_model, 0);
connect(d_cfo_model, 0, d_fader, 0);
connect(d_fader, 0, d_noise_adder, 1);
connect(d_noise, 0, d_noise_adder, 0);
connect(d_noise_adder, 0, self(), 0);
```

(b) Order of artifacts

```
self.src = mediatools.audiosource_s(["source_material/serial-s01-e01.mp3"])
self.convert2 = blocks.interleaved_short_to_complex()
self.convert3 = blocks.multiply_const_cc(1.0/65535)
self.convert = blocks.complex_to_float()
self.limit = blocks.head(gr.sizeof_float, limit)
self.connect(self.src,self.convert2,self.convert3, self.convert)
last = self.convert
```

(c) Information Source

```
class transmitter_gfsk(gr.hier_block2):
    modname = "GFSK"
    def __init__(self):
        gr.hier_block2.__init__(self, "transmitter_gfsk",
            gr.io_signature(1, 1, gr.sizeof_char),
            gr.io_signature(1, 1, gr.sizeof_gr_complex))
        self.repack = blocks.unpacked_to_packed_bb(1, gr.GR_MSB_FIRST)
        self.mod = digital.gfsk_mod(sps, sensitivity=0.1, bt=ebw)
        self.connect( self, self.repack, self.mod, self )
```

(d) GFSK modulation

Figure 3.5. Code snapshots highlighting the RML16 errors for channel effect, SNR, order of artifacts, information source, and GFK modulation.

3.4.1 CFO and SRO simulation

The clock effects CFO and SRO are negligible in RML16. Note that phase errors are not included in this dataset.

$$f^{\mathrm{LO}}_{\mathrm{err}}[n] \sim \mathscr{N}(0, \sigma_n), \quad |f^{\mathrm{LO}}_{\mathrm{err}}[n]| \le f_{\max}, \tag{3.17a}$$

$$\mathscr{P}(|f^{\mathrm{LO}}_{\mathrm{err}}| \le 3\sigma_n) = \mathscr{P}(|f^{\mathrm{LO}}_{\mathrm{err}}| \le 3\sqrt{n}\sigma), \tag{3.17b}$$

where standard deviation σ_n at the end of nth sample is $\sqrt{n}\sigma$, $n = [1....N]$. Model used for CFO simulation in [8] is (3.17). For $N = 80 \times 10^3$ and $\sigma = 0.01$, $\mathscr{P}(-3\sigma_n \le f^{\mathrm{LO}}_{\mathrm{err}} \le 3\sigma_n) = \mathscr{P}(-6\,\mathrm{Hz} \le f^{\mathrm{LO}}_{\mathrm{err}} \le +6\,\mathrm{Hz}) = 0.9973$. N is the length of transmit IQ stream. The maximum value of $N = 80 \times 10^3$ used in RML16, is considered in this error analysis. The intended maximum frequency deviation is $f_{\max} = 500\,\mathrm{Hz}$. The applied frequency error $f^{\mathrm{LO}}_{\mathrm{err}}$ is however bound by ±6 Hz with a high probability. The CFO applied is therefore negligible.

This issue also percolates to SRO simulation due to dependency of SRO on $f^{\mathrm{XO}}_{\mathrm{err}}$, see (3.11c). To get $f^{\mathrm{LO}}_{\mathrm{err}}$ to drift to 500 Hz, n $= \left(\frac{500}{3\sigma}\right)^2 = 278$ million samples (3.17b). This involves a few gigabytes of data and is prohibitively large. An alternate model that solves this issue is

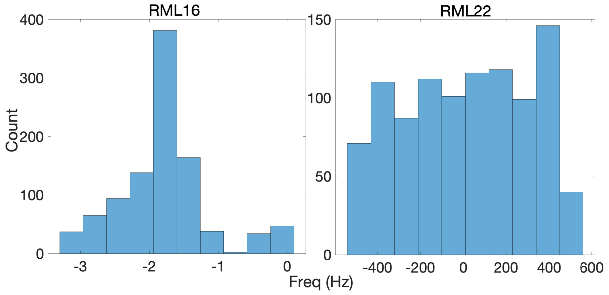


Figure 3.6. Histogram of frequency error estimated from tone passed through CFO model in RML16 and RML22. The frequency estimate is calculated for 1000 frames following respective CFO model.

proposed in (3.13b). The two models are studied as follows. A tone is sent through the two CFO models and frequency error estimated. The average CFO across 1000 frames is taken and histogram computed, see Fig. 3.6. The results indicate that the CFO is simulated correctly in RML22. The results are applicable for SRO artifact.

3.4.2 Channel effect

RML16 attempts simulating a Rician fading channel effect, but a static line of sight (LOS) channel is simulated. A stream of samples affected by channel and HW artifacts is generated. Frames of specified length are carved out at random indices of the stream. The stream of samples is regenerated, and process repeated until required frames are obtained. The noise seed for channel effects erroneously remains same across each stream. Thus, the entire Rician distribution is not sampled. Each stream is of a short duration, less than 1 s. Path gains are correlated within a few seconds, thus a static LOS channel is simulated.

In RML16, each stream has a maximum length $N = 80 \times 10^3$. For a sampling rate $F_s = 200$ kHz, number of samples $N = 80 \times 10^3$ corresponds to capturing data undergoing fading effects for 0.4 s. The frequency deviation input in the data generation is $f_d = 1$ Hz. Based on (3.16e), $v = f_d \lambda = f_d c / f_c = 0.33$ m/s for $f_d = 1$ Hz and center frequency 900 MHz. The user has moved 6 cm in the outdoor fading channel. Path gain changes are almost static in such a short duration of 0.4 s. This is effectively a static LOS channel and not Rician. Code snapshot of channel effect function call illustrating error is in Fig. 3.5a. The solution is changing the random seed for every iteration, thereby the channel effects are sampled from entire Rician pdf.

3.4.3 SNR

In RML16, it is intended to simulate thermal noise in an SNR range of -20 to 20 dB. Due to errors, the actual SNR simulated is -40 to 40 dB. The SNR is defines as

$$SNR = 10\log_{10}\left(\frac{E(|s|^2)}{E(|z|^2)}\right), \quad \sigma_{\mathrm{z}} = 10^{\frac{-\mathrm{SNR}}{20}} \text{ s.t. } E(|s|^2) = 1. \tag{3.18}$$

The thermal noise simulation block [36] use noise standard deviation as input. This standard deviation is $10^{\frac{-\mathrm{SNR}}{20}}$, see (3.18). RML16 uses $10^{\frac{-SNR}{10}}$, causing the error. Code snapshot in Fig. 3.5a highlights the error.

3.4.4 Order of artifacts

Information source is mapped to symbols, upsampled and pulse shaped to generate transmit IQ samples. Artifacts are applied to the IQ samples. The order of application of artifacts should follow the input output response chosen for simulation. For noiseless channels, the effects of incorrect order are:

$$y[n] = e^{j2\pi f_{\mathrm{err}} n T_s} \sum_l s[n - l - \zeta_{\mathrm{err}}] h[l], \tag{3.19a}$$

$$= e^{j2\pi f_{\mathrm{err}} n T_s} \mathscr{F}^{-1}(S(f)H(f)), \tag{3.19b}$$

$$y'[n] = e^{j2\pi f_{\mathrm{err}} n T_s} \sum_l s[n - l - \zeta_{\mathrm{err}}] e^{-j2\pi f_{\mathrm{err}} l T_s} h[l], \tag{3.19c}$$

$$= e^{j2\pi f_{\mathrm{err}} n T_s} \mathscr{F}^{-1}(S(f)H(f + f_{\mathrm{err}})), \tag{3.19d}$$

where y is output from correct order of artifacts, y' is the incorrect order of artifacts as followed in RML16.

The correct order of artifacts is SRO, channel effects, CFO, AWGN, see (3.19a). The

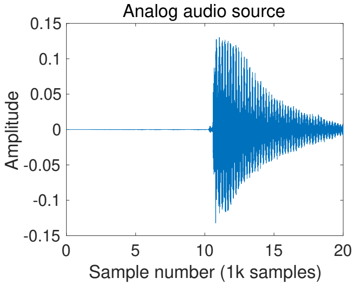


Figure 3.7. Audio recording used as source for analog modulation types AM-DSM and WBFM in RML16. At a standard audio sampling rate of 44.1 kHz, x-axis corresponds to 0.45 s. The first 10k samples is noise as it waits for signal to appear.

incorrect order followed in RML16 is SRO, CFO, channel effects, AWGN [24], see (3.19c). Code snapshot in Fig. 3.5b showcases the incorrect order. Random phase offset is not included in this analysis, since it is not used in RML16. The order of applications of artifacts are not equivalent. There is an additional unintended frequency error contribution f_{err} from the term $H(f + f_{\text{err}})$. The applied frequency error is thus $2f_{\text{err}}$ instead of f_{err}. E.g., if the intended $f_{\max}$ maximum frequency error bound for CFO is 500 Hz, the applied value is 1000 Hz.

3.4.5 Information source

The information source used for analog modulation types in RML16 is essentially noise, due to extracting a near-zero amplitude portion of the audio recording, see Fig. 3.7. All frames in RML16 for analog modulation types are affected. Code snapshot in Fig. 3.5c highlights the error.

Data source used for analog modulation is from a podcast audio file. To generate RML16, frames are extracted from a stream of samples, only the first 10k samples of the audio file is repeatedly used. The audio file with running time of 53 minutes and sampled at 44.1 kHz, has about 140 million samples. The amplitude is just noise for the first 10k samples, as it waits for signal to start. Thus, the analog modulation types (WBFM and AM-DSB) are using noise as information source. The correct method is to randomly choose from all parts of audio file.

3.4.6 GFSK modulation issue

$$\begin{aligned} s_{\text{GFSK}}[n] &= \exp(jK_m\pi\sum_k d[k]g[n-k]) \\ &= \exp(jK_s\sum_k d[k]g[n-k]), K_s = \pi K_m = 2\pi\frac{f_d}{f_s}, \end{aligned} \tag{3.20}$$

where K_m is the modulation index, K_s is the sensitivity index, f_d is the frequency deviation and f_s is the sampling rate.

To simulate GFSK modulation type either a sensitivity factor K_s or a modulation index K_m can be used. The relation between these are $K_s = \pi K_m$. $K_s = 0.1$ is used to generate GFSK in RML16. This corresponds to $K_m \approx 0.03$, which is low. In Bluetooth, $K_m = 0.5$. This results in generation of unrealistic GFSK modulated signal. Code snapshot in Fig. 3.5d captures the error. In RML22, we use $K_s = 0.5\pi$ ($K_m = 0.5$).

3.5 Deep learning for modulation classification

The received IQ samples y depends on random variables such as d information source, m modulation type, z thermal noise, h channel effect, ζ_{err} SRO, f_{err} CFO and θ_{err} phase offset:

$$y = f(d, m, z, h, \zeta_{\text{err}}, f_{\text{err}}, \theta_{\text{err}}). \tag{3.21}$$

Mapping $f(.)$ generates dataset that follows (3.7). Dataset is generated over instantiations of random variables in Table 3.2 and models in Sec. 3.2.

The goal is to learn the function that maps received samples y to correct modulation type m:

$$m = g(y; d, z, h, \zeta_{\text{err}}, f_{\text{err}}, \theta_{\text{err}}). \tag{3.22}$$

The function should ideally learn the mapping for all possible instantiations of the variables.

Table 3.4. Model architecture and training parameters.

Description	**Values**								
CNN Layer	Input	Conv	Maxpool	Conv	Maxpool	Conv	Maxpool	FC	FC/Softmax
Output Dim.	128 x 2	128 x 64	64 x 64	64 x 64	32 x 64	32 x 32	16 x 32	128	11
Model	Conv. layer: kernel size = 3, padding=1, stride=1, Max. pool layer: kernel size = 2, stride=2, Each Conv. layer include Batch norm and Dropout = 0.3, ReLU and Softmax activation functions, Num. of parameters in model: 117739								
Training	Xavier Initializer, Adam Optimizer, Regularizers: Early stopping and L2, Batch size: 32, 128, 512, Learning rate scheduler (LRS) start=1×10^{-3}, step = 8 epochs, LRS decay = 0.1, Test & validation set: 20% each								

Mapping $g(.)$ is learnt by training DL model on a dataset. Each dataset is generated over modulation types BPSK, QPSK, 8PSK, 16QAM, 64QAM, PAM4, WBFM, CPFSK, GFSK, AM-DSB and SNRs -20 to 20 dB in 2 dB steps. 2000 frames are generated per modulation type and SNR level. The frame contains raw temporal IQ samples. No feature extraction is applied, the DL model intrinsically learns feature extraction in training.

The focus is only on improving data quality and therefore a simple CNN architecture is used. CNN model [30] has three convolutional and two fully connected layers, see architecture and training parameters in Table 3.4. In each convolutional layer, data passes through batch normalization, maximum pooling, dropout, and rectified linear unit (ReLU) activation. Batch normalization [46] helps in convergence of neural network training. Pooling enables downsampling of signal across layers, reducing computation during training. It also reduces sensitivity of convolutional layers to a specific portion of frame. Dropout [3] zeroes out specified percentage of nodes in each layer, adding noise to regularize. Regularization avoids over fitting model to training data. L2 regularization with weight decay 5×10^{-4} noticeably reduced training times. Xavier initialization [47] is used to help avoid exploding and vanishing gradients. Adam optimizer [48] known to be effective and robust, is used. A learning rate scheduler is used with initial learning rate of 10^{-3}, reduced every 8 epochs by a 0.1 factor. A total of 200 epochs is attempted every training run. Early stopping as an additional regularizer is used. Training stops when validation accuracy does not decrease for 16 consecutive epochs. Hyperparameter tuning is done over batch sizes 32, 128 and 512.

Table 3.5. Performance studies done and datasets used.

Type of study	**Description of dataset**
RML22 vs RML16	RML16, Clean, AWGN, Clock, Fading, RML22 with 8 sps
Artifacts	Clean, AWGN, Clock, Fading, RML22
Information source	Analog modulation: Podcast, Digital modulation: Random sequence, Shakespeare and Sherlock Holmes
Clock effects	RML22, max. LO deviation 500 Hz (default) and 50 Hz
Doppler effects	RML22, max Doppler freq. 70 Hz (default) and 1 Hz
Sampler per symbol	RML22, sps = 2 (default) and 8
Signal bandwidth	RML22, signal bandwidth 15 kHz (default) and 100 kHz

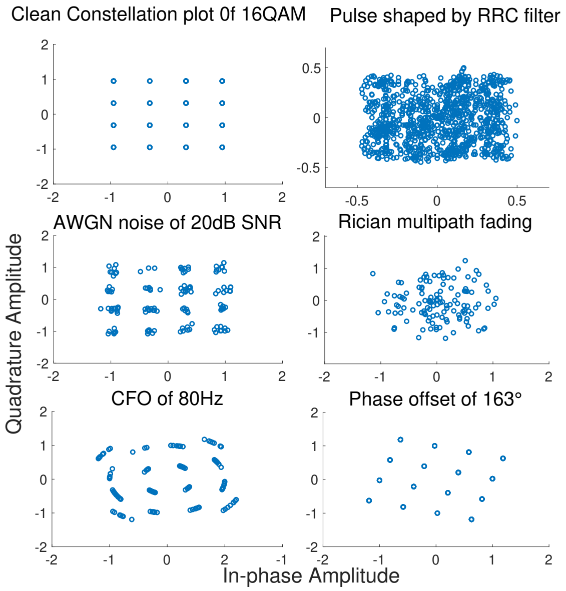


Figure 3.8. Constellation diagrams of clean symbol of (a) 16QAM that are (b) RRC pulse shaped, (c) CFO rotated, (d) phase shifted, (e) AWGN noise applied, (f) Rician faded. It is visually easy to differentiate (a) 16QAM and (g) 64QAM for clean symbols, but harder for symbols that undergo fading (b) and (h).

3.6 Results and Discussion

The end goal is a DL model performing best when deployed real time. A step towards this is generating a quality dataset. The performance of the proposed benchmark dataset RML22 is compared with RML16. Key to improving data quality is understanding the performance impact of artifacts. To study performance impact of artifacts, five datasets (clean, AWGN, clock,

Table 3.6. Accuracy of model trained on RML16 or RML22, and tested on datasets Clean, AWGN, Clock, Fading and RML22. Datasets use sps = 8. Accuracy averaged across entire SNR range.

Tested Trained	Clean	AWGN	Clock	Fading	RML22
RML16	0.63	0.38	0.64	0.6	0.44
RML22	0.87	0.56	0.87	0.86	0.67

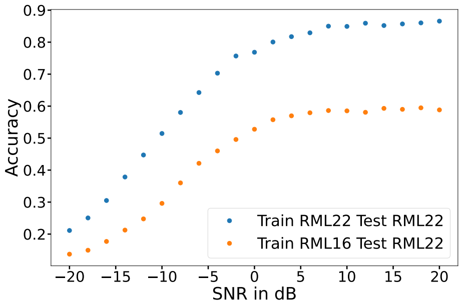


Figure 3.9. Accuracy vs. SNR for models trained on RML16 or RML22 and both tested on RML22.

fading and RML22) are generated:

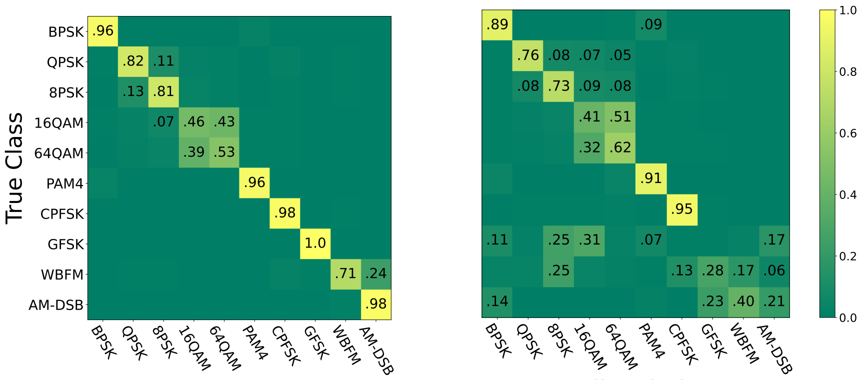
$$y_{\text{clean}}[n] = s[n], \qquad y_{\text{AWGN}}[n] = s[n] + z[n], \tag{3.23a}$$

$$y_{\text{clock}}[n] = e^{j2\pi f_{\text{err}} n T_s + \theta_{\text{err}}} s[n], \tag{3.23b}$$

$$y_{\text{fading}}[n] = \sum_{l} h[l] s[n-l], \tag{3.23c}$$

$$y_{\text{RML22}}[n] = e^{j2\pi f_{\text{err}} n T_s + \theta_{\text{err}}} \sum_{l} h[l] s[n-l-\zeta_{\text{err}}] + z[n]. \tag{3.23d}$$

Further, performance impact of artifact and signal model parameterization is studied as follows: choice of information source, parameterization of clock and Doppler effects, sps and signal bandwidth. Illustration of the effects of artifacts in a constellation diagram is in Fig. 3.8. Constellation diagram represents a complex temporal sequence as a scatter plot in an IQ plane. Though the input to a DL model is a two-dimensional real-valued sequence, constellation diagram illustration provides intuition on difficulties involved in classification. Illustration is for specific instantiation of random variables governing information source, fading, AWGN and clock effects for 128 symbols. AWGN cause spreading in clusters centered on true symbol states as shown



(a) Train RML22, Test RML22

(b) Train RML16, Test RML22

Figure 3.10. Confusion matrices for models trained on (a) RML22 or (b) RML16 and both tested on RML22. Test data is from high SNR region above −5 dB. Accuracy displayed on a 0–1 scale. Values less than 0.05 are not printed for clarity of illustration.

Table 3.7. Artifacts impact on classification. Accuracy of models trained and tested on the datasets Clean, Clock, Fading, AWGN and RML22. Accuracy averaged over SNR range for AWGN and RML22. Clean, clock and fading datasets have no thermal noise added, see Eq. 3.23

Tested Trained	Clean	Clock	Fading	AWGN	RML22
Clean	1	0.38	0.3	0.35	0.16
Clock	0.98	0.99	0.68	0.3	0.33
Fading	0.93	0.84	0.93	0.38	0.43
AWGN	0.92	0.34	0.37	0.62	0.28
RML22	0.84	0.84	0.83	0.52	0.61

in Fig. 3.8e, clock effects cause rotation as shown in Fig. 3.8c and 3.8d, while fading creates spreading across entire symbol space as shown in Fig. 3.8f. Pulse shaping in Fig. 3.8b causes spreading, however it is deterministic and therefore learnable by DL. 16QAM and 64QAM are visually indistinguishable post fading, see Fig. 3.8f and 3.8h. This highlights difficulty in modulation classification due to fading. Similar intuitions can be made for each artifact.

Description of studies done, and datasets used is in Table 3.5 with the results discussed below. Sps of value 2 is used in these tests, unless specified differently. Accuracies are specified on a scale of 0 to 1.

Comparison of RML16 and RML22: Models are trained on RML16 and RML22 and tested on the five datasets. Model trained on RML22 and RML16 have accuracies of 0.67 and 0.44 when tested on RML22, see Table 3.6. RML22 outperforms RML16 by 0.23 because, the

Table 3.8. Effect of information source, changed as randomly generated, Shakespeare and Sherlock Holmes works. Accuracy averaged across entire SNR range.

Trained \ Tested	Random	Shakespeare	Sherlock
Random	0.61	0.58	0.58
Shakespeare	0.58	0.73	0.73
Sherlock	0.56	0.71	0.73

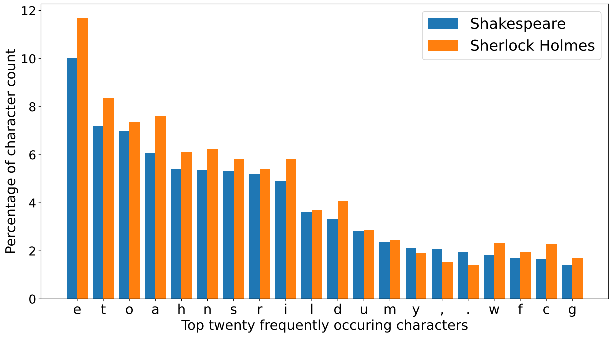
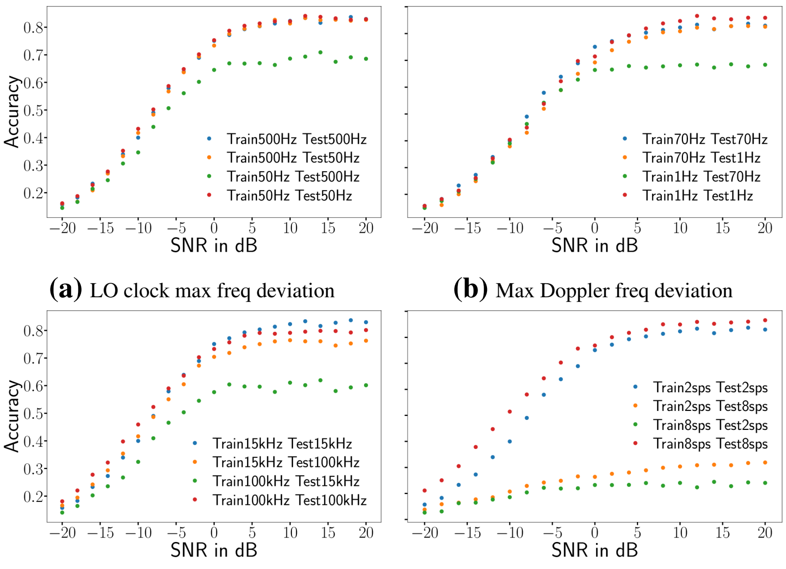


Figure 3.11. Histogram of common characters in the Sherlock Holmes and Shakespeare works.

shortcomings of RML16 are corrected towards generating RML22. RML22 outperforms RML16 by 0.23 on average when tested across the five datasets. Datasets for this test are generated with sps 8 instead of default 2, to be consistent with RML16.

Accuracy versus SNR plot for model trained on RML16 or RML22 and tested on RML22 is in Fig. 3.9. Model trained on RML22 outperforms RML16 across all SNRs, difference being larger in high SNR region. The corresponding confusion matrix is in Fig. 3.10. RML16 accuracy of analog modulation types WBFM and AM-DSB are lower than RML22 by 0.65, due to error in analog information source generation as described in Sec. 3.4.5. GFSK modulation has 0 accuracy for RML16 compared to 1.0 for RML22, due to incorrect modulation index as described in 3.4.6. Phase sensitive modulation types BPSK, QPSK, 8PSK, 16QAM and 64QAM have 0.06 higher accuracy for RML22 over RML16, likely due to incorrect clock effects as described in Sec. 3.4.1.

Artifacts: The impact of channel and HW artifacts are studied, see Table 3.7. Models are trained and tested on datasets Clean, AWGN, Clock, Fading, and RML22. For each case, we observe the following.



(a) LO clock max freq deviation

(b) Max Doppler freq deviation

(c) Signal bandwidths

(d) Sps parameterization

Figure 3.12. Dataset parameterization. (a) Trained and tested on RML22 with LO max frequency deviation 50 and 500 Hz. (b) Trained and tested on RML22 with max Doppler frequency deviation 1 and 70 Hz. (c) Trained and tested on RML22 with signal bandwidths 15 and 100 kHz. (d) Trained and tested on RML22 with sps 2 and 8.

Clean: Perfect accuracy of 1 is achieved. This is a baseline test and ensures DL model perfectly classifies amidst the non-linearity of pulse shaping. Pulse shaping, although non-linear, is deterministic and easy to learn. Further, this case ensures choice of signal model parameters such as number of symbols per frame is appropriate. If symbols in a frame do not sufficiently span the symbol states of a modulation type, performance could be affected [34, pg. 120].

Clock and Fading: DL model is robust to clock and fading effects with high accuracies of 0.99 and 0.93 respectively.

RML22: Model trained on RML22 has lower test accuracy on AWGN dataset compared to RML22, values being 0.52 and 0.61. This is contrary to expectation, since RML22 contains additional artifacts of clock and fading effects on top of AWGN. Data with mild artifacts can perform poorly on model trained on harsh artifacts. In deployment scenario, signal from TX HW with good clock (mild clock artifacts) or high transmit power (high SNR) can perform poorly on model trained on harsh artifacts. In Table 3.6, accuracy on dataset RML22 with sps of 8 is 0.67, a 6% improvement over sps 2. Further, model trained on RML22 tests worse on RML22 over clock and fading, accuracies being 0.61, 0.84, 0.83. Reason is RML22 contains AWGN on top of clock and fading. AWGN effects at low SNR are harsh, see Table 3.9, thus reducing average accuracy to 0.61.

Information source: Performance for different digital information sources are studied, see Table 3.8. In wireless systems, user intends to transmit digital message such as text, electronic mail, or analog message such as audio in frequency modulation (FM) broadcasting. Digital message is transformed bit sequence and mapped to symbol, while analog message is directly mapped to a symbol. Digital information source study includes dataset from random binary sequence, Shakespeare works, Sherlock Holmes works for digital modulation and audio podcast for analog modulation. Wireless systems perform interleaving, encryption etc., that generates randomized binary sequence as input for digital modulation block. Therefore, random binary sequence is most representative of digital information source in wireless systems. For Shakespeare and Sherlock Holmes datasets, their text has decimal numbers assigned through American standard code for information interchange (ASCII) encoding. Direct text to binary sequence for English text does not follow the assumption of a random bit sequence and show interesting results as seen below.

Model trained on random sequence performs best with 0.61 when tested on random sequence as digital source. Models trained and tested on text from Shakespeare and Sherlock Holmes have the highest accuracy of 0.73. DL model has increased performance by learning the intrinsic structure of English language. The histogram in Fig. 3.11 highlights similarity of character count between Shakespeare and Sherlock Holmes works. The high cross performance of model trained on Shakespeare, tested on Sherlock Holmes and vice-versa further validates the hypothesis. This performance improvement shown cannot be leveraged in current wireless systems, which use interleaving and encryption. Artifact and signal model parameterization impact modulation classification performance. The results for study on choice of parameterization is in Fig. 3.12. Datasets are generated with different clock errors, Doppler frequencies, sps and signal bandwidth. Datasets are generated by using RML22 generation code and changing appropriate parameters. Based on the Fig. 3.12 we observe:

Clock effects: Different HWs have varying clock effects based on their XO crystal quality. We study impact of LO maximum frequency deviation parameterization. Model trained with a maximum LO frequency deviation of 500 Hz performs best, see Fig. 3.12a. This is because maximum LO frequency deviation of 500 Hz also contains 50 Hz values.

Doppler frequency: To study Doppler frequency parameterization, a slow pedestrian and fast vehicle equivalent Doppler of 1 and 70 Hz are used. Model trained on lower Doppler of 1 Hz performs poorer, see Fig. 3.12b. Doppler causes rotational effect similar to CFO. Inline with discussion for previous case, Doppler of 70 Hz contains the effect of Doppler 1 Hz and therefore has better performance.

Sps: Sps of a frame is chosen based on a combination of radio frequency card bandwidth, ADC sample rate and computational resources. Sps parameterization values of 2 and 8 are chosen in this study, corresponding to RML22 and RML16. Sps of 2 and 8 correspond to 64 and 16 symbols, in a frame length of 128. Model trained on sps = 2 acts as a random classifier on dataset sps = 8 and vice-versa, see Fig. 3.12d. This is because, the model expects a symbol every N number of samples in its input. In deployment scenarios, models are pre-trained with large synthetic dataset and updated with real OTA dataset via transfer learning. It is essential both datasets have same sps.

Signal bandwidth: The incoming signal bandwidth varies based on technology type and throughput requirements. Models trained and tested on signal bandwiths of 15 and 100 kHz, corresponding to values used in RML22 and RML16. Model trained on larger signal bandwidth of 100 kHz performs poorer, see Fig. 3.12c. Signal with larger bandwidth is affected more by the frequency selective fading. Thus, dataset with larger bandwidth is likely noisier, which hampers learning.

Impact of RML16 shortcomings on prior work is discussed. It is apparent that results of works [49, 50] that use attention mechanisms to estimate and correct clock effects, need revisit.

For other works, it is not apparent if results hold good and therefore experiments needs to be redone.

3.7 Conclusion

We used a data centric approach to solve modulation classification problem for a single carrier, SISO signal model. We provided a benchmark dataset RML22 by correcting the shortcomings in RML16. A performance improvement of 23% was shown in using the corrected dataset RML22. Performance is poorer on RML16 for analog modulation types due to error in information source, for GFSK modulation type due to incorrect modulation index, for phase sensitive modulation types M-ary PSK and M-ary QAM due to issues with clock effects. Performance impact of digital information source were studied. Model trained on digital information source as binary sequence from English text such as Sherlock Holmes and Shakespeare works was shown to outperform a random binary sequence by 12%. A histogram of commonly occurring characters in the information source text indicated improved performance is due to DL model learning intrinsic structure of English language.

The impact of artifact and signal model parameterization were studied for clock effects, Doppler frequency, number of samples per frame and signal bandwidth. The results indicated performance significantly affected by choice of parameters. Therefore, choice of parameterization towards generating dataset should be based upon values seen in deployment.

3.8 Acknowledgements

The text of this chapter is in the full reprint of the material as it appears in V. Sathyanarayanan, P. Gerstoft and A. E. Gamal, ”RML22: Realistic Dataset Generation for Wireless Modulation Classification,” in IEEE Transactions on Wireless Communications, vol. 22, no. 11, pp. 7663-7675, Nov. 2023. The dissertation author was the primary researcher and author of this chapter. The co-authors listed in these publications directed and supervised the research.

3.9 References

[1] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, “Survey of automatic modulation classification techniques: classical approaches and new trends,” *IET Commun.*, vol. 1, pp. 137–156, 2007.

[2] S. Sagiroglu and D. Sinanc, “Big data: A review,” in *Int. Conf. Collab. Tech. and Syst*, pp. 42–47, 2013.

[3] N. Srivastava and G. Hinton, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res*, vol. 15, no. 56, pp. 1929–1958, 2014.

[4] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical Evaluation of Rectified Activations in Convolutional Network,” 2014. arXiv:1410.5093.

[5] D. Steinkraus, I. Buck, and P. Y. Simard, “Using GPUs for machine learning algorithms,” in *Int. Conf. Doc. Anal. Recog*, vol. 2, pp. 1115–1120, 2005.

[6] M. Abadi, A. Agarwal, and P. Barham, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015.

[7] A. Paszke and S. Gross, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Adv. Neural Inf. Process. Syst.*, pp. 8024–8035, 2019.

[8] T. O’Shea, J. Corgan, T. C. Clancy, C. Jayne, and L. Iliadis, “Convolutional Radio Modulation Recognition Networks,” in *Int. Conf. Eng. Appl. Neural Netw*, pp. 213–226, 2016.

[9] X.Liu, D. Yang, and A. El-Gamal, “Deep neural network architectures for modulation classification,” in *Proc. Asilomar Conf. Signals, Syst., Comput.*, pp. 915–919, 2017.

[10] T. O’Shea, T.Roy, and T. Clancy, “Over-the-Air Deep Learning Based Radio Signal Classification,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, 2018.

[11] M. Patel, X. Wang, and S. Mao, “Data Augmentation with Conditional GAN for Automatic Modulation Classification,” in *ACM Workshop Wireless Secur. Mach. Learn.*, pp. 31–36, 2020.

[12] A. Ali, F. Yangyu, and S. Liu, “Automatic modulation classification of digital modulation signals with stacked autoencoders,” *Digital Signal Process.*, vol. 71, pp. 108–116, 2017.

[13] R. Zhou, F. Liu, and C. Gravelle, “Deep Learning for Modulation Recognition: A Survey With a Demonstration,” *IEEE Access*, vol. 8, pp. 67366–67376, 2020.

[14] K. Goel and L. Orr, “Data centric AI.”

[15] “NeurIPS data centric AI workshop.”

[16] T. O’Shea and N. West, “Radio Machine Learning Dataset Generation with GNU radio,” in *Proc. GNU Radio Conf.*, 2016.

[17] "GNU radio open source software radio toolkit."

[18] T. O'Shea, "RadioML methodology source code."

[19] N. Bitar, S. Muhammad, and H. Refai, "Wireless technology identification using deep Convolutional Neural Networks," in *IEEE Int. Symp. Pers. Indoor Mob. Radio Commun.*, pp. 1–6, 2017.

[20] P. Wang, M. Vindiola, and M. Markowski, "Deep learning for modulation and coding rate classification of OFDM," in *Disruptive Technol. Inf. Sci. IV*, vol. 11419, pp. 82–92, 2020.

[21] T. Ngo, B. Kelley, and P. Rad, "Deep Learning Based Prediction of Signal-to-Noise Ratio (SNR) for LTE and 5G Systems," in *Int. Conf. Wirel. Mob.*, pp. 1–6, 2020.

[22] R. Dreifuerst, R. Heath, M. Kulkarni, and J. Charlie, "Deep Learning-based Carrier Frequency Offset Estimation with One-Bit ADCs," in *IEEE Workshop Signal Process. Adv. Wirel. Commun.*, pp. 1–5, 2020.

[23] O. Dobre, A. Abdi, and Y. Bar-Ness, "Blind modulation classification: a concept whose time has come," in *IEEE Sarnoff Sym. Adv. Wired Wireless Commun.*, pp. 223–228, 2005.

[24] T. O'Shea, "GNU radio dynamic channel model."

[25] F. Ren and Y. R. Zheng, "A low-complexity hardware implementation of discrete-time frequency-selective Rayleigh fading channels," in *IEEE Int. Symp. Circuits Syst*, pp. 1759–1762, 2009.

[26] A. Alimohammad, S. Fard, B. Cockburn, and C. Schlegel, "Compact Rayleigh and Rician fading simulator based on random walk processes," *IET Commun.*, vol. 3, pp. 1333–1342, 2009.

[27] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D Multi-Cell Channel Model With Time Evolution for Enabling Virtual Field Trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, 2014.

[28] S. Ju, O. Kanhere, Y. Xing, and T. Rappaport, "A Millimeter-Wave Channel Simulator NYUSIM with Spatial Consistency and Human Blockage," in *IEEE Glob. Commun. Conf.*, pp. 1–6, 2019.

[29] V. Sathyanarayanan, M. Wagner, and P. Gerstoft, "Over the air performance of deep learning for modulation classification across channel conditions," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, pp. 157–161, 2020.

[30] V. Sathyanarayanan, A. Jolly, and P. Gerstoft, "Novel training methodology to enhance deep learning based modulation classification," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2021.

[31] "GNU Radio documentation of Frequency modulation."

[32] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice Hall, 2001.

[33] D. Pozar., *Microwave and Rf Design of Wireless Systems*. Wiley Publishing, 2000.

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

[35] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[36] M. Mueller, "GNU Radio thermal noise block."

[37] M. Rice, *Digital Communications: A Discrete-Time Approach*. Pearson Education Inc, 2012.

[38] J. Zyren and W. McCoy, "Overview of the 3GPP Long Term Evolution Physical Layer."

[39] P. Kolb, "Securing compartmented information with smart radio systems."

[40] S. Ramjee, S. Ju, D. Yang, X. Liu, A. El-Gamal, and Y. Eldar, "Ensemble Wrapper Subsampling for Deep Modulation Classification," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1156–1170, 2021.

[41] X. Zhang, T. Seyfi, S. Ju, S. Ramjee, A. El-Gamal, and Y. Eldar, "Deep Learning for Interference Identification: Band, Training SNR, and Sample Selection," in *IEEE Workshop Signal Process. Adv. Wireless Commun.*, pp. 1–5, 2019.

[42] "3rd Generation Partnership Project (3GPP). Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) Radio Transmission and Reception. Version 14.3.0. Number 36.101."

[43] "Universal Software Radio Peripheral N310."

[44] S. Hauser, W. Headley, and A. Michaels, "Signal detection effects on deep neural networks utilizing raw IQ for modulation classification," in *IEEE Mil. Commun. Conf.*, pp. 121–127, 2017.

[45] T. O'Shea, "RML16 Source code errors.." SNR error: https://github.com/radioML/dataset/blob/4ecf612cfbc5bfc80eb8b0dbe63ed685d0a73c44/generate_RML2016.10a.py#L45-L46, Information source error: https://github.com/radioML/dataset/blob/4ecf612cfbc5bfc80eb8b0dbe63ed685d0a73c44/source_alphabet.py#L35-L41, GFSK modulation error: https://github.com/radioML/dataset/blob/4ecf612cfbc5bfc80eb8b0dbe63ed685d0a73c44/transmitters.py#L72-L80.

[46] S. Ioffe and C.Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Int. Conf. on Mach. Learn.*, vol. 37, pp. 448–456, 2015.

[47] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, 2010.

[48] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Int. Conf. Learn. Represent.*, 2015.

[49] K. Yashashwi, A. Sethi, and P. Chaporkar, “A Learnable Distortion Correction Module for Modulation Recognition,” *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 77–80, 2019.

[50] T. O’Shea, P. Latha, and D. Batra, “Radio transformer networks: Attention models for learning to synchronize in wireless systems,” in *Proc. Asilomar Conf. Signals, Syst., Comput.*, pp. 662–666, 2016.

Chapter 4

Novel Training Methodology to Enhance Deep Learning Based Modulation Classification

Automatic Modulation Classification (AMC) is central to dynamic spectrum sensing. This work aims to improve the performance of deep learning (DL) models applied to AMC. Novel training methodology is introduced to improve performance. Over-the-air (OTA) data is collected using software-defined radio (SDR) over a range of modulation types and SNR levels. Collected dataset is partitioned into subsets across SNR levels. A group of identical models is trained on these multiple subsets and performance is compared against model trained on whole dataset. Efforts are taken to identify and isolate corrupted OTA data caused by interferences. Convolutional neural network (CNN) based architecture is used. We show an average improvement of 6% in the classifier's performance on OTA data using this SNR partitioning approach.

4.1 Introduction

Modulation is the act of embedding information onto an electromagnetic wave by manipulating amplitude, frequency or phase of a signal. AMC detects modulation type of wireless signal without apriori knowledge. It is an essential part of spectrum sensing. Latest wireless communication designs rely on accurate spectrum sensing to maximize usage of under utilized spectrum. Spectrum sensing is thus critical to newer standards of cellular and WiFi technologies.

Traditional approaches to modulation classification used hand-crafted features created from in-phase and quadrature-phase (IQ) signals. Popular analytical methods include likelihood based [1], feature based [2] and artificial neural network [3] based techniques. More recent analytical approaches can be found in [4, 5, 6, 7]. Inspired by recent success of DL in computer vision, DL is used for AMC and has greatly outperformed analytical methods [8, 9, 10, 11]. Popular architectures used are CNN, CNN with residual connections (ResNet), Convolutional Long Short-term Deep Neural Network (CLDNN) and Long Short-Term Memory (LSTM). Most works of DL over AMC use the two standard synthetic datasets [8] and [9].

Current approaches to applying DL over AMC typically include attempting varying DL architectures to the standard datasets. In this work, we propose a new training methodolgy via signal to noise ratio (SNR) partitioning. Models are trained on subset of the data and test accuracies compared over model trained on complete dataset. We showcase an overall accuracy improvement of 6% using the SNR partitioning approach. SDR USRP N310 are used to collect OTA datasets in this work. Datasets are collected for channels emulating simple additive white gaussian noise (AWGN) and multipath non line of sight (NLOS).

Full version of paper to be submitted post-acceptance shall be updated as described below. New SNR partitioning approach shall be tested on benchmark datasets RADIOML 2018.01A and RADIOML 2016.10A. Perfect SNR estimates have been assumed in the results presented. Blind SNR estimate algorithms shall be implemented on test data and test accuracy results updated.

4.2 SNR partitioning approach

The artifacts experienced by a wireless signal can be modelled as,

$$r(t) = Ae^{j2\pi\Delta ft}e^{j\theta}\sum_{k=1}^{K} e^{j\phi_k}\tilde{s}_k g(t-(k-1)T-\varepsilon)+z(t), \tag{4.1}$$

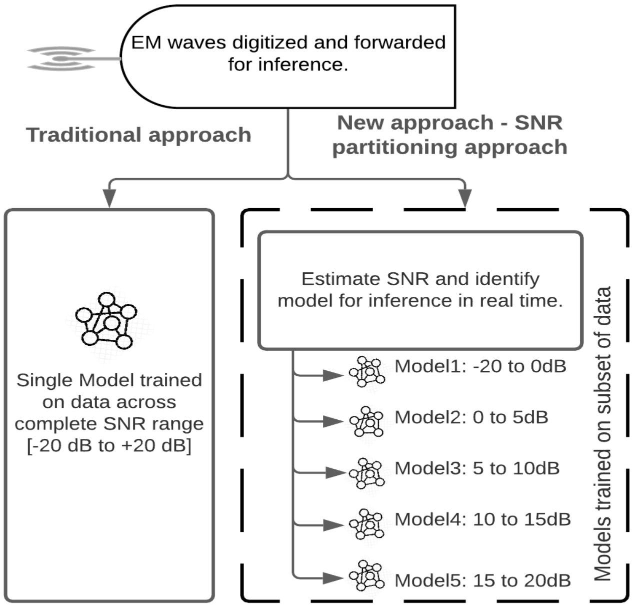

EM waves digitized and forwarded for inference.
Traditional approach
New approach - SNR partitioning approach
Single Model trained on data across complete SNR range [-20 dB to +20 dB]
Estimate SNR and identify model for inference in real time.
Model1: -20 to 0dB
Model2: 0 to 5dB
Model3: 5 to 10dB
Model4: 10 to 15dB
Model5: 15 to 20dB
Models trained on subset of data

Figure 4.1. Illustration of SNR partitioning approach. NDA SNR estimation algorithm used for SNR estimation. Appropriate model picked for inference based on SNR estimate.

where A is path loss, Δf is frequency offset, θ is phase offset, ϕ_k is phase jitter, $g(t)$ is channel impulse, T is symbol period, $\varepsilon < T$ is time offset, $\{\tilde{s}_k\}_{k=1}^{K}$ are K modulation symbols. Random variables in Eq. (4.1) are represented as $\boldsymbol{u} = \left[z, A, \Delta f, \theta, \{\phi_k\}_{k=1}^{K}, g(t), \varepsilon, \{\tilde{s}_k\}_{k=1}^{K}\right]^T$. The goal is finding a function $f(.)$ mapping signal $r(t, \boldsymbol{u})$ to the correct modulation type, i. A DL classifier learns $f(.)$ by training on occurrences of $\boldsymbol{u}$.

Learning a function can be difficult, especially over feature space spanning numerous random variables in u. Here, we learn over subset of feature space of the underlying probability distribution. We learn over data partitioned across the random variables: path loss A and noise z that constitute SNR. Likely, a DL model learns better over the subset of data and shows higher test accuracies.

Fig. 4.1 illustrates the SNR partitioning approach. We leverage prior work in non-data aided (NDA) SNR estimation [12, 13, 14, 15] to improve test performance of DL over AMC. The dataset is partitioned into subsets and each model learns on a specific subset of the SNR region. During inference the SNR estimation algorithm will identify the SNR subset the data belongs to. Data is passed through the model trained on identified subset for inference.

Partitioning of SNR used in this work is specified in Table. 4.1. SNR estimation algorithms perform poorly below 0 dB. Therefore, SNR range from -20 to 0 dB form one subset. SNR range from 0 to 20 dB is further split into four equispaced subsets. Choice of SNR partitioning is a hyper parameter. Availability of perfect SNR estimates is assumed in this work. Models trained on one SNR subset are able to classify data belonging to a different subset with good accuracy [11]. There is thus a good margin for error in SNR estimation.

4.3 Experimental Testbed

An illustration of the data generation and test setup is shown in Fig. 4.2. Random data bits are generated over uniform distribution. They are mapped to modulation symbols, pulse shaped using raised root cosine filter with roll-off factor specified in Table. 4.1.

Data is transmitted and received via USRP N310 for two channels, 11 modulation types and 21 SNR levels as outlined in Table. 4.1. The data is captured separately for channels emulating AWGN and NLOS channels in an indoor environment. Receive USRP noise floor is measured at around -87 dBm. Transmit system host controls an external programmable attenuator to maintain transmit power relative to measured noise floor, see (4.2). This helps achieve requested SNR.

$$y_{tx} = s + z, \tag{4.2a}$$

$$z \sim \mathscr{CN}(0, \sigma^2), \tag{4.2b}$$

$$SNR = 10\log_{10}\left(\frac{||s||_2^2}{||z||_2^2}\right), \tag{4.2c}$$

where y_{tx}, s and z ε $\mathbb{C}^{128}$, correspond to the transmitted signal, clean signal and noise respectively. z is complex Gaussian with variance σ^2 such that a specific SNR is met.

This method of setting SNR for modulation data is sanitized by calculating SNR on a pure tone. A pure tone is transmitted at a specific power relative to noise floor. SNR values estimated for pure tone via least squares approach match intended value within an error margin of 0.2 dB.

$$x_{\text{tone}}[n] = \alpha e^{j[2\pi(f_c+f_{\text{err}})(\frac{n}{f_s})+\phi_{\text{err}}]}, \tag{4.3a}$$

$$y_{\text{tone}}[n] = x_{\text{tone}}[n] + z[n], \tag{4.3b}$$

$$\widehat{\alpha}, \widehat{f}_{\text{err}}, \widehat{\phi}_{\text{err}} = \underset{\alpha, f_{\text{err}}, \theta_{\text{err}}}{\arg\min} \frac{1}{N} \sum_{n=1}^{N} (y_{\text{tone}}[n] - x_{\text{tone}}[n])^2, \tag{4.3c}$$

$$\widehat{z}[n] = y_{\text{tone}}[n] - \widehat{\alpha} e^{j[2\pi(f_c+\widehat{f}_{\text{err}})(\frac{n}{f_s})+\widehat{\phi}_{\text{err}}]}, \tag{4.3d}$$

$$SNR_{\text{estimate}} = 10\log_{10}\left(\frac{\left\|\widehat{\alpha} e^{j[2\pi f_c(\frac{n}{f_s})]}\right\|_2^2}{\|\widehat{z}[n]\|_2^2}\right), \tag{4.3e}$$

where x_{tone} and y_{tone} represent received signal with and without thermal noise z. α, f_{err}, ϕ_{err} represent path loss, frequency error and phase error that are unknown. n, f_s and f_c represent sample index, sampling rate and transmit tone frequency.

Tone frequency is at an offset of 50 KHz from center frequency to avoid leakage from DC offset. Least squares (LS) approach outlined in (4.3c) is used to estimate parameters of the tone. A two staged coarse and fine grid search over possible values of α, f_{err}, ϕ_{err} is done to solve the LS problem. Coarse search uses large step size and fine search uses small step size. In first stage, f_{err} search space uses knowledge of hardware clock stability. The step size values and search space for α and ϕ_{err} are hyper parameters. Estimated values from coarse search are used as seed for fine search. Estimated parameters of fine search are used to compute SNR using (4.3d) and (4.3e).

TX and RX USRPs are connected via cable to emulate AWGN channel. To emulate NLOS, antennae are placed with blocked line of sight [16]. In the NLOS data collection, data are corrupted from unexpected user movements, interferers etc. Channel is thus monitored by transmitting and receiving pure tone via co-located antennae. LS SNR estimation is computationally

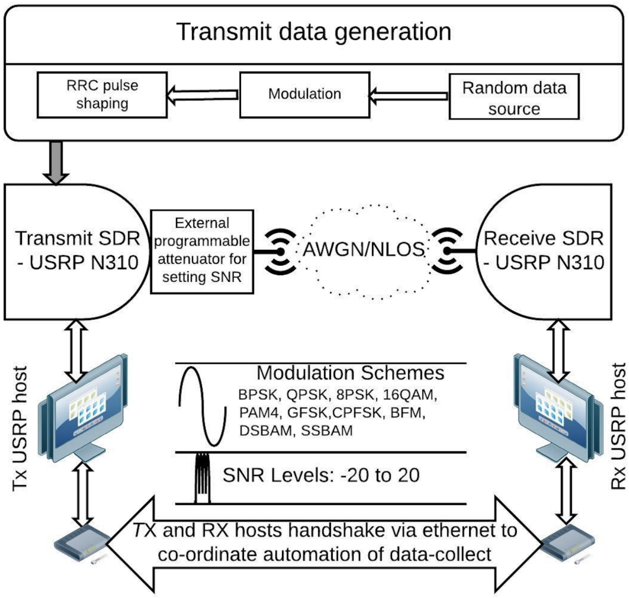

Transmit data generation
RRC pulse shaping
Modulation
Random data source
Transmit SDR - USRP N310
External programmable attenuator for setting SNR
AWGN/NLOS
Receive SDR - USRP N310
Tx USRP host
Modulation Schemes
BPSK, QPSK, 8PSK, 16QAM, PAM4, GFSK,CPFSK, BFM, DSBAM, SSBAM
SNR Levels: -20 to 20
Rx USRP host
TX and RX hosts handshake via ethernet to co-ordinate automation of data-collect

Figure 4.2. Data collect workflow

Table 4.1. Data generation and testbed parameters

Description	**Values**
Modulation types	SSBAM,DSBAM,BFM,CPFSK,GFSK,PAM4, 64QAM,16QAM,8PSK,QPSK,BPSK
SNR Levels	-20 to $+20$ dB in steps of 2 dB
SNR Subsets	Subset1: -20 to 0 dB, Subset2: 0 to 5 dB, Subset3: 5 to 10 dB, Subset4: 10 to 15 dB, Subset5: 15 to 20 dB, Complete set: -20 to 20 dB
Channel types	AWGN and Non-Line-of-sight (NLOS).
Center frequency	900 MHz and 903 MHz ISM band for modulation data and tone respectively
Sampling rate	1.25 MHz and 0.25 MHz for modulation data and tone respectively
Rolloff factor	roll-off picked from uniform distribution $U(0.1,0.4)$, every input frame
Duration of each input frame	102.5 μs
Input frame dimensions	128 by 2
Symbols per input frame	16
Samples per symbol	8
Dataset size per channel type	693K for training and 115K for testing
Dataset size per channel type per per modulation per SNR	3k for training, 5K for testing
Clock source	Internal clock with 0.1 ppm frequency stability
Antennae type	External Antennae 2.2 dBi Gain, Model No. APAMSTJ-138

expensive. To reduce time of computation, 1 ms chunks are extracted every 100 ms. Estimated SNR on these 1ms chunks confirmed data capture is clean.

Table 4.2. CNN architecture, training and hyper parameter tuning details

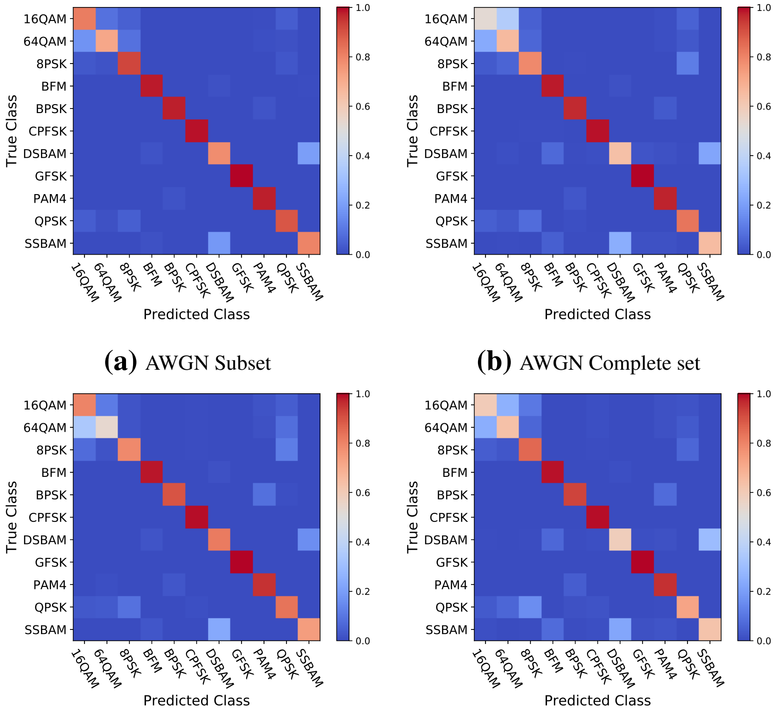
CNN layers	Output dimensions	Model layer and tuning details
Input	128 × 2	Conv layer: kernel size = 8, padding=4, stride=1
Conv	128 × 64	Max pool layer: kernel size = 2, stride=2
Maxpool	64 × 64	Optimizers used: Adam, SGD
Conv	64 × 64	Learning rates: 1e-2, 1e-3, 1e-4
Maxpool	32 × 64	Batch size: 2,4,8,16,32
Conv	32 × 32	Initialization: Xavier
Maxpool	16 × 32	Activation: ReLU
FC	512 × 128	Regularization: Early stopping, Batch norm
FC/Softmax	128 × 11	Library: Pytorch

4.4 Models

CNN is a popular choice of DL architecture [10, 8] for modulation classification. Details of CNN architecture proposed and used in this work are in Table. 4.2. CNN model with residual connections (ResNet) was attempted, but did not show good test accuracy likely due to over fitting. Dimension of input frame to model is 128 by 2 and output is 11. Input frame contains 128 complex wireless samples represented as 128 by 2 real array. Output represents the 11 modulation types.

The first 3000 input frames of the capture per modulation and SNR level, are used for training and validation. Succeeding 5000 input frames are for testing. Choice of training data size is consistent with [9], where it was shown larger data size doesn't improve accuracy. However, larger test data is favoured as it reduces variance in accuracy.

Hyper parameter tuning is essential to find the optimal choice of learning rate, batch size, etc. for the given training data. Training process is repeated for the learning rates 0.01, 0.001 and 0.0001. The batch sizes used are 2, 4, 8, 16 and 32. Adam and Stochastic Gradient Descent (SGD) optimizers were attempted for the specified values of learning rate and batch size. Best test accuracies are obtained using Adam optimizer, for a learning rate of 0.001 and batch size of 16.



(a) AWGN Subset

(b) AWGN Complete set

(c) NLOS Subset

(d) NLOS Complete set

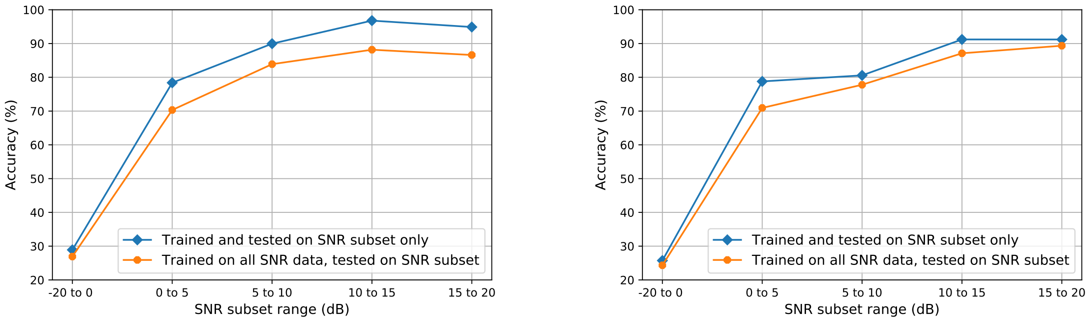
Figure 4.3. Confusion matrices for models trained on subset and complete dataset for positive SNR region.

4.5 Results and Discussion

There are two datasets belonging to AWGN and NLOS channels. An identical CNN architecture introduced in section 4.4 is used in all the tests. For each dataset, there is a model trained on complete SNR range and 5 models trained on each of the 5 SNR subsets, see Table. 4.1. Test data is split into 5 parts corresponding to SNR subsets. For each of 5 parts, test accuracies are computed for model trained on complete SNR and model trained on same SNR subset.

In Fig. 4.4, the AWGN and NLOS dataset test accuracies for models trained on SNR subset versus complete SNR set is displayed. Improvements in test accuracies of model trained on subset over complete SNR set is listed in Table 4.3. The average accuracy improvement for AWGN and NLOS datasets is 8% and 4%, for positive SNR region.

The SNR subset −20 to 0 dB has least improvement in accuracy. It is hard for a classifier



(a) AWGN Subset

(b) AWGN Complete set

Figure 4.4. Accuracy Comparison for SNR partitioning approach. Models are trained on subset and complete set. Test accuracies corresponding to each of the five subsets are plotted for AWGN and NLOS datasets.

to learn at low SNR, likely placing a cap on the performance benefits that could be obtained. Thus even with SNR partitioning approach, we do not see improvements in negative SNR region. Subset 0 to 5 dB has the best improvement in accuracy, when averaged across both channels. The overall improvements in accuracy is higher for AWGN than NLOS. This is likely because NLOS dataset is collected in harsher multipath channel and more difficult to learn. Overall, we see improvements in test accuracy on all subsets across both channels.

Fig. 4.3 contains confusion matrices for models trained on subset and complete dataset. Performance improvements is seen only for positive SNR and thus confusion matrices are plotted for test data in this region. Test results of four models trained on positive SNR region are averaged to obtain confusion matrix for subset. This is compared against model trained on complete dataset. 16QAM and DSBAM have maximum accuracy improvement for models trained on subset over complete set. This is noticed across AWGN and NLOS datasets. 64QAM for NLOS performs better for model trained on complete set over subset, contradicting the general trend.

The cost paid for performance improvements is extra computation both at training and inference. This is because we train on five classifiers and use SNR estimation algorithm at inference, compared to single model at training and testing.

Table 4.3. Gain in classification performance (in %) for each subset using SNR partitioning method

	Subset1 **-20 to 0 dB**	**Subset2** **0 to 5 dB**	**Subset3** **5 to 10 dB**	**Subset4** **10 to 15 dB**	**Subset5** **15 to 20 dB**	**Avg. over** **SNR $>=$ 0 dB**
AWGN	1.98	8.07	6.07	8.62	8.28	7.76
NLOS	1.4	7.85	2.77	4.1	1.85	4.14

4.6 Conclusion

This work has demonstrated improved performance when incorporating new training methodology via SNR partitioning approach. A comparative study of the classifier's test accuracy is done for the model trained on the SNR subsets and complete dataset. There is an average improvement of 8% and 4% for the AWGN and NLOS channels respectively for the positive SNR region. The cost of improved accuracies is more computation at training and inference. OTA data captured using SDR is used in this work.

4.7 Acknowledgements

The text of this chapter is in the full reprint of the material as it appears in V. Sathyanarayanan, A. Jolly and P. Gerstoft, ”Novel Training Methodology to Enhance Deep Learning Based Modulation Classification,” 2021 55th Asilomar Conference on Signals, Systems, and Computers, pp. 356-360, 2021. The dissertation author was the primary researcher and author of this chapter. The co-authors listed in these publications directed and supervised the research.

4.8 References

[1] J. A. Sills, "Maximum-likelihood modulation classification for PSK/QAM," *Proc. IEEE MILCOM*, vol. 1, pp. 217–220, 1999.

[2] S. S. Soliman and S. Z. Hsue, "Signal classification using statistical moments," *IEEE Trans. Commun.*, vol. 40, pp. 908–916, 1992.

[3] A. K. Nandi and E. E. Azzouz, "Modulation recognition using artificial neural networks," *IEEE Signal Process.*, vol. 56, pp. 165–175, 1997.

[4] A. B. Sergienko and A. V. Osipov, "Digital modulation recognition using circular harmonic approximation of likelihood function," *IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 3460–3463, 2014.

[5] Z. Zhu and A. K. Nandi, "Modulation classification in MIMO fading channels via expectation maximization with non-data-aided initialization," *IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 3014–3018, 2015.

[6] S. Fki, A. Aïssa-El-Bey, and T. Chonavel, "Blind equalization and automatic modulation classification based on pdf fitting," *IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 2989–2993, 2015.

[7] H. Sarieddeen, M. M. Mansour, L. M. A. Jalloul, and A. Chehab, "Efficient near optimal joint modulation classification and detection for MU-MIMO systems," *IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 3706–3710, 2016.

[8] T. J. O'Shea, J. Corgan, T. C. Clancy, C. Jayne, and L. Iliadis, "Convolutional radio modulation recognition networks," *Int. Conf. Eng. Appl. Neural Netw.*, pp. 213–226, 2016.

[9] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, pp. 168–179, 2018.

[10] X. Liu, D. Yang, and A. E. Gamal, "Deep neural network architectures for modulation classification," *Proc. Asilomar Conf. Signals, Syst., Comput.*, pp. 915–919, 2017.

[11] X. Wang, S. Ju, X. Zhang, S. Ramjee, and A. E. Gamal, "Efficient training of deep classifiers for wireless source identification using test snr estimates," *IEEE Wirel. Commun.*, vol. 9, pp. 1314–1318, 2020.

[12] S. Dan and L. Ge, "On the blind snr estimation for IF signals," *IEEE Int. Conf. Inn. Comp. Info. Cont.*, vol. 2, pp. 374–378, 2006.

[13] M. Hamid, N. Björsell, and S. B. Slimane, "Sample covariance matrix eigenvalues based blind SNR estimation," *IEEE Int. Instrum. Meas. Technol. Proc.*, pp. 718–722, 2014.

[14] R. Matzner and F. Englberger, "An SNR estimation algorithm using fourth-order moments," *IEEE Int. Symp. Inf. Theory.*, p. 119, 1994.

[15] R. L. Valcarce and C. Mosquera, “Sixth-order statistics-based non-data-aided SNR estimation,” *IEEE Commun. Lett.*, vol. 11, pp. 351–353, 2007.

[16] V. Sathyanarayanan, “Video of NLOS channel condition used for data capture,” *Youtube*, 2020. Available at https://www.youtube.com/watch?v=FP0jM7JTrSg&ab.

Chapter 5

Over the Air Performance of Deep Learning for Modulation Classification Across Channel conditions

Deep learning (DL) models used for modulation classification are mostly trained on simulated data. Their performance drops significantly on real test data, due to disparity in probability distributions between simulated and real data. The eventual goal is building a DL model classifying modulation type accurately on real data. This work empirically studies the performance impact due to disparity in probability distributions between training and test data. We borrow best performing deep learning models from literature for our analysis. Models are tested on data belonging to channel conditions they were trained on and otherwise. Software defined radios (SDR) collect training and test data under channel conditions of additive white Gaussian noise, line-of-sight (LOS) and non-line-of-sight (NLOS). Convolutional neural network (CNN) and Residual neural network (ResNet) architectures are used. Test accuracies of the models are compared across model architectures, channel conditions, modulation types and SNR. Performance results of DL models on real data, are presented for wide set of scenarios. Dataset is available for download and can be used for evaluating deep learning models.

5.1 Introduction

Modulation is the manipulation of the amplitude, frequency or phase of an electromagnetic (EM) wave with the intent of transmitting information. In general the transmitter informs the receiver about the modulation scheme. However, for dynamic spectrum sensing and electronic warfare inferring modulation type, modulation classification is important. Early works used hand crafted features from raw temporal signals such as zero crossing locations [1], square law classifiers [2], phase based classifiers [2], and statistical moment classifiers [3]. Recent analytical approaches are in [4, 5, 6]. Latest advancements in DL model architecture, computing software and hardware have made DL an accessible tool. The traditional models worked well for specific modulation types and channel conditions but there was no approach that was universally applicable for all modulation types. DL models with many parameters show much improved performances [7, 8, 9, 10, 11, 12, 13, 14, 15].

The goal is to build a robust DL model that can accurately predict the modulation type of real data under all possible channel conditions. The focus mostly has been to attempt different DL architectures to improve performance on the benchmark RadioML dataset [7]. Performance results for real data using non-DL techniques such as support vector machine [16] and likelihood-based approach[17] are modest. Real data performance on a subset of SNR, modulation and channel types with DL based approaches, fully-connected neural networks and convolutional-auto encoders is in [18, 19].

A carefully trained DL model on a large synthetic dataset was developed in [15] with excellent performance on simulated test data. However, their performance on real data dropped significantly under benign LOS fading conditions for a high SNR of 10 dB. We performed over the air tests of this DL model using the setup in Fig. 5.1. Radiated tests were done for SNR above 10 dB. The performances match when the LOS component was significant. When the antenna were moved further away, the performance was erratic, sometimes dropping significantly and was over-sensitive to antennae placement. The SNR at the receiver was maintained by transmitting signal with digitally added AWGN noise and at a power well above receiver noise floor. These results pose a question on the over-reliance on simulated data for training. The plausible reason is that the training and test data did not belong to the same probability distributions. A DL model trained on data from an AWGN channel will perform poorly when tested on data from fading channel since the probability distribution governing a pure AWGN channel is different from a fading channel.

We conduct quantitative studies of DL performance tested on data belonging to channel conditions they were trained on and otherwise. Test accuracies are compared across DL architectures, channel conditions, modulation types, and SNR. The availability of low-cost SDRs and open source GNU radio has made real data collection accessible. SDRs are used to collect training and test data in a AWGN channel and radiated LOS and NLOS fading channels. The DL models used are CNN from [20] and ResNet from [15] since they performed well on the RadioML dataset [7].

5.2 Wireless System Model

The noiseless electromagnetic signal transmitted over-the-air is modeled as follows.

$$r(t) = Ae^{j2\pi\Delta f t}e^{j\theta}\sum_{k=1}^{K} e^{j\phi_k}\tilde{s}_k g(t-(k-1)T-\varepsilon)\,, \qquad 0 \le t \le KT \tag{5.1}$$

where A is the attenuation due to path loss between transmitter and receiver, Δf is the carrier frequency offset, θ is carrier phase offset, ϕ_k is phase jitter, $g(t)$ is the effective impulse response of the channel given by the convolution of the transmitter pulse shaping filter and the channel impulse response, T is a symbol period, $\varepsilon < T$ is the time offset from the start of a symbol period and $\{\tilde{s}_k\}_{k=1}^{K}$ are K complex transmitted data symbols drawn from a finite size modulation format. We represent the set of parameters in (5.1) through $\boldsymbol{u} = \left[A, \Delta f, \theta, \{\phi_k\}_{k=1}^{K}, g(t), \varepsilon, \{\tilde{s}_k\}_{k=1}^{K}\right]^T$.

The goal is finding a function $f(.)$ mapping signal $r(t, \boldsymbol{u})$ to the correct modulation type, i.

$$i = f(r(t), \boldsymbol{u}). \tag{5.2}$$

We learn function $f(.)$ by training a deep learning model on all possible occurrences of $\boldsymbol{u}$.

5.3 Data and Experimental Testbed

An illustration of the data generation and test setup is in Fig. 5.1. Random data bits over uniform distribution are generated, mapped to modulation symbols, pulse shaped using raised root cosine filter and Gaussian noise added to set the SNR level. Gaussian noise is included as follows,

$$Y_{tx} = S + Z, \tag{5.3}$$

$$Z \sim \mathbb{CN}(0, \sigma^2), \tag{5.4}$$

$$SNR = 10\log_{10}\left(\frac{\|S\|_2^2}{\|Z\|_2^2}\right), \tag{5.5}$$

where Y_{tx}, S and Z ε $\mathbb{C}^{1024}$, correspond to the transmitted signal, clean signal and noise respectively. Z is sampled from a complex valued Gaussian distribution with variance σ such that a specific SNR is met. SNR is set during transmit data generation since this is easier in comparison to controlling received signal power over the noise floor.

The generated data is transmitted and received using USRP N310 operating in the 900 MHz ISM band at a sampling rate of 1.25 MHz. We implement three types of channels. AWGN channel is set up by connecting the transmit and receive ports of two USRPs using RF cables. For LOS and non-LOS channels, antennaes are placed as shown in Fig. 5.2. It is likely LOS path dominates reflected path which makes the LOS channel a rician channel. For NLOS channel, there is likely no single dominant path due to obstruction. This assumption makes NLOS channel a rayleigh channel.

For each of the channel conditions, in-phase and quadrature phase (IQ) samples are transmitted and received across 11 modulation types and 21 SNR levels as detailed in Table 5.1. Amongst them GFSK, CPFSK and BFM are frequency modulation types, BPSK, QPSK and 8PSK are phase modulation types, PAM4, DSBAM and SSBAM are amplitude modulation types while 16QAM and 64QAM belong to a combination of phase and amplitude. 8PSK, 16QAM and 64QAM are tightly packed with lesser spacing between symbols and therefore more prone to misclassification.

The input fed to the DL has dimension 1024 by 2 corresponding to a signal of duration 820 μs. There are 5000 data points for each instance of modulation and SNR. The dataset size for data points belonging to all modulation types and SNR is 1.155 million. There are three datasets each belonging to channel conditions AWGN, LOS and NLOS. The dataset is available [21].

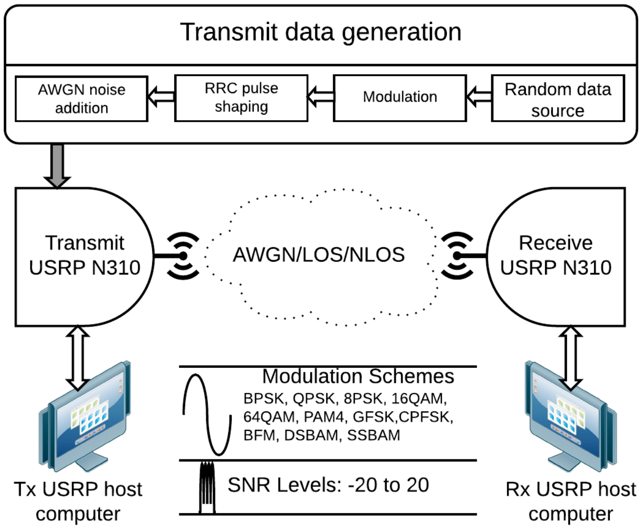
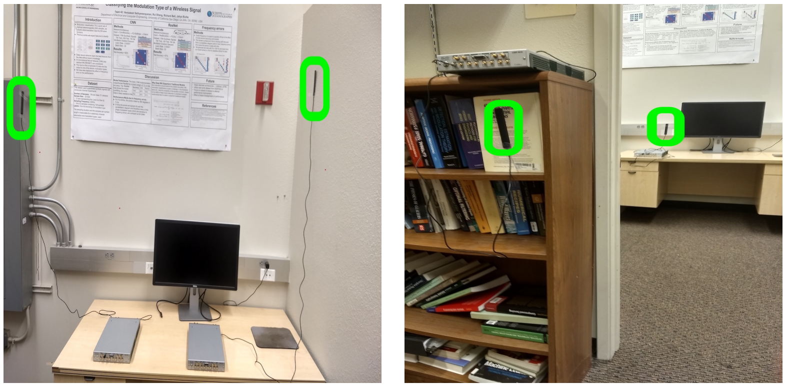


Figure 5.1. Data collect workflow.

Table 5.1. Data generation and testbed parameters

Description	**Values**
Modulation types	SSBAM,DSBAM,BFM,CPFSK, GFSK,PAM4,64QAM,16QAM, 8PSK,QPSK,BPSK
SNR Levels	-20 to $+20$ dB in steps of 2 dB
Channel types	AWGN, LOS, NLOS.
Center frequency	900 MHz ISM band
Sampling rate	1.25 MHz
Duration of each input data point	820 μs
Input data point dimensions	1024 by 2
Symbols per input data point	128
Samples per symbol	8
Dataset size per channel type	1.155 million
” ” per modulation	105000
” ” per SNR level	55000
Clock source	Internal clock with 0.1 ppm frequency stability
Antennae type	External Antennae 2.2 dBi Gain, Model No. APAMSTJ-138



(a) (a) LOS Fading

(b) (b) NLOS Fading

Figure 5.2. Photograph of antenna setups used for (a)LOS and (b)NLOS fading.

5.4 Models

CNN and ResNet are commonly used DL models for modulation classification [7, 10, 11, 12, 13, 14, 15]. Specific architectures used are described in Table 5.2. The DL input has dimension 1024 by 2 and the output is the 11 modulation types. Each convolutional block in CNN contains a convolutional layer, batch normalization, pooling and relu activation in sequence. Each residual block in ResNet consists of one convolutional layer, two residual units and a max pooling layer.

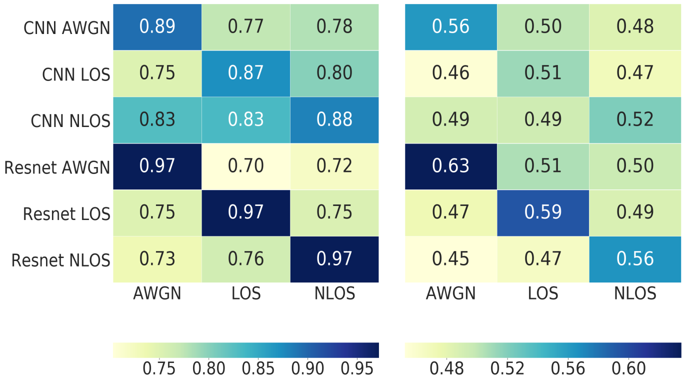
The CNN architecture from [20] which is a variant of the VGG model [22], was chosen since it displayed a test accuracy of 90% on simulated fading channel conditions. The ResNet

Table 5.2. CNN and ResNet architecture

CNN layers	Output dimensions
Input	1024 × 2
Conv	1024 × 16
Maxpool	512 × 16
Conv	512 × 24
Maxpool	256 × 24
Conv	256 × 32
Maxpool	128 × 32
Conv	128 × 48
Maxpool	64 × 48
Conv	64 × 64
Maxpool	32 × 64
Conv	32 × 96
Avgpool	1 × 96
FC/Softmax	1 × 11

ResNet layers	Output dimensions
Input	1024 × 2
Residual Stack	512 × 32
Residual Stack	256 × 32
Residual Stack	128 × 32
Residual Stack	64 × 32
Residual Stack	32 × 32
Residual Stack	16 × 32
FC/SeLU	1 × 128
FC/SeLU	1 × 128
FC/Softmax	1 × 11

architecture is from [15], since it demonstrated test accuracy of 87% on real data under indoor conditions. Further a sanity test of the models was done by training and testing on a simulated rician channel and they displayed test accuracy of 85%. Overall CNN has 98,323 trainable parameters and the ResNet model has 236,344 trainable parameters. The data split for training and testing is 80% and 20%. The open source library PyTorch [23] is used in this work.



(a) High SNR: above −5 dB **(b)** Low SNR: below −5 dB

Figure 5.3. Test accuracies on data of various channel types averaged over all modulation types and SNR levels. Row labels indicate type of trained model and column labels the test data.

5.5 Results and Discussion

The training data is partitioned into three datasets corresponding to the channels AWGN, LOS and NLOS. Each dataset of size 1.155 million, has data points across all modulation types

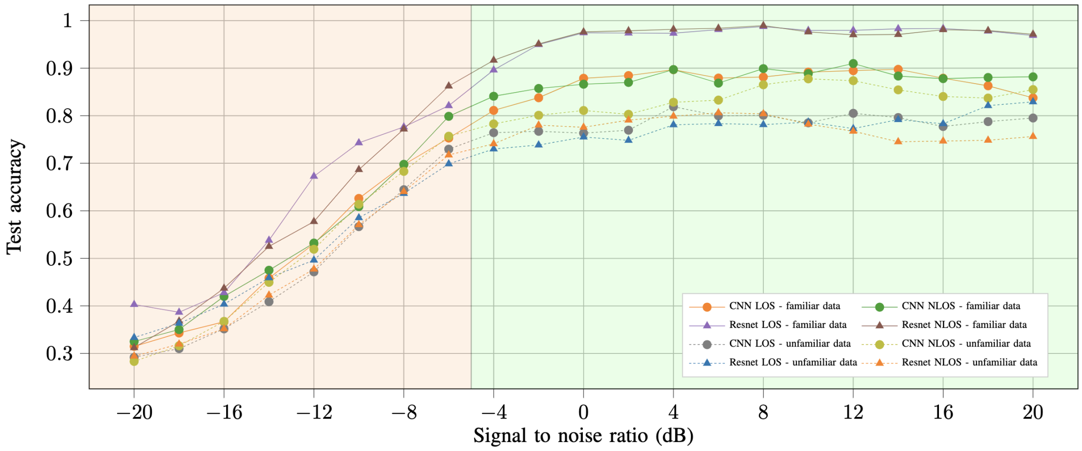


Figure 5.4. Test accuracies on familiar channel data versus unfamiliar channel data across SNR. The two shades regions represent High and Low SNR regimes. Test performance on familiar data (solid lines) and unfamiliar data (dotted lines). CNN model (circular markers) and ResNet model (triangular markers)

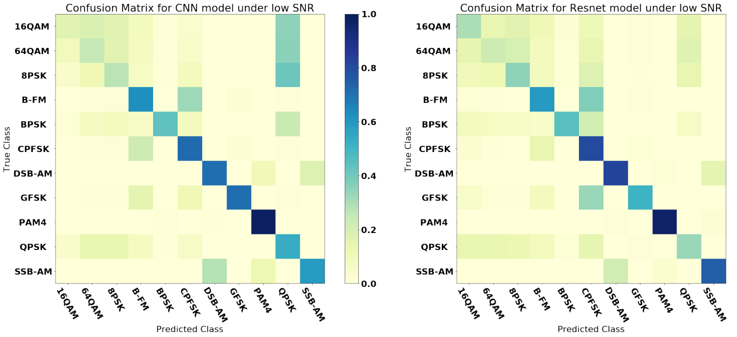
and SNR. CNN and ResNet models are trained on each of these three datasets, thus generating six models. The six models are tested on each of the three datasets independently. For every trained model, one of the three datasets belongs to the same channel conditions as the training data. This dataset will be referred to as *familiar dataset* and the other two as *unfamiliar datasets*. Eg., an AWGN dataset is unfamiliar to a CNN model trained on LOS data. Each of these models are tested on the three datasets and results obtained across modulation types and SNR levels.

The test accuracies on datasets across channel types and two SNR regions, averaged over all modulation types is displayed in Fig. 5.3. The dataset is partitioned into high and low SNR regions at the point of -5 dB SNR, since at values lower than this point the test accuracies drop steeply. The test accuracy is best on familiar dataset for both high and low SNR regions. Also, in wireless communications a LOS fading channel is typically known to corrupt the data more than a pure AWGN channel. Based on difficulty level the channel list would be AWGN, LOS and NLOS. Thus a model trained on LOS fading channel and tested on AWGN is expected to perform better than a model trained on AWGN and tested on LOS. The results observed confirm this expectation at high SNR. For low SNR, CNN test accuracy is in line with this expectation, however ResNet test accuracy numbers are flipped for reasons not apparent.

The test accuracies of models on familiar and unfamiliar data is plotted across SNR levels in Fig. 5.4. The test accuracies are smoothened using a three point moving average filter. The smoothened test accuracies are plotted in Fig. 5.4 and Fig. 5.7. ResNet outperforms CNN at high SNR on familiar data while it under-performs against CNN on unfamiliar data. ResNet has likely overfitted on the training data and therefore under-performs on unfamiliar data. In low SNR, ResNet models outperform CNN on unfamiliar data, contrary to the results in high SNR. This may be attributed to the fact that the datasets are highly corrupted at low SNR and difficult to learn which prevents ResNet from over-fitting at training. Note that the shape of the test accuracy versus SNR is consistent with [15].

As expected, models performed better at classifying data from familiar dataset, see Fig. 5.4. When averaged over all types of trained models, both CNN and ResNet gave a test accuracy of 0.70. Next, models were trained on a cumulative dataset that is an aggregate of AWGN, LOS and NLOS datasets. The average test accuracies of CNN and ResNet on the cumulative dataset was 0.79 and and 0.81 respectively, which is a 10% improvement from 0.70.

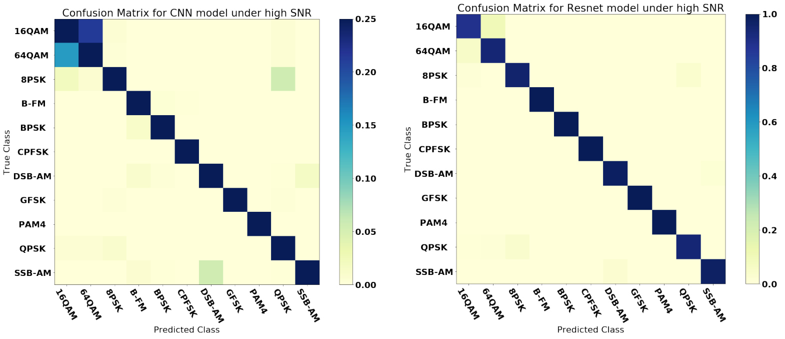
The confusion matrix for CNN and ResNet models trained and tested on the cumulative dataset is illustrated in Fig. 5.5 and 5.6. The maximum value of colorbar is set to 0.25 in Fig. 5.6 to highlight the off-diagonal elements. 16QAM and 64QAM are confused as each other, while other modulation types have low misclassification rate in the high SNR case. In the low SNR case, there is confusion amongst most of the modulation types. Fig. 5.7 provides test accuracies across all SNRs and modulation types. The test accuracy is close to 100% above -2 dB SNR point for most modulation types. ResNet outperforms CNN at low SNR across most types of modulation. It is known that phase sensitive modulation types have lower classification accuracies and higher order schemes within them are more vulnerable [7, 11, 15]. The results indicate that BPSK, QPSK, 8PSK, 16QAM and 64QAM have poor performance, with 16QAM and 64QAM performing the worst. 16QAM and 64QAM are higher order modulation schemes that are tightly packed in the IQ constellation. The presence of HW and channel impairments will spread the points in the constellation and make it really hard to learn.



(a) CNN low SNR

(b) ResNet low SNR

Figure 5.5. CNN and ResNet confusion matrices for low SNR region below −5 dB.



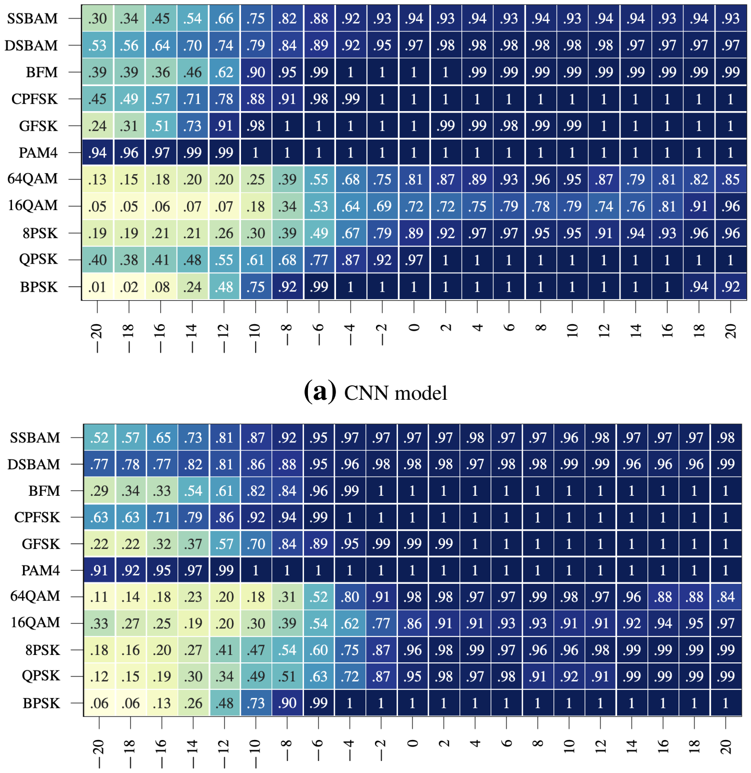
(a) CNN high SNR

(b) ResNet high SNR

Figure 5.6. CNN and ResNet confusion matrices for high SNR region above −5 dB. Maximum value of colorbar set to 0.25 to highlight off-diagonal elements.

5.6 Conclusion

In this work, data belonging to the channel conditions AWGN, LOS and NLOS are collected using software defined radios across a wide set of modulation types and SNR levels. CNN and ResNet models are trained on data belonging to specific channel conditions. Test accuracies are evaluated across model architectures, channel conditions, modulation types and SNR.



	−20	−18	−16	−14	−12	−10	−8	−6	−4	−2	0	2	4	6	8	10	12	14	16	18	20
SSBAM	.30	.34	.45	.54	.66	.75	.82	.88	.92	.93	.94	.93	.94	.93	.94	.93	.94	.94	.93	.94	.93
DSBAM	.53	.56	.64	.70	.74	.79	.84	.89	.92	.95	.97	.98	.98	.98	.98	.98	.98	.97	.97	.97	.97
BFM	.39	.39	.36	.46	.62	.90	.95	.99	1	1	1	1	.99	.99	.99	.99	.99	.99	.99	.99	.99
CPFSK	.45	.49	.57	.71	.78	.88	.91	.98	.99	1	1	1	1	1	1	1	1	1	1	1	1
GFSK	.24	.31	.51	.73	.91	.98	1	1	1	1	1	.99	.99	.98	.99	.99	1	1	1	1	1
PAM4	.94	.96	.97	.99	.99	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
64QAM	.13	.15	.18	.20	.20	.25	.39	.55	.68	.75	.81	.87	.89	.93	.96	.95	.87	.79	.81	.82	.85
16QAM	.05	.05	.06	.07	.07	.18	.34	.53	.64	.69	.72	.72	.75	.79	.78	.79	.74	.76	.81	.91	.96
8PSK	.19	.19	.21	.21	.26	.30	.39	.49	.67	.79	.89	.92	.97	.97	.95	.95	.91	.94	.93	.96	.96
QPSK	.40	.38	.41	.48	.55	.61	.68	.77	.87	.92	.97	1	1	1	1	1	1	1	1	1	1
BPSK	.01	.02	.08	.24	.48	.75	.92	.99	1	1	1	1	1	1	1	1	1	1	1	.94	.92

(a) CNN model

	−20	−18	−16	−14	−12	−10	−8	−6	−4	−2	0	2	4	6	8	10	12	14	16	18	20
SSBAM	.52	.57	.65	.73	.81	.87	.92	.95	.97	.97	.97	.97	.98	.97	.97	.96	.98	.97	.97	.97	.98
DSBAM	.77	.78	.77	.82	.81	.86	.88	.95	.96	.98	.98	.98	.97	.98	.98	.99	.99	.96	.96	.96	.99
BFM	.29	.34	.33	.54	.61	.82	.84	.96	.99	1	1	1	1	1	1	1	1	1	1	1	1
CPFSK	.63	.63	.71	.79	.86	.92	.94	.99	1	1	1	1	1	1	1	1	1	1	1	1	1
GFSK	.22	.22	.32	.37	.57	.70	.84	.89	.95	.99	.99	.99	1	1	1	1	1	1	1	1	1
PAM4	.91	.92	.95	.97	.99	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
64QAM	.11	.14	.18	.23	.20	.18	.31	.52	.80	.91	.98	.98	.97	.97	.99	.98	.97	.96	.88	.88	.84
16QAM	.33	.27	.25	.19	.20	.30	.39	.54	.62	.77	.86	.91	.91	.93	.93	.91	.91	.92	.94	.95	.97
8PSK	.18	.16	.20	.27	.41	.47	.54	.60	.75	.87	.96	.98	.99	.97	.96	.96	.98	.99	.99	.99	.99
QPSK	.12	.15	.19	.30	.34	.49	.51	.63	.72	.87	.95	.98	.97	.98	.91	.92	.91	.99	.99	.99	.99
BPSK	.06	.06	.13	.26	.48	.73	.90	.99	1	1	1	1	1	1	1	1	1	1	1	1	1

(b) ResNet model

Figure 5.7. CNN and ResNet model accuracies trained and tested on cumulative set across modulation type and SNR.

Test accuracies were higher on data belonging to same channel conditions as training data. Also, test accuracies were higher on models tested on data belonging to less challenging conditions than the training data. Models were further trained on cumulative dataset belonging to all channel conditions and the test accuracies increased by 10%. Resnet had better performance at high SNR while CNN had better performance at low SNR. Phase sensitive and tightly packed modulation schemes had the poorest test accuracies.

The work is intended to find ways to improve real time deployment performance of DL models for modulation classification. The results provided empirically establish the following intuitions on real data. The test and training data should belong to the same probability distribution for the best performance. Further, training a model on challenging fading conditions makes it automatically learn patterns of benign channels. Also, caution is needed in using powerful models such as ResNet that can over-learn and perform poorly on data not familiar.

5.7 Acknowledgments

The text of this chapter is in the full reprint of the material as it appears in V. Sathyanarayanan, M. Wagner and P. Gerstoft, ”Over The Air Performance of Deep Learning for Modulation Classification across Channel Conditions”, 2020 54th ASILOMAR Conference on Signals, Systems, and Computers, 2020, pp. 157-161. The dissertation author was the primary researcher and author of this chapter. The co-authors listed in these publications directed and supervised the research.

5.8 References

[1] S. Z. Hsue and S. S. Soliman, “Automatic modulation classification using zero crossing,” in *IEE Proc. F., Radar Signal Process.*, pp. 459–464, 1990.

[2] K. Kim and A. Polydoros, ““digital modulation classification: the bpsk versus qpsk case”,” in *Proc. IEEE MILCOM*, p. 431–436, 1988.

[3] J. E. Hipp, ““modulation classification based on statistical moments”,” in *Proc. IEEE MILCOM*, p. 20.2.1–20.2.6, 1986.

[4] Z. Zhu and A. K. Nandi, “Modulation classification in mimo fading channels via expectation maximization with non-data-aided initialization,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (3014–3018, ed.), 2015.

[5] A. Tsakmalis, S. Chatzinotas, and B. Ottersten, “Modulation and coding classification for adaptive power control in 5g cognitive communications,” in *IEEE Int. Work. on Signal Process. Adv. Wireless Comm.*, p. 234–238, 2014.

[6] H. Sarieddeen, M. M. Mansour, L. M. A. Jalloul, and A. Chehab, “Efficient near optimal joint modulation classification and detection for mu-mimo systems,” in *Proc. IEEE Int. Conf. Acoust.*, p. 3706–3710, 2016.

[7] T. J. O’Shea, J. Corgan, T. C. Clancy, C. Jayne, and L. Iliadis, “Convolutional radio modulation recognition networks,” *IEEE Trans. Neural Netw.*, pp. 213–226, 2016.

[8] M. Zhang, Y. Zeng, Z. Han, and Y. Gong, “Automatic modulation recognition using deep learning architectures,” in *IEEE Int. Work. on Signal Process. Adv. Wireless Comm.*, pp. 1–5, 2018.

[9] Y. Wu, X. Li, and J. Fang, “A deep learning approach for modulation recognition via exploiting temporal correlations,” in *IEEE Int. Work. on Signal Process. Adv. Wireless Comm.*, pp. 1–5, 2018.

[10] B. Kim, J. Kim, H. Chae, D. Yoon, and J. W. Choi, “Deep neural network-based automatic modulation classification technique,” in *Int. Conf. Inf. Commun. Technol. Convergence*, p. 579–582, 2016.

[11] T. J. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, p. 563–575, 2017.

[12] N. E. West and T. J. O’Shea, “Deep architectures for modulation recognition,” in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw.*, p. 1–6, 2017.

[13] K. Karra, S. Kuzdeba, and J. Petersen, “Modulation recognition using hierarchical deep neural networks,” in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw.*, p. 1–3, 2017.

[14] X. Liu, D. Yang, and A. E. Gamal, “Deep neural network architectures for modulation classification,” in *Proc. Asilomar Conf. Signals, Syst., Comput.*, p. 915–919, 2017.

[15] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, p. 168–179, 2018.

[16] C. D. Vrieze, L. Simić, and P. Mähönen, ““the importance of being earnest: Performance of modulation classification for real rf signals,” in *Proc. IEEE Int. Symp.*, pp. 1–5, 2018.

[17] S. Foulke, J. Jagannath, A. Drozd, T. Wimalajeewa, P. K. Varshney, and W. Su, “Multisensor modulation classification (mmc): Implementation considerations- usrp case study,” in *IEEE MILCOM*, p. 1663–1668, 2014.

[18] J. Jagannath, N. Polosky, N. Polosky, D. O’Connor, L. N. Theagarajan, B. Sheaffer, S. Foulke, and P. K. Varshney, “Artificial neural network based automatic modulation classification over a software defined radio testbed,” in *Proc. IEEE Int. Conf. Commun.*, pp. 1–6, 2018.

[19] Z. L. Tang, S. M. Li, and L. J. Yu, “Implementation of deep learning based automatic modulation classifier on fpga sdr platform,” *Electron*, vol. 7, no. 7, 2018.

[20] MATHWORKS, “Modulation Classification with Deep Learning..” https://www.mathworks.com/help/deeplearning/ug/modulation-classification-with-deep-learning.html. accessed June 12, 2019.

[21] V. Sathyanarayanan, “Real dataset across channel conditions, modulation types and SNR levels..” https://drive.google.com/drive/folders/1Uo69wuyeltpF9BH4vwhid9-q0mSEVU-5?usp=sharing.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv*, vol. 1, no. 1409.1556, pp. 1–10, 2014.

[23] A. Paszke, S. Gross, and F. Massa, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035, 2019.

Chapter 6

Conclusion

The thesis focuses on two pivotal cogs of spectrum sensing: modulation classification, and characterization of of emanations. In the first paradigm, the focus is to improve DL-based modulation classification performance using a hybrid approach. The domain knowledge from the fields of wireless systems is leveraged to improve performance. The second paradigm focuses on the characterization of emanations that are used to detect anomalous activities.

6.1 Modulation classification

6.1.1 Over the Air Performance of Deep Learning for Modulation Classification Across Channel conditions

Channel artifacts significantly affect performance in any wireless system. In the first work under modulation classification, the impact of channel conditions within the context of training and test data disparity is studied. This is also one of the first efforts in attempting modulation classification using DL on real OTA data for a wide set of modulation types and SNR. CNN and ResNet architecture models are trained and tested on OTA data belonging to channel conditions AWGN, LOS, and NLOS. The trained models are tested on each of the three datasets independently. For every trained model, one of the three datasets belongs to the same channel conditions as the training data. This dataset will be referred to as a familiar dataset and the other two as unfamiliar datasets. Eg., an AWGN dataset is unfamiliar to a CNN model trained on LOS data.

The dataset is partitioned into high and low SNR regions at the point of -5 dB SNR, since at values lower than this point the test accuracies drop steeply. The test accuracy is best on familiar datasets for both high and low SNR regions. Also, in wireless communications a LOS fading channel is typically known to corrupt the data more than a pure AWGN channel. Based on increasing difficulty levels the channel list would be AWGN, LOS, and NLOS. Thus a model trained on LOS fading channel and tested on AWGN is expected to perform better than a model trained on AWGN and tested on LOS. The results observed confirm this expectation at high SNR. For low SNR, CNN test accuracy is in line with this expectation, however, ResNet test accuracy numbers are flipped for reasons not apparent. Further, ResNet outperforms CNN at high SNR on familiar data while it under-performs against CNN on unfamiliar data. ResNet has likely overfitted on the training data and therefore under-performs on unfamiliar data. In low SNR, ResNet models outperform CNN on unfamiliar data, contrary to the results in high SNR. This may be attributed to the fact that the datasets are highly corrupted at low SNR and difficult to learn which prevents ResNet from over-fitting at training.

Overall, test accuracies are higher on data belonging to the same channel conditions as training data. Thus, efforts should be taken to use training data whose channel conditions are similar to the test data. Also, test accuracies are higher on models tested on data belonging to less challenging conditions compared to training data. Thus training a model on challenging fading conditions makes it automatically learn patterns of data belonging to benign channel conditions. Thermal noise is typically modeled as added at the receiver. Accurate noise floor estimation at receiver, and calibration of TX and RX power are needed to achieve this in real data collection. This is non-trivial and therefore in this work thermal noise is added synthetically before transmission. In the next work, this problem is addressed.

6.1.2 Novel Training Methodology to Enhance Deep Learning Based Modulation Classification

In this second work, signal processing advances in blind SNR estimation are leveraged to improve modulation classification performance. Considering that SNR significantly affects the performance of wireless systems, a divide and conquer approach is proposed where models learn to classify on a subset of SNR conditions. In this novel training methodology, each model is trained on data belonging to specific subsets based on SNR. At inference, signals belonging to specific SNR subset are passed through the appropriate model for inference. OTA data belonging to AWGN and NLOS channels, is captured using USRP SDR across a wide range of modulation and SNR are used in this work.

For results comparison, CNN models are trained on each subset and complete set of data belonging to both channel types. SNR range from -20 to 0 dB form one subset. SNR range from 0 to 20 dB is further split into four equispaced subsets. The choice of SNR partitioning is a hyperparameter. The SNR subset -20 to 0 dB has least improvement in accuracy. It is hard for a classifier to learn at low SNR, likely placing a cap on the performance benefits that could be obtained. Thus even with the SNR partitioning approach, we do not see improvements in negative SNR regions. Subset 0 to 5 dB has the best improvement in accuracy. The overall improvements in accuracy are higher for AWGN than NLOS. This is likely because the NLOS dataset is collected in the harsher multipath channel and is more difficult to learn. Overall, we see improvements in test accuracy on all subsets across both channels.

The cost paid for performance improvements is extra computation both at training and inference. This is because we train on five classifiers and use the SNR estimation algorithm at inference, compared to a single model at training and testing. Overall, a cumulative 6% performance improvement is shown by using the novel training method.

6.1.3 RML22: Realistic Dataset Generation for Wireless Modulation Classification

For the final work under modulation classification, a data-centric approach is taken to solve the modulation classification problem. We identified errors and ad-hoc choice of parameters in RML16 which is the current state of art synthetic dataset. Building upon RML16, a realistic and correct methodology for generating a dataset is provided. The performance of the proposed benchmark dataset RML22 is compared with RML16. Further, the performance impact of artifact and signal model parameterization is studied as follows: choice of information source, parameterization of clock and Doppler effects, sps, and signal bandwidth.

Models are trained on RML16 and RML22 and tested on the five datasets. Models trained on RML22 and RML16 have accuracies of 0.67 and 0.44 when tested on RML22. RML22 outperforms RML16 by 0.23 because the shortcomings of RML16 are corrected towards generating RML22. RML16 accuracy of analog modulation types WBFM and AM-DSB are lower than RML22 by 0.65, due to errors in analog information source generation. GFSK modulation has 0 accuracy for RML16 compared to 1.0 for RML22, due to incorrect modulation index. Phase-sensitive modulation types BPSK, QPSK, 8PSK, 16QAM, and 64QAM have 0.06 higher accuracy for RML22 over RML16, likely due to incorrect clock effects.

Further performance of models is studied in the context of generalization for different instantiation of the following parameters. For clock effects, different HWs have varying clock effects based on their XO crystal quality. A model trained with a maximum LO frequency deviation of 500 Hz performs best. This is because a maximum LO frequency deviation of 500 Hz also contains 50 Hz values. To study Doppler frequency parameterization, a slow pedestrian and fast vehicle equivalent Doppler of 1 and 70 Hz are used. The model trained on a lower Doppler of 1 Hz performs poorer. Doppler causes a rotational effect similar to CFO. In line with the discussion for the previous case, Doppler of 70 Hz contains the effect of Doppler 1 Hz and therefore has better performance. Sps of a frame is chosen based on a combination of radio frequency card bandwidth, ADC sample rate, and computational resources. Sps parameterization values of 2 and 8 are chosen, corresponding to RML22 and RML16. The model trained on sps = 2 acts as a random classifier on dataset sps = 8 and vice-versa. This is because the model expects a symbol for every N number of samples in its input. In deployment scenarios, models are pre-trained with large synthetic datasets and updated with real OTA datasets via transfer learning. Thus, it is essential both datasets have the same sps.

Data is generated for a single carrier, the SISO signal model. It is of interest to expand to multicarrier as encountered in OFDM-based technologies such as LTE, 5G, and also MIMO. Several HW artifacts are not considered in data generation such as IQ imbalance, local oscillator (LO) leakage causing spur at DC, power amplifier (PA) non-linearities, etc. It will be of interest to test the performance impact of these artifacts. Further, current work does not involve testing models trained on RML16, and RML22, upon real data. This should be done to confirm that RML22 is indeed a realistic corrected dataset.

6.2 Anomalous Activity Detection using RF Emanations

This work provides a profiling-free HW-agnostic technique for detecting emanations. We introduce detection of the harmonics of clock leakages as a generic signature of emanations. The pitch of the harmonic was used to characterize the emanation detected. The harmonic signature undergoes unintended modulation that is characteristic of the physics of the HW, and unknown to us. This is captured in the transmit emanation model. The transmitted emanation signal undergoes HW and channel artifacts, captured in the receive emanation model.

A pre-processing technique is introduced to remove modulation and artifacts. Derivations are provided to show how preprocessing removes the effect of modulation, channel, and clock artifacts and helps extract the harmonics. These derivations are extended for multi-harmonics cases and when overt signals as interferer are present. Further, Welch-based PSD estimation is done to reduce the variance and improve SNR gain. In the PSD, dominant peaks are found using

SNR and prominence metrics. The peaks with SNR exceeding a given threshold are picked. They are pruned further using the prominence metric. A robust percentile-based approach is used to estimate the noise floor and threshold, which does not assume a specific model for signal peaks. The frequency and SNR of dominant peaks are used to find the pitch frequency. There could be one or more harmonic series. The intention is to detect each of them and estimate the corresponding pitch. The pitch estimation procedure from audio processing fields is improved upon to detect multiple harmonic series, for the wireless signals in this work.

Backing the theoretical exposition of the algorithm, the performance of the algorithm is shown on real IQ data. IQ data is collected in bandwidth from 0.1–1.1 GHz in a shielded room. Damaged electronic peripherals are emulated by exposing cables of the mouse, and keyboard, and data transfer is emulated by copying data onto an SD card and pen drive. Emanation patterns across the 1 GHz bandwidth are different for both use cases compared to a baseline of an idle laptop. Thus we show the detection of anomalous activity using emanations, in a profiling-free manner without HW knowledge. Anomalous activities are shown to be detected using the emanation patterns.

As a future step, a study needs to be done to improve multi-pitch estimation performance. In real OTA data, harmonics of dominant pitches sometimes make detection of weaker pitches difficult. Further, SNR gains to be had with longer IQ capture should be explored. Emanation detection needs to be done from an exhaustive set of HW devices to check the performance of the algorithm. Further, overt signals could be introduced and algorithm performance checked in their presence practically.