# Adversarial RFML: Evading Deep Learning Enabled Signal Classification

Bryse Austin Flowers

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

Richard M. Buehrer, Chair

William C. Headley, Co-Chair

Guoqiang Yu

Ryan M. Gerdes

June 6, 2019

Blacksburg, Virginia

Keywords: Adversarial Signal Processing, Cognitive Radio Security, Machine Learning,

Modulation Identification, Radio Frequency Machine Learning

# Adversarial RFML: Evading Deep Learning Enabled Signal Classification

Bryse Austin Flowers

## ABSTRACT

Deep learning has become an ubiquitous part of research in all fields, including wireless communications. Researchers have shown the ability to leverage deep neural networks (DNNs) that operate on raw in-phase and quadrature samples, termed Radio Frequency Machine Learning (RFML), to synthesize new waveforms, control radio resources, as well as detect and classify signals. While there are numerous advantages to RFML, this thesis answers the question "is it secure?" DNNs have been shown, in other applications such as Computer Vision (CV), to be vulnerable to what are known as adversarial evasion attacks, which consist of corrupting an underlying example with a small, intelligently crafted, perturbation that causes a DNN to misclassify the example. This thesis develops the first threat model that encompasses the unique adversarial goals and capabilities that are present in RFML. Attacks that occur with direct digital access to the RFML classifier are differentiated from physical attacks that must propagate over-the-air (OTA) and are thus subject to impairments due to the wireless channel or inaccuracies in the signal detection stage. This thesis first finds that RFML systems are vulnerable to current adversarial evasion attacks using the well known Fast Gradient Sign Method originally developed for CV applications. However, these current adversarial evasion attacks do not account for the underlying communications and therefore the adversarial advantage is limited because the signal quickly becomes unintelligible. In order to envision new threats, this thesis goes on to develop a new adversarial evasion attack that takes into account the underlying communications and wireless channel models in order to create adversarial evasion attacks with more intelligible underlying communications that generalize to OTA attacks.

# Adversarial RFML: Evading Deep Learning Enabled Signal Classification

Bryse Austin Flowers

## GENERAL AUDIENCE ABSTRACT

Deep learning is beginning to permeate many commercial products and is being included in prototypes for next generation wireless communications devices. This technology can provide huge breakthroughs in autonomy; however, it is not sufficient to study the effectiveness of deep learning in an idealized laboratory environment, the real world is often harsh and/or adversarial. Therefore, it is important to know how, and when, these deep learning enabled devices will fail in the presence of bad actors before they are deployed in high risk environments, such as battlefields or connected autonomous vehicle communications. This thesis studies a small subset of the security vulnerabilities of deep learning enabled wireless communications devices by attempting to evade deep learning enabled signal classification by an eavesdropper while maintaining effective wireless communications with a cooperative receiver. The primary goal of this thesis is to define the threats to, and identify the current vulnerabilities of, deep learning enabled signal classification systems, because a system can only be secured once its vulnerabilities are known.

# Contents

**1 Introduction and Motivation** **2**

1.1 Research Contributions . . . . . 5

1.2 Thesis Outline . . . . . 6

1.3 Publications . . . . . 10

**2 Background** **11**

2.1 Wireless Physical Layer . . . . . 12

2.2 Cognitive Radio . . . . . 14

2.2.1 Key Enablers of Cognitive Radio . . . . . 15

2.2.1.1 Software Defined Radio . . . . . 15

2.2.1.2 Radio Frequency Machine Learning . . . . . 16

2.2.1.3 Spectrum Sensing: Blind Signal Classification . . . . . 18

2.2.2 Competition in Ad Hoc Networks . . . . . 19

2.3 Threats to RFML Signal Classification . . . . . 20

2.3.1 Adversarial Machine Learning . . . . . 21

2.3.1.1 Privacy Attacks . . . . . 21

2.3.1.2 Causative Attacks . . . . . 23

2.3.1.3 Evasion Attacks . . . . . 24

2.4 Related Work . . . . . 24

**3 Attack Evaluation Methodology** **28**

3.1 Automatic Modulation Classification System Model . . . . . 29

3.2 Threat Model . . . . . 30

3.2.1 Adversarial Goals . . . . . 31

3.2.1.1 Direct Access . . . . . 34

3.2.1.2 Self Protect . . . . . 34

3.2.1.3 Cover . . . . . 36

3.2.2 Adversarial Capabilities . . . . . 38

3.2.3 Threat Model Assumed in the Current Work . . . . . 39

3.3 AMC Target Network . . . . . 41

3.3.1 Network Architecture . . . . . 41

3.3.2 Dataset A . . . . . 43

3.3.3 Dataset B . . . . . 43

3.3.4 Training Results . . . . . 46

**4 Direct Access Evasion Attacks** **51**

4.1 Introduction to Adversarial Machine Learning . . . . . 52

4.2 Adapting FGSM . . . . . 55

4.3 Baseline Evaluation . . . . . 57

4.4 Attack Effectiveness versus NN Input Size . . . . . 58

4.5 Analyzing Individual Adversarial Examples . . . . . 60

4.5.1 Difference in Logits . . . . . 61

4.5.2 Classifier Output versus Attack Intensity . . . . . 62

4.5.3 Mutation Testing with AWGN . . . . . 64

4.6 Conclusion . . . . . 69

**5 Self Protect Evasion Attacks** **70**

5.1 Simulation Environment . . . . . 72

5.1.1 Modulation . . . 73
5.1.2 Adversarial ML . . . 73
5.1.3 Channel Model . . . 74
5.1.4 Demodulation . . . 75
5.1.5 Automatic Modulation Classification Evaluation . . . 75
5.2 Impact of Additive White Gaussian Noise . . . 76
5.3 Impact of Center Frequency Offsets . . . 83
5.4 Impact of Timing Offsets . . . 86
5.5 Conclusion . . . 89

**6 Communications Aware Evasion Attacks** **91**

6.1 System Model . . . 93
6.1.1 Transmitter . . . 93
6.1.2 Receiver . . . 95
6.1.3 Eavesdropper . . . 95
6.1.4 Threat Model . . . 95
6.2 Methodology . . . 96
6.2.1 Loss Function . . . 97

6.2.1.1 Adversarial Loss . . . . . 98
6.2.1.2 Communications Hinge Loss . . . . . 99
6.2.1.3 Perturbation Power Regularization . . . . . 101
6.2.2 Training Implementation . . . . . 102
6.2.2.1 Transmitter . . . . . 102
6.2.2.2 Root Raised Cosine Filter . . . . . 102
6.2.2.3 Combining Signals . . . . . 104
6.2.2.4 Receiver . . . . . 104
6.2.2.5 Eavesdropper Channel and Signal Isolation Model . . . . . 105
6.2.2.6 Training Procedure . . . . . 106
6.2.3 Adversarial Residual Network Architecture . . . . . 106
6.2.4 Automatic Modulation Classification . . . . . 107
6.2.5 Evaluation Procedure . . . . . 107
6.3 Results . . . . . 108
6.3.1 Trading Off Communication for Deception . . . . . 108
6.3.2 Evading Modulation Recognition . . . . . 110
6.3.3 Maintaining the Communications Link . . . . . 112

6.3.4 Adversarial Spectrum Usage . . . . . 117

6.4 Conclusion . . . . . 119

**7 Conclusion** **121**

7.1 Current RFML Vulnerabilities . . . . . 123

7.2 Hardening RFML . . . . . 126

7.3 Limitations of the Current Work and Suggested Future Directions . . . . . 127

**Bibliography** **130**

# List of Figures

2.1 Overview of a digital communications transmitter/receiver pair. . . . . 13

2.2 Overview of a Radio Frequency Machine Learning (RFML) signal classification system (eavesdropper device). The eavesdropper could use RFML signal classification for traffic recognition [62], modulation recognition [16], or specific emitter identification [13, 24]. . . . . 19

2.3 The current work is only concerned with attacking the signal classification subsystem. This is studied using modulation recognition as a reference signal classification task. . . . . 21

2.4 Threat Surface of a RFML signal classification system presenting an overview of where attacks could be launched from. . . . . 22

3.1 Threat Model for evading RFML signal classification systems. . . . . 32

3.2 Example confidence outputs of a model for confidence reduction, untargeted misclassification, and targeted misclassification. In all plots, the original example belongs to Class A and is classified correctly. The output of the classifier on the "initial" example is shown in blue, while, the output of the classifier on the adversarial example, or the output when "attacked", is shown in orange. 33

3.3 Overview of locations an adversarial evasion attack could be launched from. Direct access attacks are considered in Chapter 4 while Chapters 5 and 6 discuss self protect attacks . . . . . 35

3.4 Convolutional Neural Network Architecture, first introduced in [8] and modified according to [9, 48], used in the current work for AMC. . . . . 42

3.5 Random samples from Frequency Shift Keying (FSK) and Analog Modulations in Dataset A. The Signal-to-Noise Ratio (SNR) was restricted to 18 dB but the specific examples in time were selected randomly from that subset. The frequency content is averaged across all examples that have 18 dB SNR and the corresponding modulation. . . . . 44

3.6 Random samples from Linear Digital Amplitude Phase Modulation (LDAPM) Modulations in Dataset A. The SNR was restricted to 18 dB but the specific examples in time were selected randomly from that subset. The frequency content is averaged across all examples that have 18 dB SNR and the corresponding modulation. . . . . 45

3.7 Random samples from the 128 sample version of Dataset B. The $E_s/N_0$ was restricted to 20 dB but the specific examples in time were selected randomly from that subset. The center frequency offset is shown by averaging the frequency content across all samples with $-1\%$ offset and 1% offset. . . . . 47

3.8 Dataset A test accuracy vs SNR. . . . . 49

3.9 Dataset B test accuracy vs SNR for three different Deep Neural Networks (DNNs) input sizes. In this Figure, there are no center frequency offsets during evaluation. As expected, increasing the input size results in increasing test accuracy over the entire SNR range studied. . . . . 50

4.1 BPSK adversarial example with a 10 dB ($E_s/E_j$) perturbation, created with the Fast Gradient Sign Method (FGSM) [39] algorithm, applied. . . . . 54

4.2 Classification accuracy of a model trained on Dataset A for a direct access attack. This plot compares the average classification accuracy for BPSK, QPSK, 8PSK, QAM16, and QAM64 when FGSM in used to apply a specific adversarial perturbation to the accuracy when "jammed" with a Gaussian noise signal at the same power ratio. . . . . 58

4.3 (top) Overall classification accuracy of models trained on Dataset B in the presence of a direct access FGSM attack for different input sizes. (bottom) The relative classification accuracy ranking of the three different models for each $E_s/E_j$. . . . . 59

4.4 Output of the model trained on Dataset A for a direct access FGSM attack using a single, randomly selected, BPSK adversarial example across varying $E_s/E_j$ (top) and the corresponding difference in logits (bottom). The areas shaded red represent regions where a correct classification occurred (therefore the adversary was unsuccessful) while the areas shaded green represent an incorrect classification (therefore the adversary was successful). Note that the regions are only shaded to visualize (4.12). . . . . 63

4.5 Output of the model trained on Dataset A for a direct access FGSM attack using a single, randomly selected, QAM16 adversarial example across varying $E_s/E_j$ (top) and the corresponding difference in logits (bottom). The areas shaded red represent regions where a correct classification occurred (therefore the adversary was unsuccessful) while the areas shaded green represent an incorrect classification (therefore the adversary was successful). Although it is a low confidence prediction, the classification is narrowly correct when $E_s/E_j > 35$(dB) and narrowly incorrect when $E_s/E_j < 5$(dB). With the model having trouble distinguishing between QAM16 and QAM64. Note that the regions are only shaded to visualize (4.12). . . . . 65

4.6 The effect of noise on the output of the model trained on Dataset A for a single, randomly selected, BPSK adversarial example with an $E_s/E_j$ of 10 dB. The line represents the mean of the difference in logits, at a specific $E_s/N_0$, while the shaded region represents the 25th and 75th percentiles in order to show the variance of the output. . . . . 67

4.7 The effect of noise on the output of the model trained on Dataset A for a single, randomly selected, QPSK adversarial example with an $E_s/E_j$ of 10 dB. The line represents the mean of the difference in logits, at a specific $E_s/N_0$, while the shaded region represents the 25th and 75th percentiles in order to show the variance of the output. . . . . 68

5.1 Block diagram of the evaluation methodology developed for the current work. The current work assumes perfect knowledge of the target DNNs and therefore the DNNs shown in the Automatic Modulation Classification (AMC) Evaluation and Adversarial Machine Learning (ML) blocks are identical and simply separated for clarity. . . . . 72

5.2 Classification accuracy and Bit Error Rate (BER) at varying $E_s/E_j$ and $E_s/N_0$ for self protect untargeted adversarial attacks using FGSM on the model trained with Dataset A and a source modulation class of BPSK. Note that when using a high powered perturbation, where $E_s/E_j$ is a small value, not all BERs are attainable in the SNR range studied. In all plots, the adversary desires to have lower curves because lower BER implies a lower impact to the underlying communication and a lower accuracy implies untargeted adversarial success. Conversely, the eavesdropper desires to have higher curves which indicate a larger increase in BER for an adversary to evade signal classification. 78

5.3 Classification accuracy and BER at varying $E_s/E_j$ and $E_s/N_0$ for self protect untargeted adversarial attacks using FGSM on the model trained with Dataset A and a source modulation class of QPSK. Note that when using a high powered perturbation, where $E_s/E_j$ is a small value, not all BERs are attainable in the SNR range studied. In all plots, the adversary desires to have lower curves because lower BER implies a lower impact to the underlying communication and a lower accuracy implies untargeted adversarial success. Conversely, the eavesdropper desires to have higher curves which indicate a larger increase in BER for an adversary to evade signal classification. . . . . 79

5.4 Classification accuracy and bit error rates at varying $E_s/E_j$ and $E_s/N_0$ for self protect untargeted adversarial attacks using FGSM on the model trained with Dataset A and a source modulation class of 8PSK. Note that when using a high powered perturbation, where $E_s/E_j$ is a small value, not all BERs are attainable in the SNR range studied. In all plots, the adversary desires to have lower curves because lower BER implies a lower impact to the underlying communication and a lower accuracy implies untargeted adversarial success. Conversely, the eavesdropper desires to have higher curves which indicate a larger increase in BER for an adversary to evade signal classification. . . . . 81

5.5 Classification accuracy and bit error rates at varying $E_s/E_j$ and $E_s/N_0$ for self protect untargeted adversarial attacks using FGSM on the model trained with Dataset A and a source modulation class of QAM16. Note that when using a high powered perturbation, where $E_s/E_j$ is a small value, not all BERs are attainable in the SNR range studied. In all plots, the adversary desires to have lower curves because lower BER implies a lower impact to the underlying communication and a lower accuracy implies untargeted adversarial success. Conversely, the eavesdropper desires to have higher curves which indicate a larger increase in BER for an adversary to evade signal classification. . . . . 82

5.6 Classification accuracy vs normalized center frequency offset at varying $E_s/E_j$ for self protect untargeted adversarial attacks using FGSM. The model used is trained on Dataset B with an input size of 128. This dataset has a training distribution of $\pm 1\%$ frequency offset that has been normalized to the sample rate. . . . . 85

5.7 Classification accuracy vs time window offsets at varying $E_s/E_j$ for self protect untargeted adversarial attacks using FGSM. The model used is trained on Dataset A. . . . . 88

6.1 Overview of the system model assumed in the current chapter where the contributions are encapsulated in the "adversarial network" block. . . . . 93

6.2 Training procedure used in the current work. All elements are implemented in PyTorch. The forward pass, or the "signals", are shown in black. The backward pass, or the "gradient of the loss", is shown in red. In order to extract the symbols, $S_{tx+j}$, the transmitted signal is match filtered and then downsampled to achieve one sample per symbol. The symbol error indicator, $I_s$, is created by applying Additive White Gaussian Noise (AWGN) to the signal before performing the same filtering and downsampling process to extract the received symbols, $S_{rx}$, which can then be compared with the transmitted symbols, $S_{tx}$, to compute $I_s$. The signal used for AMC is also passed through an AWGN channel but the power of the noise, $\sigma^2_{\text{adv}}$, is varied independently of the power of the noise at the receiver, $\sigma^2_{\text{rx}}$. When creating discrete adversarial examples to pass through the AMC model, the starting index is varied uniformly to ensure that adversarial success does not depend on time synchronization. . . . . 103

6.3 Mean communications and adversarial loss values of the best epoch for multiple trained Adversarial Residual Network (ARN)s with a source modulation of 8PSK. The optimal location would be the upper left of the plot. . . . . 109

6.4 Mean training loss per batch, before scaling with $\alpha$ and $\beta$, for an ARN being trained on a source modulation of 8PSK, $\alpha$ parameter of 50%, and an $E_s/E_j$ limit of 5 dB. At the beginning of training, the ARN trades off communication's ability (orange) for deceiving the eavesdropper (blue). For reference, horizontal dashed lines are added to represent the mean classification confidence in the true class that would produce a specific adversarial loss (blue). . 111

6.5 (top) BER for adversarial attacks at differing $E_s/E_j$ for a source modulation of BPSK as well as the theoretical BER for comparison. (bottom) Classification accuracy for the same adversarial attacks with the baseline classification accuracy, which would occur when no adversarial machine learning is performed, shown. The classification accuracy is presented with uniformly distributed time offsets. . . . . 113

6.6 (top) BER for adversarial attacks at differing $E_s/E_j$ for a source modulation of QPSK as well as the theoretical BER for comparison. (bottom) Classification accuracy for the same adversarial attacks with the baseline classification accuracy, which would occur when no adversarial machine learning is performed, shown. The classification accuracy is presented with uniformly distributed time offsets. . . . . 114

6.7 (top) BER for adversarial attacks at differing $E_s/E_j$ for a source modulation of 8PSK as well as the theoretical BER for comparison. (bottom) Classification accuracy for the same adversarial attacks with the baseline classification accuracy, which would occur when no adversarial machine learning is performed, shown. The classification accuracy is presented with uniformly distributed time offsets. . . . . 115

6.8 (top) BER for adversarial attacks at differing $E_s/E_j$ for a source modulation of QAM16 as well as the theoretical BER for comparison. (bottom) Classification accuracy for the same adversarial attacks with the baseline classification accuracy, which would occur when no adversarial machine learning is performed, shown. The classification accuracy is presented with uniformly distributed time offsets. . . . . 116

6.9 Normalized power spectral density for a 8PSK source modulation with a perturbation crafted by an ARN with an $\alpha$ of 0.5 and $E_s/E_j$ limit of 0 dB. (left) Frequency content of each signal separately. (right) Frequency content of the combination of both signals, which represents what would be transmitted. . . 118

# List of Tables

3.1 Summary of the pros and cons of adversarial evasion attacks from each location. 37

# Abbreviations

**AAE** Adversarial Auto-Encoder

**ADC** Analog to Digital Converter

**AMC** Automatic Modulation Classification

**API** Application Programmer Interface

**ARN** Adversarial Residual Network

**ATN** Adversarial Transformation Network

**AWGN** Additive White Gaussian Noise

**BER** Bit Error Rate

**CR** Cognitive Radio

**CV** Computer Vision

**DAC** Digital to Analog Converter

**DARPA** Defense Advanced Research Projects Agency

**DNNs** Deep Neural Networks

**DQN** Deep Q Network

**DSA** Dynamic Spectrum Access

**DSP** Digital Signal Processing

**EW** Electronic Warfare

**FGSM** Fast Gradient Sign Method

**FSK** Frequency Shift Keying

**IQ** In-Phase and Quadrature

**LDAPM** Linear Digital Amplitude Phase Modulation

**ML** Machine Learning

**OTA** Over-the-Air

**P-ATN** Perturbation - Adversarial Transformation Network

**RF** Radio Frequency

**RFFE** Radio Frequency Front End

**RFML** Radio Frequency Machine Learning

**SDR** Software Defined Radio

**SNR** Signal-to-Noise Ratio

**SWaP** Size, Weight, and Power

**UAP** Universal Adversarial Perturbation

**USRP** Universal Software Radio Peripheral

# Chapter 1

# Introduction and Motivation

Always-on connectivity and high data rates have become ubiquitous in most modern countries. This powers many applications such as social media, Smart Home devices, and connected vehicles. However, network usage is still rapidly expanding with mobile traffic expected to increase seven-fold (from 2017) by 2022 and, at that time, connected devices will outnumber humans 1.5 : 1 [5]. While the usage increases, the wireless spectrum remains a finite natural resource. Therefore, we must invent more intelligent ways to efficiently use this resource.

Growing from the success of Deep Neural Networks (DNNs) in other applications, such as Computer Vision (CV), deep learning now permeates nearly all facets of wireless communications research. It has been applied to create new wireless waveforms [6], control radio resources [7], and aid in spectrum sensing through signal classification [8–13]. The combination of all these applications can then be used in the development of Cognitive Radio (CR) [14, 15] in order to build intelligent radios that opportunistically access the spectrum in a manner that is both efficient and creates highly reliable communications.

These applications, in particular signal classification, have largely been studied before. However, previous iterations of this technology were likelihood or feature based [16–21], while more recent approaches leverage the advances in DNNs to operate directly on raw In-Phase and Quadrature (IQ) samples [8–13]. The current work uses the term Radio Frequency Machine Learning (RFML) to succinctly describe these systems based on DNNs due to the Defense Advanced Research Projects Agency (DARPA) program by that same name [22]. While prior approaches were application specific, because of their need to infuse human

knowledge via expert features or statistics, RFML offers a potentially future-proof technology by being able to autonomously adapt to new types of signals and radio environments when provided with training data. Further, this autonomy removes the need for a human in-the-loop and thus a RFML system could adapt to new situations in the field or learn from its experiences, a key goal of CR [15] and the DARPA RFML systems program [22]. Yet, before a system of this nature should be realized as a deployed system, we must first answer the question "are these systems secure?" This thesis seeks to answer that question, specifically for RFML signal classification.

DNNs have been shown, in other applications such as CV, to be vulnerable to what are known as adversarial examples. Adversarial examples are small, imperceptible to humans, perturbations that are intelligently crafted and applied to the input of DNNs to cause a misclassification at inference time. The crafting of these adversarial examples is termed adversarial machine learning, specifically, an adversarial evasion attack. Adversarial machine learning could be used, in the context of RFML, to disrupt Dynamic Spectrum Access (DSA) systems through primary user emulation [23], evade mobile transmitter tracking [24], or avoid demodulation by confusing an Automatic Modulation Classification (AMC) system [16]. Although this type of vulnerability, and defenses to counter it, have been extensively studied in CV [25–42], this vulnerability is only beginning to be studied in the context of RFML [43–47].

While RFML research on adversarial machine learning evasion attacks and defenses can build off of the large body of literature present in the CV domain, RFML has additional

adversarial goals and capabilities beyond those typically considered in CV. Adversarial goals must be split between attacks that can perturb the signal digitally, directly at the eavesdropper and physical attacks that can only perturb the signal before transmission Over-the-Air (OTA). While attacks with direct access to the eavesdropper's classification network are able to inject pristine perturbations, due to their digital access to the classifier's input, physical attacks are impaired by all of the common sources of noise in a RFML system such as thermal noise, multi-path propagation, and signal detection effects [48] which can all impair an adversary's ability to evade classification. Additionally, in the context of wireless communications, attacks must be characterized against the primary metric of interest, Bit Error Rate (BER). An adversary may seek to evade an eavesdropping classifier but that is of limited benefit if it also corrupts the transmission to a cooperative receiver.

## 1.1 Research Contributions

This thesis makes the following contributions:

- Chapter 3 and [1] consolidate the additional adversarial goals and capabilities present in RFML and proposes a new threat model for evaluating adversarial evasion attacks in the context of wireless communications.

- Chapters 4 and 5, as well as [1], then presents results outlining the vulnerabilities of RFML systems to current adversarial machine learning methodologies using the well known Fast Gradient Sign Method (FGSM) attack [39]. Chapter 4 shows the

vulnerabilities of RFML systems to adversarial machine learning when the adversary has direct access to the classifier while Chapter 5 shows their vulnerabilities to OTA adversarial machine learning attacks.

- In order to envision future threats to RFML, the current work presents a communications aware attack in Chapter 6 (and in [2]) which considers both BER and receiver effects in its training procedure. This attack creates perturbations which generalize over receiver effects, have lower impact to the underlying communication, and do not rely on gradient computation during operation which would allow for higher data rates.

## 1.2 Thesis Outline

Chapter 2 presents the background information on the wireless physical layer that is necessary to understand the evaluation methodology of Chapter 5 and attack methodology of Chapter 6. It then introduces CR, defines it, and outlines the key technologies that enable its realization, as well as additional reasons why these systems would be deceived. Chapter 2 then focuses on RFML signal classification and outlines the threats that adversarial machine learning poses to these systems. While the current work is primarily focused on evasion attacks, RFML would also be vulnerable to privacy or causative attacks and they are therefore outlined. Chapter 2 concludes with a discussion of the most closely related work and how this thesis contributes to the state of the art.

Chapter 3 begins by outlining the system model used in the current work, specifically

for AMC, which is the reference signal classification task studied. A threat model is then presented, that encompasses the unique adversarial goals and capabilities an adversary may have or possess in the context of RFML. This threat model details how the current work, and the related work presented in Chapter 2, fit within this model. Chapter 3 concludes with a presentation of the specific DNNs studied in the current work along with their training data.

The study of specific evasion attacks begins in Chapter 4 with the presentation of a direct access evasion attack. Chapter 4 begins by providing a gentle introduction to adversarial machine learning before specifically describing how FGSM creates adversarial examples. The FGSM algorithm is then modified to create perturbations that are constrained by power ratios, instead of simply a distance in the feature space, in order to more intuitively capture the relationship between perturbations and the underlying signal. Chapter 4 then first performs a baseline evaluation to verify that FGSM is effective for a direct access attack and shows that FGSM provides a 10 dB improvement over simply adding Gaussian noise to the signal. After confirming the viability of adversarial machine learning in RFML, Chapter 4 then evaluates the impact of one key difference between DNNs in CV and DNNs in RFML: the input dimensionality is vastly lower in most RFML applications. It is then found that this lower input dimensionality actually provides some robustness to adversarial examples by showing that larger input sizes are more accurate when the model is not under attack, but, smaller input sizes are more accurate when the model is under attack by FGSM. Chapter 4 concludes by analyzing the impact of Additive White Gaussian Noise (AWGN) on individual

adversarial examples and finds that an FGSM attack can be successful when there is little to no noise applied to the sample; however, the evasion attack can become less successful as the perturbation power approaches the noise floor.

Chapter 5 expands the small-scale study of the impact of noise on adversarial RFML, in Chapter 4, into a large scale study of physical evasion attacks using the FGSM algorithm. After describing the simulation environment that allows for the evaluation, Chapter 5 presents results outlining the impact of three RFML specific effects that would undoubtedly occur in an OTA attack: AWGN, sample time offsets, and center frequency offsets. The impact of AWGN on adversarial RFML presented at the end of Chapter 4 on individual examples is shown to generalize in the results of Chapter 5. Further, it is shown that the additional signal detection effects of sample time offsets and center frequency offsets can impact adversarial success by as much as 20%. Chapter 5 also evaluates the impact on BER, the primary metric of interest in wireless communications, under the assumption of perfect synchronization and concludes that a direct translation of adversarial machine learning methodology from CV is less effective in the case of higher order modulations, such as QAM16, where the perturbation has a higher impact on BER than to classification accuracy.

Following from the conclusions of Chapter 5, Chapter 6 presents adversarial methodology that directly accounts for BER and does not depend on time synchronization between the adversarial transmitter and eavesdropper, which would never realistically occur. The proposed methodology is shown to perform (in terms of adversarial success) as well, or better, than the FGSM attack outlined in Chapter 5. Further, the methodology presented in Chapter

6 achieves these results by encapsulating the adversarial generation methodology in DNNs and therefore does not require solving an optimization problem for every communications block that needs to be transmitted, greatly freeing resources in an adversarial transmitter. Chapter 6 concludes with a discussion of the spectrum usage by the proposed methodologies and how this provides benefits to the intended receiver, by allowing them to filter out parts of the perturbation with a matched filter, and the limitations this could have in an attack against a real system where an eavesdropper may obtain that same benefit during its signal isolation stage.

This thesis then concludes in Chapter 7 where the current vulnerabilities of RFML are summarized. Chapter 7 then presents a short discussion of what it means to be "secure" against adversarial machine learning in both a civilian and military context, where the recourse for being under attack is different. Chapter 7 then concludes with proposed future directions for research in adversarial RFML.

## 1.3 Publications

### Thesis Publications

[1] **Bryse Flowers**, R. Michael Buehrer, and William C. Headley. Evaluating adversarial evasion attacks in the context of wireless communications. Submitted to *IEEE Transactions on Information Forensics and Security*, Feb 2019.

[2] **Bryse Flowers**, R. Michael Buehrer, and William C. Headley. Communications aware adversarial residual networks. Submitted to *IEEE Military Communications Conference (MILCOM)*, May 2019.

### Relevant Publications

[3] **Bryse Flowers**. Adversarial RFML: Threats to deep learning enabled cognitive radio. Invited speaker at *ACM Workshop on Wireless Security and Machine Learning (WiseML 2019)*, May 2019.

[4] Samuel Bair, Matthew DelVecchio, Bryse Flowers, Alan J. Michaels, and William C. Headley. On the limitations of targeted adversarial evasion attacks against deep learning enabled modulation recognition. In *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, WiseML 2019, pages 25–30, New York, NY, USA, 2019. ACM.

# Chapter 2

# Background

This chapter begins by outlining the wireless physical layer, which provides the necessary background to understand the communications aware attack presented in Chapter 6. It then describes what a CR is, why it would be used, and how DNNs are natural enablers of this category of device. After zooming in on a subset of CR, the deep learning enabled signal classification stage, this chapter then overviews all threats to that sub-module. Traditionally, these threats would fall under either cybersecurity or electronic attacks; however, the current work is primarily concerned with adversarial machine learning, specifically it is concerned with evasion attacks. While a quick overview of evasion attacks is presented in this chapter, along with the most closely related prior work, evasion attacks in the context of wireless communications is discussed in more detail, and a threat model for RFML is formulated, in Chapter 3.

## 2.1 Wireless Physical Layer

The primary goal of any wireless communications system is to communicate information from a transmitter to a receiver. A simplistic diagram, for a digital communications system, is shown in Figure 2.1. In this diagram an "application" seeks to transmit bits to an application running on another device. For simplicity, Figure 2.1 does not make any distinction between possible applications and omits all other layers of the networking stack. The transmitter would then "modulate" that information by first encoding the bits into a symbol space, interpolating and filtering the signal ("pulse shape") in order to limit the spectrum usage,

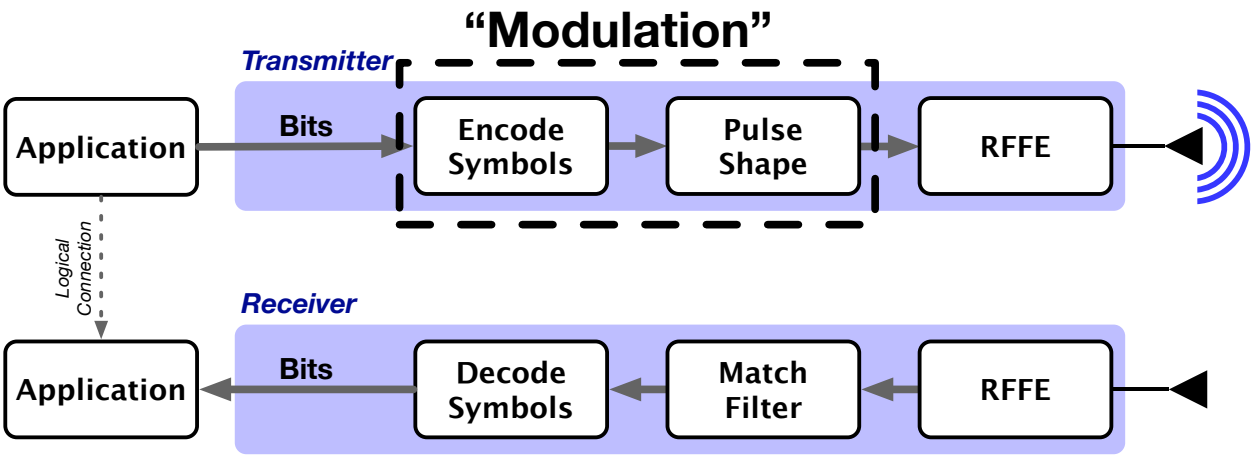


**Figure 2.1:** Overview of a digital communications transmitter/receiver pair.

and then sending the signal to a Radio Frequency Front End (RFFE) [1]. The RFFE would translate the signal from digital to analog using a Digital to Analog Converter (DAC), shift the signal in frequency onto the carrier frequency, and then amplify and transmit the signal using an antenna. The receiver implements the inverse of this process. The RFFE of the signal receives and amplifies the signal before shifting the signal in frequency down to baseband and digitizing it using an Analog to Digital Converter (ADC). A "matched filter" is applied to the signal, which maximizes the Signal-to-Noise Ratio (SNR) of the received symbols. The signal is then decimated back to one sample per symbol, ideally at the optimal sampling time to maximize SNR, and a hard decision is typically made to decode the symbols back into bits.

In any real environment, there will be multiple transmitters and receivers. One pair's signal is another pair's source of interference because the wireless spectrum is finite. With

[1] The current work is focused on Linear Digital Amplitude Phase Modulation (LDAPM).

the recent explosion in Internet of Things devices, as well as Connected Vehicles, the number of wireless devices in the environment is growing exponentially, but the available spectrum does not increase and therefore becomes more scarce with each new device. One approach to dealing with this scarcity is through CR [14, 15].

## 2.2 Cognitive Radio

CR is a term coined in 1999 [14] and is described in [15] as an intelligent device that senses and understand its environment via processing of Radio Frequency (RF) signal data (spectrum sensing) and adjusts its transmission parameters based on this knowledge of the environment (subject to some policy constraints) in order to achieve highly reliable communications or another user-defined goal such as efficient spectrum usage. One way to achieve more efficient spectrum usage is through DSA.

There are three typical spectrum access technologies [49]: licensed, unlicensed, and managed. A traditional cellular system would be centrally administered where a base station governs the spectrum usage in order to achieve a desired receiver performance; however, CRs have the ability to form decentralized, or *ad hoc*, networks that could take advantage of spectrum holes in unlicensed or managed frequencies [15]. These spectrum holes are time, space, and frequency dependent based on the transmitters operating in the vicinity of the CR. A CR must therefore, take in a wideband signal (observe), perform analysis of that spectrum (orient), adjust transmission parameters to take advantage of spectrum holes or

better adapt to the interference of the environment (decide), and then transmit (act). In the simplest case, a CR could identify an unused frequency band and subsequently transmit on this band.

### 2.2.1 Key Enablers of Cognitive Radio

CR and Software Defined Radio (SDR) go hand-in-hand as it is nearly impossible to realize a CR that does not operate on top of SDR. SDRs provide the ability to reconfigure transmission waveforms on the fly and CRs provide the intelligence needed to determine which parameters to reconfigure. Although a CR could operate without Machine Learning (ML), such as with a static policy database that simulates intelligence, ML has become pervasive in wireless communications research. This is natural because ML can scale to larger action spaces that would become memory prohibitive in a purely database enabled radio. Further, ML is able to adapt without a human in the loop, and therefore can accomplish a key goal of CR which is to learn from its environment and actions. These two key enablers, SDR and ML, are discussed in more detail below.

#### 2.2.1.1 Software Defined Radio

SDRs were introduced around the same time as CRs [50,51]. Their defining characteristic was that the majority of the signal processing is performed in software as opposed to hardware. Performing signal processing in software allows for nearly limitless flexibility through the lifetime of a radio because it can be updated via OTA software patches, but, comes at the

cost of computing overhead. As computing power becomes cheaper, and specialized chips for Digital Signal Processing (DSP) become available, SDR becomes a feasible solution for real world applications.

SDRs have also benefited from open source frameworks that reduce the cost to create a waveform [52–54]. Although the exact methodology differs between each framework, they all provide reusable signal processing blocks, such as for filtering, power spectral density estimation, or modulation, that can quickly be employed for each new waveform. The current work utilizes GNU Radio [52] which provides life-cycle management, scheduling of signal processing blocks, a graphical programming interface for waveform creation, and good software driver support for a range of SDR devices known as a Universal Software Radio Peripheral (USRP) [55].

#### 2.2.1.2 Radio Frequency Machine Learning

DARPA RFML Systems [22] is a program aimed at developing "the foundations for applying modern data-driven Machine Learning to the RF Spectrum domain." Two primary goals of the program are to develop technologies to improve spectrum awareness as well as to autonomously control radio resources. The key difference between RFML and prior applications of machine learning to wireless communications is the desire to remove the need for hand engineered features. Therefore, while the current work would not consider feature based machine learning [19, 21] as RFML, it uses RFML to succinctly describe the latest iteration of technologies, based on deep learning, that do not use human engineered features

as input [8, 9, 13, 48]. Removing the need for human engineered features can be very beneficial, particularly in quickly changing environments, because it removes the need for a human in-the-loop and thus greatly speeds up adaptation. However, it is important to understand the security risks to these systems, which is the focus of the current work.

Machine learning is a natural enabler of CR in many aspects. One of the key goals of CR is to be able to learn from the environment. Modern advances in reinforcement learning have shown the ability of DNNs to learn a Q function that approximates the reward of a set of actions given the current state of the environment [56]. A similar technology can be applied to CR where a deep reinforcement learning agent controls radio resources [7]. Deep learning enabled waveforms have also been developed. In general, a known modulation form is used; however, with auto-encoders, researchers have shown the ability to use DNNs to directly synthesize a waveform that is conditioned on the data [6].

Even if machine learning does not control the entire radio, it can aid in spectrum sensing [57] by building off of the large body of success with deep learning in CV classification systems by applying these technologies to signal classification tasks given raw IQ inputs [8, 9, 13, 48]. Traditionally, signal classification would consist of two distinct steps. The first step would detect and isolate a signal in time and frequency while the second step would then classify that isolated signal. While DNNs can combine these two steps [58–61] in a process known as semantic segmentation, the current work does not consider these technologies and specifically focuses on traditional signal classification aided by DNNs as that is the most mature technology.

Threats to all of the CR tasks performed by DNNs mentioned in the current section can have similar goals; however, the current work specifically focuses on threats to the signal classification tasks as signal classification has been the most widely researched RFML task thus far. While there is some loss of generality from focusing on these signal classification tasks, the methodology developed can be easily transferred to all other tasks.

#### 2.2.1.3 Spectrum Sensing: Blind Signal Classification

The radio environment of a CR is not known *a priori* and therefore must be sensed in real time. A receiver would typically not know when or where (in space or frequency) a transmission would occur. Therefore, a CR must first detect a signal and isolate it in time and frequency, before subsequent classification. After isolation, parameters of the signal and the transmitter of that signal can be estimated or classified for usage in the decision stage. In the context of CR[2], estimating the power, center frequency, and bandwidth of a signal can aid in identifying spectrum holes. A CR could classify the modulation of the signal used, which would allow for automatic demodulation if the format is not known *a priori* or a CR transmitter adapts the modulation on the fly. Further, the CR could perform specific emitter identification, which would aid in tracking transmitter's in the environment even as their transmission parameters change. Knowing a transmitter's traffic patterns can aid in predicting when they will transmit and therefore aid in identifying future spectrum holes.

A simplistic diagram of a RFML system for signal classification is shown in Figure 2.2.

[2]Although CR is used as the motivating example in the current work, signal classification tasks, and therefore the threats to signal classification, have obvious applications to signals intelligence.

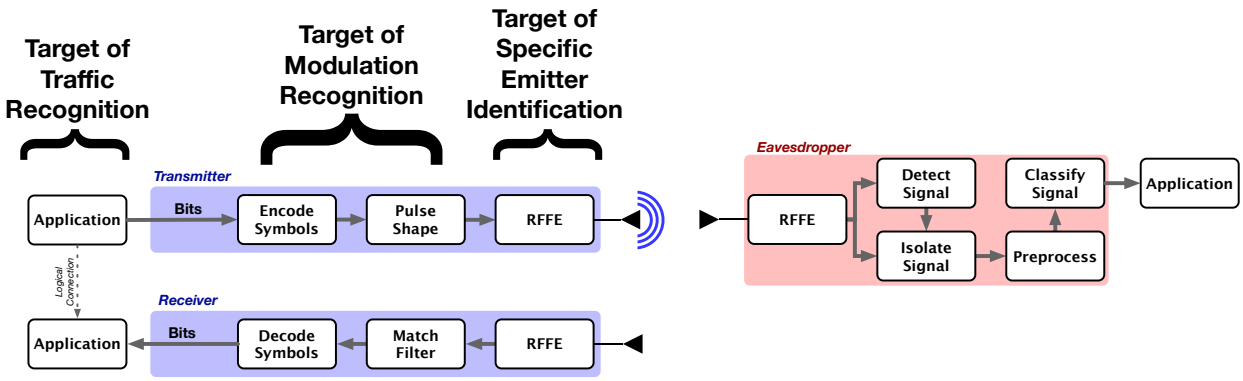


**Figure 2.2:** Overview of a RFML signal classification system (eavesdropper device). The eavesdropper could use RFML signal classification for traffic recognition [62], modulation recognition [16], or specific emitter identification [13, 24].

This system has no *a priori* knowledge of the signals it will encounter and is thus performing blind signal classification. Therefore, it first detects and isolates a signal in time and frequency, before performing signal classification on short segments of a continuous signal using DNNs with minimal pre-processing. The outputs of the classification stage can then aid in spectrum awareness.

### 2.2.2 Competition in Ad Hoc Networks

The spectrum is a limited natural resource and therefore CRs may compete for spectrum resources. In [15] Haykin describes a game theoretic approach to this competition that leads to a Nash equilibrium provided that each CR is "rational", and therefore takes the most optimal action, and that each CR has a "view of the world". Therefore, a natural threat to this system is if a CR deceives the spectrum sensing stage of all neighboring CRs in

order to exploit the system and gain spectrum resources for themselves. The current work focuses on envisioning ways that a CR could deceive deep learning enabled spectrum sensing, specifically the signal classification task.

## 2.3 Threats to RFML Signal Classification

There are multiple attack vectors to compromise a RFML system. Electronic Warfare (EW) refers to the usage of the spectrum to impede an enemy. While this could be stretched to apply to the current work, the current work uses EW to describe attacks on the signal detection stage or RFFE in Figure 2.3. An adversary could direct energy at the RFFE to remove its ability to perceive other signals or could create communications signals which have a low probability of detection and thus would never be sent to the signal classification stage. The current work does not consider attacks against the RFFE or signal detection and instead considers them static and therefore do not respond to the attack.

RFML systems are built on top of SDR [63] and thus are subject to traditional cyber-security attacks against these software implementations [64]. The vulnerabilities of GNU Radio [52] to these attacks have been demonstrated in [65,66]. While cyber-security attacks could clearly impact RFML signal classification, these are not the focus of the current work.

The current work examines the threats specific to the signal classification stage and is thus concerned with adversarial machine learning [67] which has seen a surge of activity in the context of CV [25].

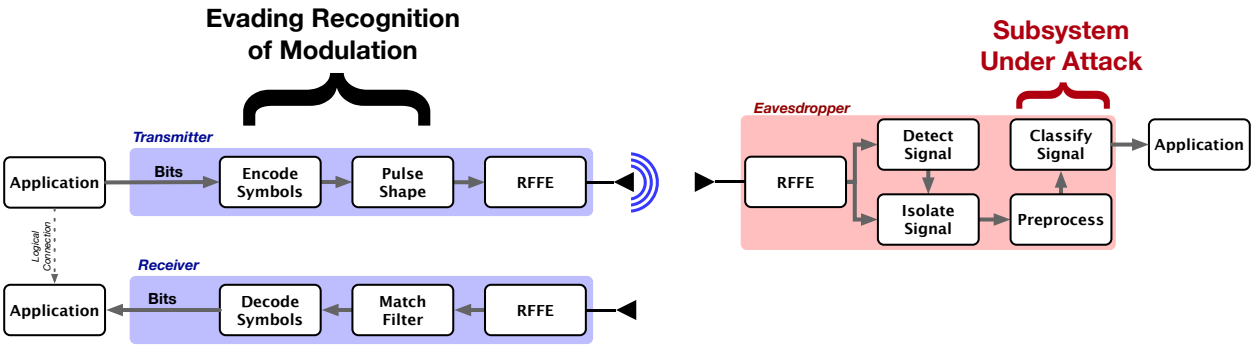


**Figure 2.3:** The current work is only concerned with attacking the signal classification subsystem. This is studied using modulation recognition as a reference signal classification task.

### 2.3.1 Adversarial Machine Learning

The threat surface for adversarial machine learning in the context of RFML is surveyed in Figure 2.4. The types of attacks shown can be split into three logical categories: privacy attacks, causative attacks, and evasion attacks.

#### 2.3.1.1 Privacy Attacks

Privacy attacks observe information about the inputs and outputs of a classifier in order to gain information about how it works. Membership Inference attacks determine whether a specific input was a part of the training dataset of the classifier [68]. Model Extraction attacks use the same information in order to build a nearly equivalent model [69]. Privacy attacks are, in general, a concern because they compromise the confidentiality of the model. This is especially prevalent when the training data could be sensitive, such as in healthcare applications.

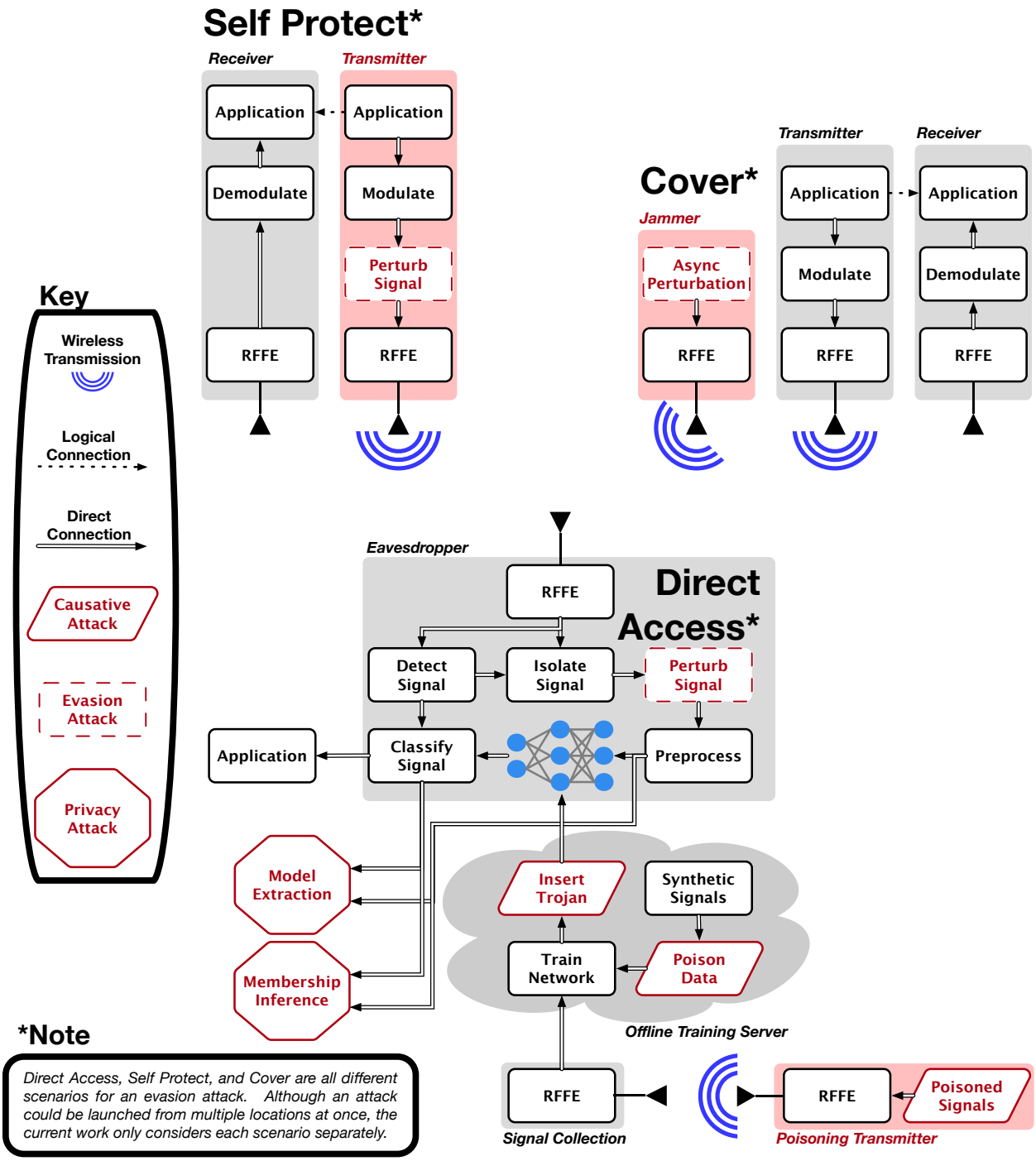


**Figure 2.4:** Threat Surface of a RFML signal classification system presenting an overview of where attacks could be launched from.

In addition to being an attack in their own right, these privacy attacks can be exploratory and provide an adversary with more knowledge in order to attempt an evasion attack. Therefore, while these attacks could be supplementary to the current work, they are not the focus and are therefore not discussed further.

#### 2.3.1.2 Causative Attacks

Causative attacks exert influence over a model's training process in order to inject vulnerabilities. A data poisoning attack would manipulate the training data in order to change the learned decision boundaries of a model [70–72]. This attack type will be extremely concerning to any system that (re-)trains using OTA data captures because an adversary would be able to inject training data just by being in proximity to the signal collection device. These attacks are already beginning to be researched in the context of CR [44, 73].

A software Trojan attack compromises the training stage of a network in order to inject a vulnerability in the learned parameters that can later be exploited [74]. This type of attack assumes a very high level of access because it requires the ability to directly set the parameters of the DNNs which it is attacking. While this may seem unreasonable, it can be a serious threat when considering that multiple companies typically collaborate on a single product. If Company A is employed to train the model while Company B assembles and sells the final product, then Company B may like to verify that the model it is being supplied is not compromised in some way that can later be exploited.

The eventual goal of causative attacks is to degrade classifier performance at inference

time through attacks at training time. Therefore, while the current work does not consider the high level of access needed for a causative attack, it does explore evasion attacks which have similar goals but don't assume any influence over the training process.

#### 2.3.1.3 Evasion Attacks

The current work focuses on adversarial evasion attacks in the context of RFML. While the specific goals and capabilities an adversary must possess to achieve adversarial success are discussed in Chapter 3, it can, in short, be described as an attack that assumes a fully trained and static model to which it has control over the inputs. An adversary then intelligently crafts "perturbations" to the input that cause a misclassification. This type of attack is well studied in CV literature [27, 30, 32–34, 37, 39, 40, 75, 75–77] but is just beginning to be researched in the context of RFML [43–45, 73].

## 2.4 Related Work

Prior security threats to cognitive signal classifiers have been researched [78, 79], however, the state of the art signal classification systems use deep learning techniques [8–13] whose vulnerabilities have not been studied extensively in the context of RF. In [44] and [73], the authors consider adversarial machine learning for intelligently jamming a deep learning enabled transmitter, at transmission time and sensing time, to prevent a transmission. Their work considers learning OTA by observing an acknowledgement from a receiver as a binary feedback. While their work is primarily concerned with preventing transmission, the current

work is primarily concerned with enabling transmission while avoiding eavesdroppers and is thus fundamentally different. Constraining a perturbation to be interpretable, in the form of demodulation by the intended receiver, places a much tighter bound on the problem.

The work presented in [43] was the first research into adversarial evasion attacks in RF. In this preliminary work, the authors present a study of the effectiveness of two adversarial algorithms, a variant of the FGSM and Universal Adversarial Perturbation (UAP), against DNNs trained using the RML2016.10A dataset [80]. Therefore, [43] is a close analogy to Chapter 4 in the current work. However, while the authors of [43] showed results at varying SNRs, they implicitly assumed direct access to the classifier by not adding noise to the perturbation. Chapter 4 extends this work to consider the impact of AWGN on the perturbation on a small scale and Chapter 5 provides a large scale analysis of AWGN as well as other effects to show that adversarial success rates, when an attack is launched OTA and therefore cannot directly feed adversarial examples into the classifier, are lowered. Further, Chapter 5 then shows that these adversarial methodologies can become prohibitive for higher order modulations, such as QAM16, because the underlying signal is no longer interpretable by the intended receiver. Therefore, deception can only be achieved at the cost of forgoing much of the initial communication capacity of the link. The work in Chapters 4 and 5 were then published in [1].

The very recent, and independent, work published in [45] then echos the findings of Chapter 5 that AWGN has a negative impact on adversarial success and begin evaluating the attack in terms of BER, as is done in Chapter 5. While Chapter 5 merely evaluates

current attacks in terms of BER, [45] develops methodology that directly accounts for BER when crafting the perturbations. However, their methodology misses two key practical points of deploying an adversarial RFML system. First, [45] assumes the eavesdropper and the adversary are synchronized in time, which cannot be assumed because the eavesdropper is performing blind signal classification. Second, [45] presents methodology that requires solving an optimization problem for every communications block to be transmitted, which is computationally expensive. Chapter 6 addresses both of these issues. First, the underlying training methodology does not assume time synchronization and therefore the perturbations created do not depend on synchronization for adversarial success. Second, the methodology presented in Chapter 6 works by encapsulating the perturbation creation into a fully convolutional neural network. Therefore, once trained, this network could be easily deployed as a non-linear filter and would not require the extra computation of solving an optimization problem for each communications block.

Adversarial RFML is quickly becoming an active area of research with new literature being put out every day; therefore, the current work is not all encompassing of current progress in adversarial RFML. The current work only considers untargeted attacks, which seek to degrade classification accuracy but do not seek to masquerade as a specific target class. The work of Chapter 4 has been extended into a targeted attack in [4]. The work shown in [4] substantiates the claim in Chapter 3 that targeted attacks are more difficult than untargeted attacks. Further, [4] shows that targeting a "difficult" class, such as an analog modulation format when starting with a digital modulation format can require high

powered perturbations. Therefore, while the methodology presented in Chapter 6 could be extended into a targeted attack, it is not considered in the current work due to the power requirements to achieve high success rates.

The first step to securing any system is understanding the threats to that system. The current work takes that first step by characterizing where the threats could come from, envisioning new threats, and describing the limitations of adversarial evasion attacks in the context of wireless communications. However, the current work does not explicitly consider any defensive measures on the eavesdropper device. While many defensive strategies have been considered in CV, one of the best is adversarial training [34, 41]. The work in [47] has applied this concept to RF with good preliminary results. The current work concludes with a short discussion of what it means to be secure against adversarial RFML in both a civilian and military context.

# Chapter 3

# Attack Evaluation Methodology

This chapter outlines the common methodology used throughout the current work to evaluate the success of the developed adversarial evasion attacks. Although the current work uses AMC as a reference CR task, the threat model (presented from the perspective of the eavesdropper) would hold provided that:

1. The adversary's primary goal is the wireless transmission of information to a cooperative receiver.

2. The eavesdropper employs a DNNs that aids in signal classification or decision making by outputting the most likely class or most beneficial action.

Therefore, the vocabulary used in this chapter can be used to describe threats to nearly all deep learning enabled CRs. The chapter begins by describing the system model used in the current work, then describes the unique threats that can be posed to that model, and concludes with a reference implementation of the eavesdropper's classification network.

## 3.1 Automatic Modulation Classification System Model

The current work considers the task of blind signal classification where an eavesdropper attempts to detect a signal in the spectrum, isolate it in time and frequency, and perform modulation classification. This task assumes that the signal is a wireless communication between a transmitter and a cooperative receiver where the eavesdropper is not synchronized and has very limited *a priori* information about the communication. Ultimately, the eaves-

dropper could then use the output for DSA, signals intelligence, and/or as a preliminary step to demodulating the signal and extracting the actual information transmitted.

The study of adversarial examples in this model could be framed from the perspective of either the eavesdropper or the transmitter. First, this study can be considered a vulnerability analysis of RFML systems and the information gained can then be used to produce a more robust eavesdropper that is hardened against deception by adversarial machine learning [47]. Additionally, this study could be considered a feasibility analysis for methodology to protect transmissions from eavesdroppers. Evading an eavesdropper can limit tracking of the transmitter or automatic demodulation of its transmission. The current work does not take a side in the application of this technology and presents a case for both sides; however, **the term adversary is used to describe the transmitter that seeks to evade an eavesdropper for the remainder of the current work.**

## 3.2 Threat Model

A rich taxonomy already exists for describing threat models for adversarial machine learning in the context of CV; however, threat models which only consider CV applications lack adversarial goals and capabilities that are unique to RFML. Therefore, the current work extends the threat model initially proposed in [37] for RFML in Figure 3.1 and outlines how the current work fits in with the related literature. Before describing the current work's place within the literature, this section first expands on the unique categories of adversarial

goals and capabilities that must be considered when discussing adversarial threats to RFML systems. The goals an adversary may have are presented across the horizontal axis (Figure 3.1) and the capabilities, or prior knowledge, an adversary may possess are shown along the vertical axis. Therefore, the "easiest" adversarial evasion attack would be presented in the upper left of the diagram and the "hardest" adversarial evasion attack would be presented in the bottom right.

### 3.2.1 Adversarial Goals

Three main goals are traditionally considered for adversarial machine learning [37]: confidence reduction, untargeted misclassification, and targeted misclassification. An example of what the classifier's output could look like in a successful case of each attack is shown in Figure 3.2. Confidence reduction is the easiest goal an adversary can have. It simply refers to introducing uncertainty into the classifier's decision even if it ultimately determines the class of signal correctly. To put more simply, an adversary would desire to lower the output of the true class of a network's output but not necessarily care if it is still the maximum output of the network.

An adversary whose goal is simply to be classified as any other signal type than its true class, can be described as untargeted misclassification. Put simply, an adversary desires to lower the output of the true class of a network's output. In this case, the adversary does not care which other class becomes the maximum output of the network as long as it is not the true class.

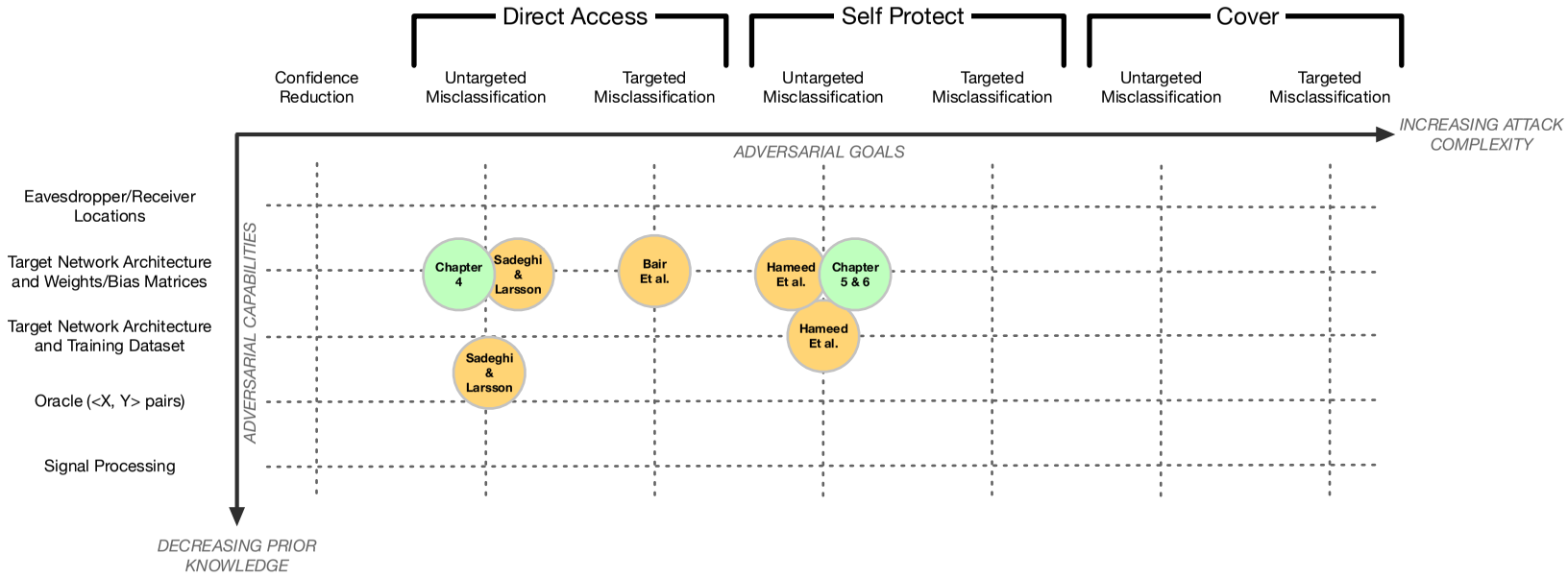


**Figure 3.1:** Threat Model for evading RFML signal classification systems presented in the style of [37]. The current work is highlighted as green circles while the related work by Sadeghi & Larsson [43], Bair et al. [4], and Hameed et al. [38] are highlighted as orange circles.

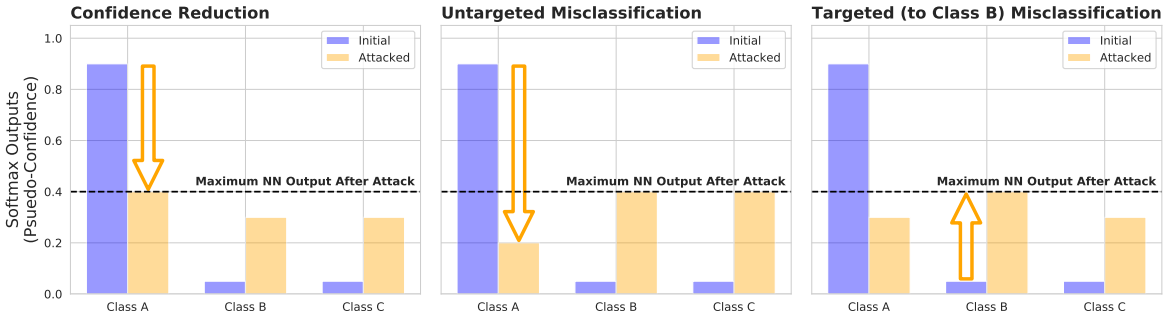


**Figure 3.2:** Example confidence outputs of a model for confidence reduction, untargeted misclassification, and targeted misclassification. In all plots, the original example belongs to Class A and is classified correctly. The output of the classifier on the "initial" example is shown in blue, while, the output of the classifier on the adversarial example, or the output when "attacked", is shown in orange.

Targeted misclassification is typically the most difficult goal of adversarial machine learning. It occurs when an adversary desires a classifier to output a specific (and incorrect) target class instead of simply any class that is not the true class. Due to the hierarchical nature of human engineered modulations, the difficulty of targeted misclassification for AMC depends heavily on the signal formats of the true and target class. Targeted misclassification are sometimes split between attacks that start with a real input [35, 39] versus those that start with noise [38]. The threat model presented in Figure 3.1 only considers the former because the current work assumes that an adversary's primary goal is to transmit information and not simply degrade classifier performance. Without this assumption, the problem simplifies to replaying a known signal that corresponds to the desired target class or transmitting noise at the eavesdropper if it does not care about the eavesdropper's classification.

Further, the current work categorizes adversarial goals based on where the attack is

launched from: "at the eavesdropper" with direct access, "from a transmitter" with self protect, or "from a separate device" with cover. The specific locations as well as when they are discussed in the current work is shown in Figure 3.3, the following subsections go into further details about each location, and the pros and cons of attacks from each location are later summarized in Table 3.1.

#### 3.2.1.1 Direct Access

Traditional adversarial machine learning, such as those generally considered in CV or the attack considered in [43], fall into the direct access category. This category of attack is performed "at the eavesdropper" as part of their signal processing chain. Therefore, the propagation channel and receiver effects for the example is perfectly known at the time of crafting the perturbation, the perturbation itself is not subjected to any receiver effects, and the perturbation will have no effect on the intended receiver because it is not sent OTA. Attacks at this level are very useful for characterizing the worst case vulnerabilities of a classifier but they are less realistic in the context of RFML because it assumes that the signal processing chain has been compromised.

#### 3.2.1.2 Self Protect

When the adversarial perturbation is added at the transmitter and propagates along with the transmitted signal to the eavesdropper, this can be categorized as self protect. By adding the perturbation at the transmitter, the perturbation can still be completely synchronous with

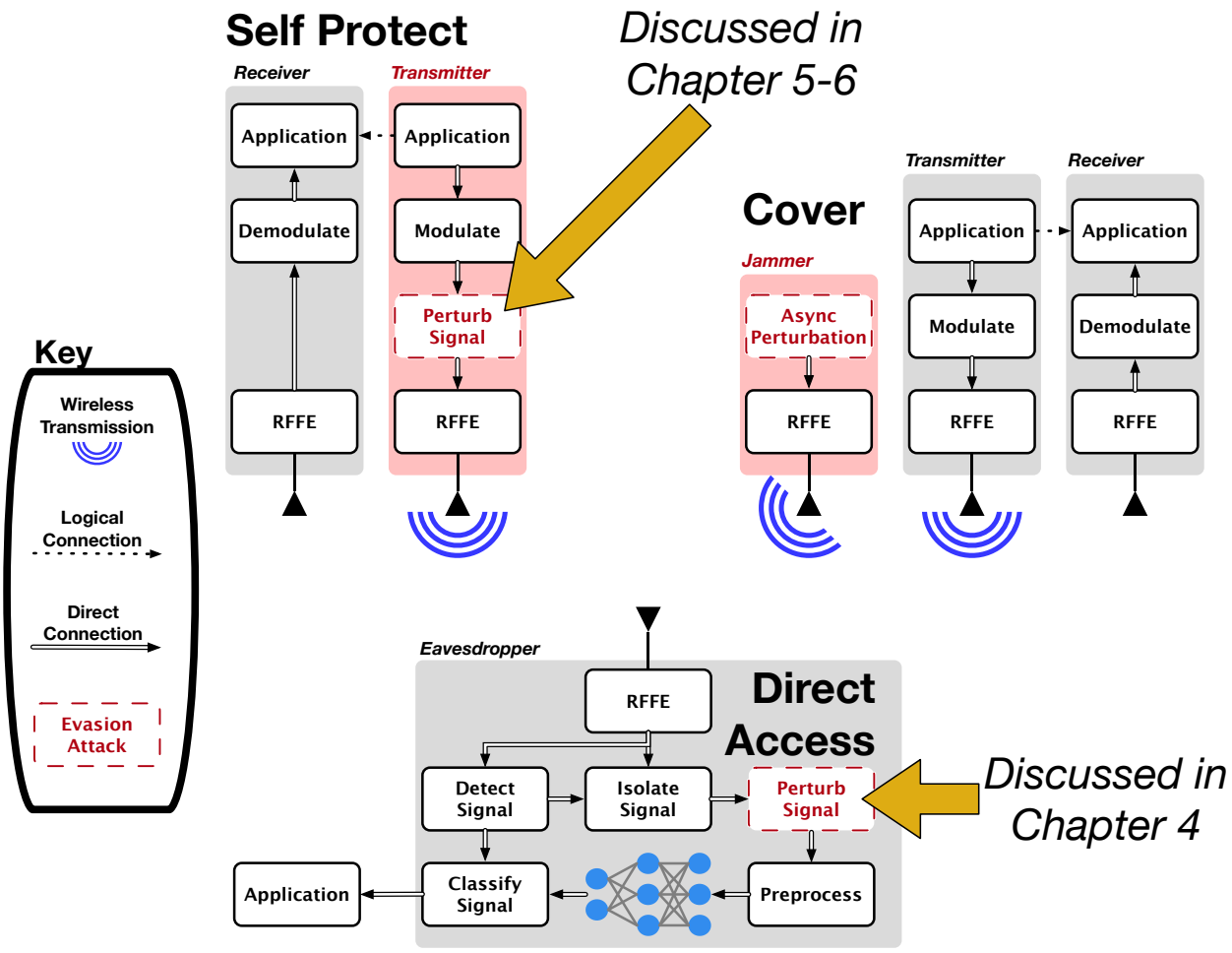


**Figure 3.3:** Overview of locations an adversarial evasion attack could be launched from. Direct access attacks are considered in Chapter 4 while Chapters 5 and 6 discuss self protect attacks

the signal transmission; however, the perturbation will now be subjected to all of the receiver effects traditionally considered in RFML and will also impact the intended receiver. While many of the algorithms that are successful for the direct access category of attacks will be applicable to self protect, the evaluation of adversarial success must take into account receiver effects. Therefore, attacks that seek to create minimal perturbations, such as the modified FGSM method presented in [43], will no longer work because adversarial success can not be guaranteed due to the signal being subjected to a stochastic process. The concurrently developed work presented in [45] would fall under this category of attack. Further, the current work focuses specifically on this category of attack and presents advances over the current adversarial methodology in Chapter 6.

#### 3.2.1.3 Cover

RFML allows for a third category of adversarial goals, in which the adversarial perturbation originates from a separate emitter from the transmitter and is only combined at the eavesdropper device[1]. Low cost transmitters can be Size, Weight, and Power (SWaP) constrained. Therefore, it may be beneficial to have a single unit provide cover for multiple SWaP constrained nodes. However, because these attacks cannot rely on synchronization between the transmission and perturbation, the perturbations must be time shift invariant [43] making this category of attack more difficult. The current work does not present a study of this category of adversarial goal and leaves that to future work.

[1]Although the intention of a cover device is to impact the eavesdropper, it is likely that unintended emissions would also impact the cooperate receiving device as well.

| Attack Location | Pros | Cons |
|---|---|---|
| *Direct Access* | 1. Perturbation not subject to noise.<br>2. No impact to intended receiver. | 1. Requires compromising the signal processing chain. |
| *Self Protect* | 1. Does not require access to the eavesdropper's device.<br>2. Synchronous with the underlying transmission. | 1. Impacts the interpretation of the signal by the intended receiver.<br>2. Requires increased computation complexity on the transmission device, possibly lowering data rates.<br>3. Perturbation is subject to noise. |
| *Cover* | 1. Can provide cover for multiple transmitters.<br>2. Does not require increased computational complexity on the transmission device.<br>3. Could use beamforming to steer energy at eavesdropper or away from receiver. | 1. Asynchronous with the transmission.<br>2. Could impact intended receiver.<br>3. Perturbation is subject to noise.<br>4. Eavesdropper could use adaptive beamforming to null energy from the cover device. |

**Table 3.1:** Summary of the pros and cons of adversarial evasion attacks from each location.

### 3.2.2 Adversarial Capabilities

An adversary can break the evasion task into two subsets. First, the adversary must determine the optimal signal that would disrupt the classification but not the receiver's demodulation. Then, the adversary must be able to transmit a signal such that this optimal signal would be received by the eavesdropper and receiver with minimal corruption. Traditional adversarial machine learning capabilities, such as those described in [37], generally help with determining "what you want a classifier to see" by providing information about the target DNNs that can subsequently be used to optimize the input. An adversary with a high level of capability may have perfect knowledge of the learned parameters of the model. These attacks are referred to as white-box in most literature. In a slightly more realistic case, the attacker may have access to the network architecture and training dataset, but not the learned parameters. The attacker must then create adversarial examples that generalize over all possible models created from the dataset and architecture. In a very limited case, the attacker may only have access to what is deemed an oracle, an entity that will label a limited number of $X, Y$ pairs for the attacker through an Application Programmer Interface (API) [32] or an observable wireless transmission [44,73]. This allows the attacker to perform limited probes against the target network in order to build up an attack.

Adversarial machine learning applied to RFML has a different class of capabilities an attacker can possess that can be thought of as "the ability to make a classifier see a specific example". RF propagation can be directed through the use of smart antennas and the use of this ability generally falls under physical layer security. However, the use of these

concepts typically only apply when the transmitter knows the location of the receiver and/or eavesdropper. It could direct its energy only at the receiver, thus likely minimizing the SNR at the eavesdropper. Similarly, a jammer could direct energy only at the eavesdropper, maximizing the impact of perturbations on classification accuracy while minimizing the impact to the receiver.

Signal processing chains can present an impediment to adversarial success. Traditionally, RFFEs are built to reject out of band interference and therefore adversarial perturbations consisting of high frequencies could be filtered out. Power amplifiers can exhibit non-linear characteristics which would distort the perturbation. The precision of the analog to digital converter could limit the attack to stair stepped ranges. Further, the adversarial perturbation could have cascading effects on the DSP present on the device such as impacting the signal detection and isolation stage, resulting in sample rate or center frequency offsets between the transmitter and eavesdropper. The signal processing chain is assumed by the related work [43, 45] and these effects are not discussed. Although some of these effects are modeled in Chapter 5 and 6, examining this is largely left to future work. Until these effects are accounted for, it is difficult to claim that any attack would work in a real environment with the same success rates seen in simulation.

### 3.2.3 Threat Model Assumed in the Current Work

In the current work we assume full knowledge of the learned parameters of the target DNNs and set the goal as untargeted misclassification. The current work considers perturbations

that are specific to the underlying transmitted signal and characterizes their effectiveness in the presence of receiver effects such as noise, sample time offsets, and frequency offsets. Therefore, both direct access attacks as well as self protect are considered.

The related work by Sadeghi and Larsson [43] presented an analysis of two untargeted misclassification attacks against AMC without a channel model applied to the perturbations. One attack assumed perfect knowledge of the target network and the other only assumed knowledge of the entire training dataset, but did not assume knowledge of the target network architecture. The related work by Hameed et al. [45] evaluated the attack in the presence of an AWGN channel and additionally considered BER for both evaluation and during crafting of adversarial perturbations. Many of the attacks in [45] were presented with full knowledge of the learned parameters but one attack only considered knowledge of the training dataset and target architecture and instead used a separately trained model to craft the perturbations.

The current work (as well as the related work [43,45]) does not assume knowledge of either the eavesdropper or receiver locations and therefore does not consider directional antennas and instead shows results across varying SNR ranges. Further, the current work assumes that the receiver is fixed and thus does not introduce any modifications to the receive chain that accounts for the perturbations added to the signal. In [43], because it was a direct access attack, there was no impact to the receiver chain. In [45], the authors added convolutional coding to the bit stream in order to properly demodulate the signal in the presence of their attack. The current work does not assume there is any coding on the bit stream and presents

results without it.

Further, the current and related work [43, 45] implicitly assume knowledge of the signal processing chain. This implicit assumption comes because none of the current adversarial evasion attacks in the context of RFML fully model the cascading effects on signal detection and isolation or the impact of non-linearities or quantization error in the RFFE.

## 3.3 AMC Target Network

### 3.3.1 Network Architecture

The current work uses the DNNs architecture shown in Figure 3.4 which was first introduced in [8] for a reference raw IQ AMC model, but, the methodology in this work would hold for all network architectures. This architecture consists of two convolutional layers followed by two fully connected layers. This network takes the IQ samples as a $[1, 2, N]$ tensor which corresponds to 1 channel, IQ, and $N$ input samples. The current work uses extended filter sizes as done in [9] and [48], using filters with 7 taps and padded with 3 zeros on either side. The first convolutional layer has 256 channels, or kernels, and filters $I$ and $Q$ separately. The first layer does not use a bias term as this led to vanishing gradients during our training. The second layer consists of 80 channels and filters the $I$ and $Q$ samples together using a two-dimensional real convolution. This layer includes a bias term. The feature maps are then flattened and fed into two fully connected layers, the first consisting of 256 neurons and the second consisting of the number of output classes. All layers use ReLU as the activation

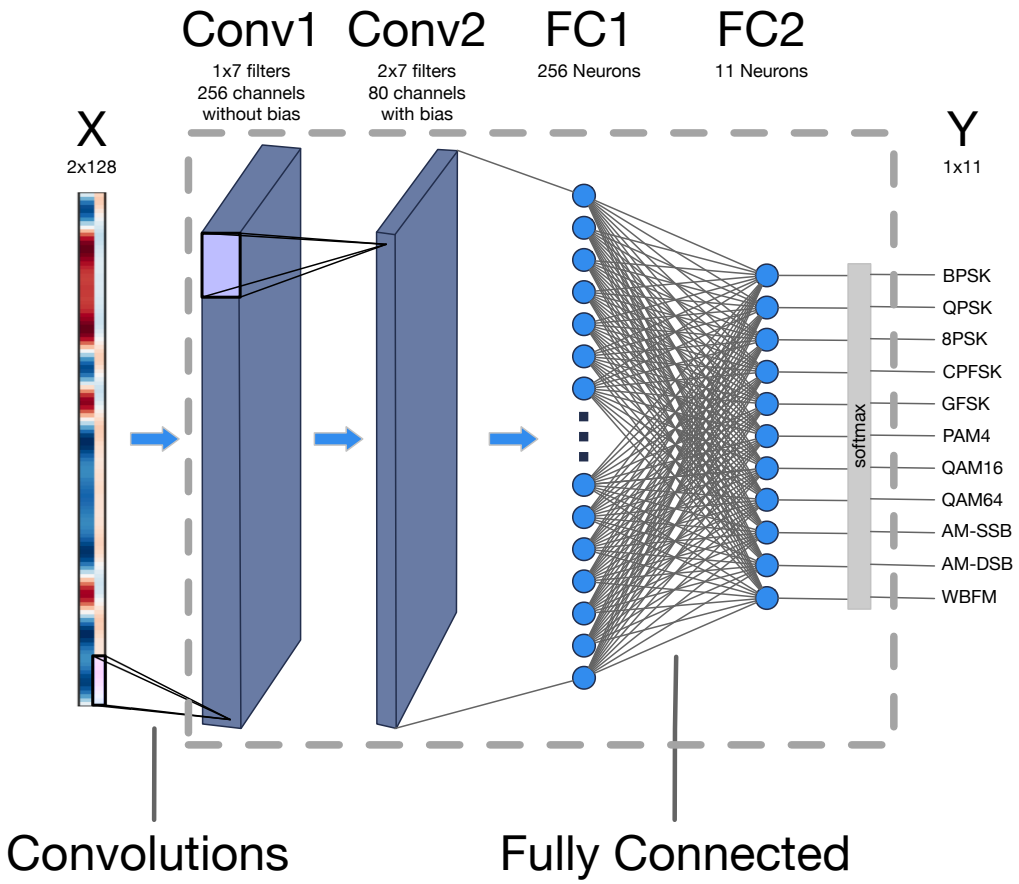


**Figure 3.4:** Convolutional Neural Network Architecture, first introduced in [8] and modified according to [9, 48], used in the current work for AMC.

function (except for the output layer). As a pre-processing step, the average power of each input is normalized to 1.

### 3.3.2 Dataset A

The majority of this work uses the open source RML2016.10A dataset introduced in [80]. This synthetic dataset consists of 11 modulation types: BPSK, QPSK, 8PSK, CPFSK, GFSK, PAM4, QAM16, QAM64, AM-SSB, AM-DSB, and WBFM. These signals are created inside of GNU Radio and passed through a dynamic channel model to create sample signals at SNRs ranging from $-20$ dB to 18 dB. A subset of the data is shown in Figure 3.5 and 3.6.

Using an open source dataset allows for reproduction of results; however, this dataset only provides one input size, 128 complex samples. Furthermore, this dataset contains limited center frequency offsets. Therefore, it was necessary to create an additional dataset to perform the additional evaluations in terms of center frequency offset contained in Chapter 5.

### 3.3.3 Dataset B

The main differences between Dataset A and Dataset B are the channel model as well as the modulations used. Dataset B contains a static channel model with center frequency offsets and SNR calculated as $E_s/N_0$ while Dataset A contains a dynamic channel model but no center frequency offsets. Dataset A contains a static multi-path effect that Dataset B does not. Additionally, Dataset B only uses a subset of the modulations that Dataset A contains

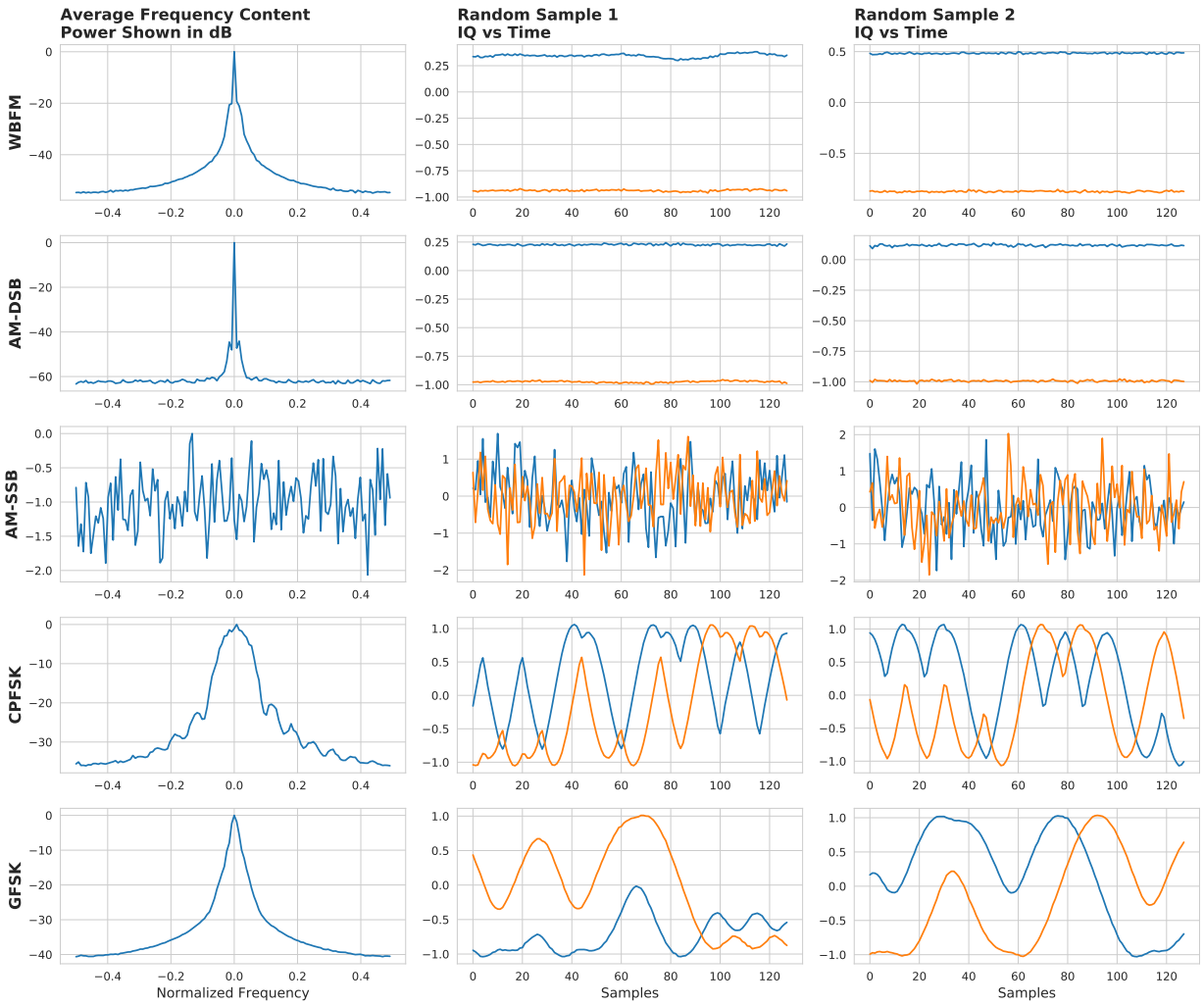


**Figure 3.5:** Random samples from FSK and Analog Modulations in Dataset A. The SNR was restricted to 18 dB but the specific examples in time were selected randomly from that subset. The frequency content is averaged across all examples that have 18 dB SNR and the corresponding modulation.

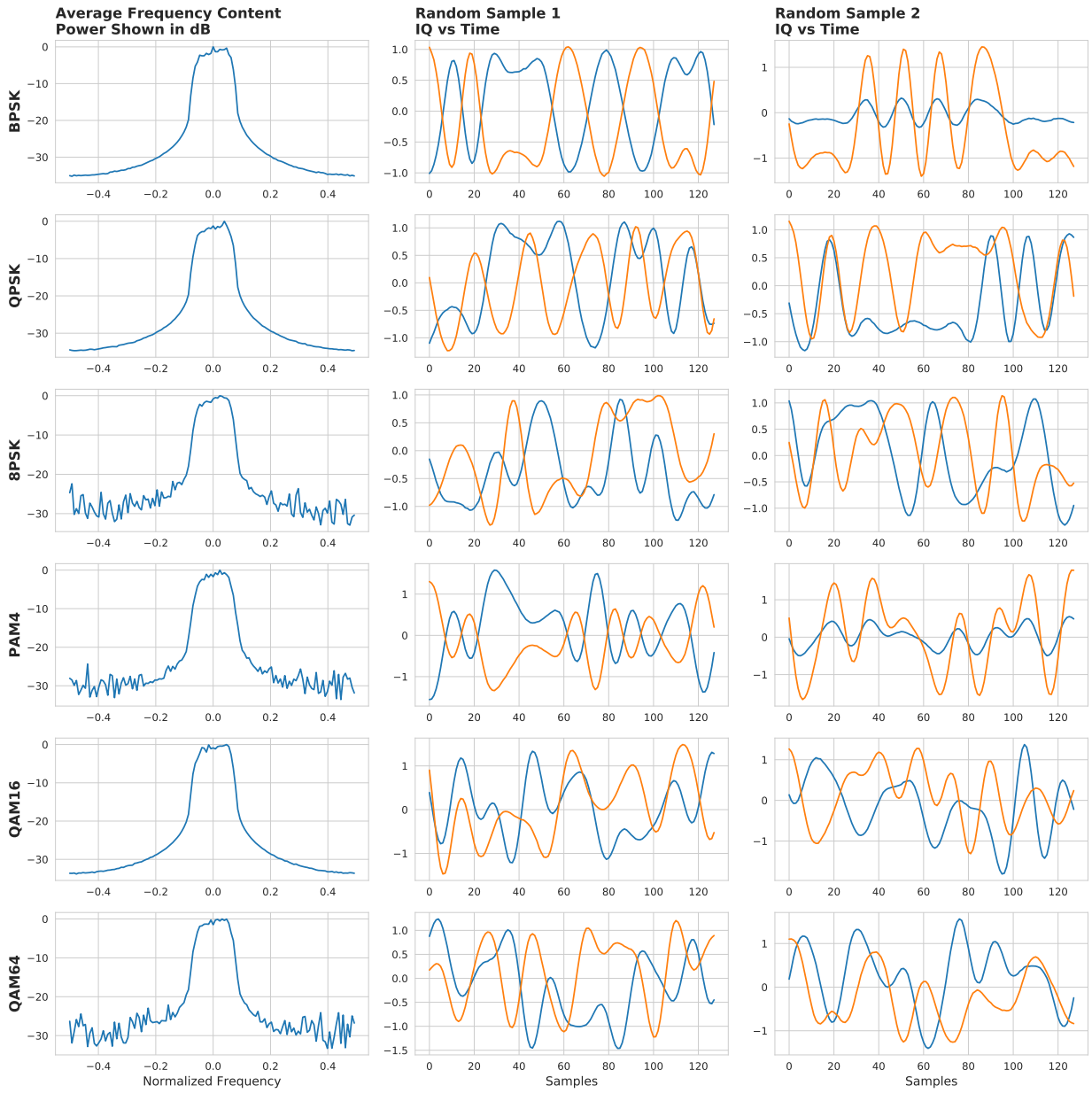


**Figure 3.6:** Random samples from LDAPM Modulations in Dataset A. The SNR was restricted to 18 dB but the specific examples in time were selected randomly from that subset. The frequency content is averaged across all examples that have 18 dB SNR and the corresponding modulation.

for simplicity.

This additional dataset was also created using synthetic data from GNU Radio. Three datasets were created with varying input size (128, 256, and 512). These synthetic datasets consists of 5 modulation schemes: BPSK, QPSK, 8PSK, QAM16, and QAM64. Keeping with the RML2016.10A Dataset, the samples per symbol of the root raised cosine filter were fixed at 8. The one sided filter span in symbols is varied uniformly from 7 to 10 with a step size of 1. The roll-off factor of the root raised cosine was varied uniformly from 0.34 to 0.36 with a step size of 0.01. For the channel model, the modulated signal was subjected to AWGN and given a center frequency offset as described by (5.1) to simulate errors in the receiver's signal detection stage [48] . The power of the AWGN is calculated using $E_s/N_0$ and varied uniformly from 0 dB to 20 dB with a step size of 2. The center frequency offset, which was normalized to the sample rate, is swept uniformly from $-1\%$ to $1\%$ with a step size of $0.2\%$ [2]. A subset of the data is shown in Figure 3.7.

### 3.3.4 Training Results

The network is implemented in PyTorch and trained using an NVIDIA 1080 GPU with the Adam [81] optimizer. The batch size used is 1024 when the network is trained with Dataset A and 512 when trained with Dataset B due to the increased example sizes. Models trained on Dataset A use dropout for regularization, as was initially proposed in [8]; however, models

[2]While $\pm 1\%$ is used in the current work as the range of center frequency offsets for Dataset B, [48] showed that DNNs could generalize over even wider ranges when performing AMC; however, the focus of the current work is to examine how even minute center frequency offsets can have negative impacts on the adversary.

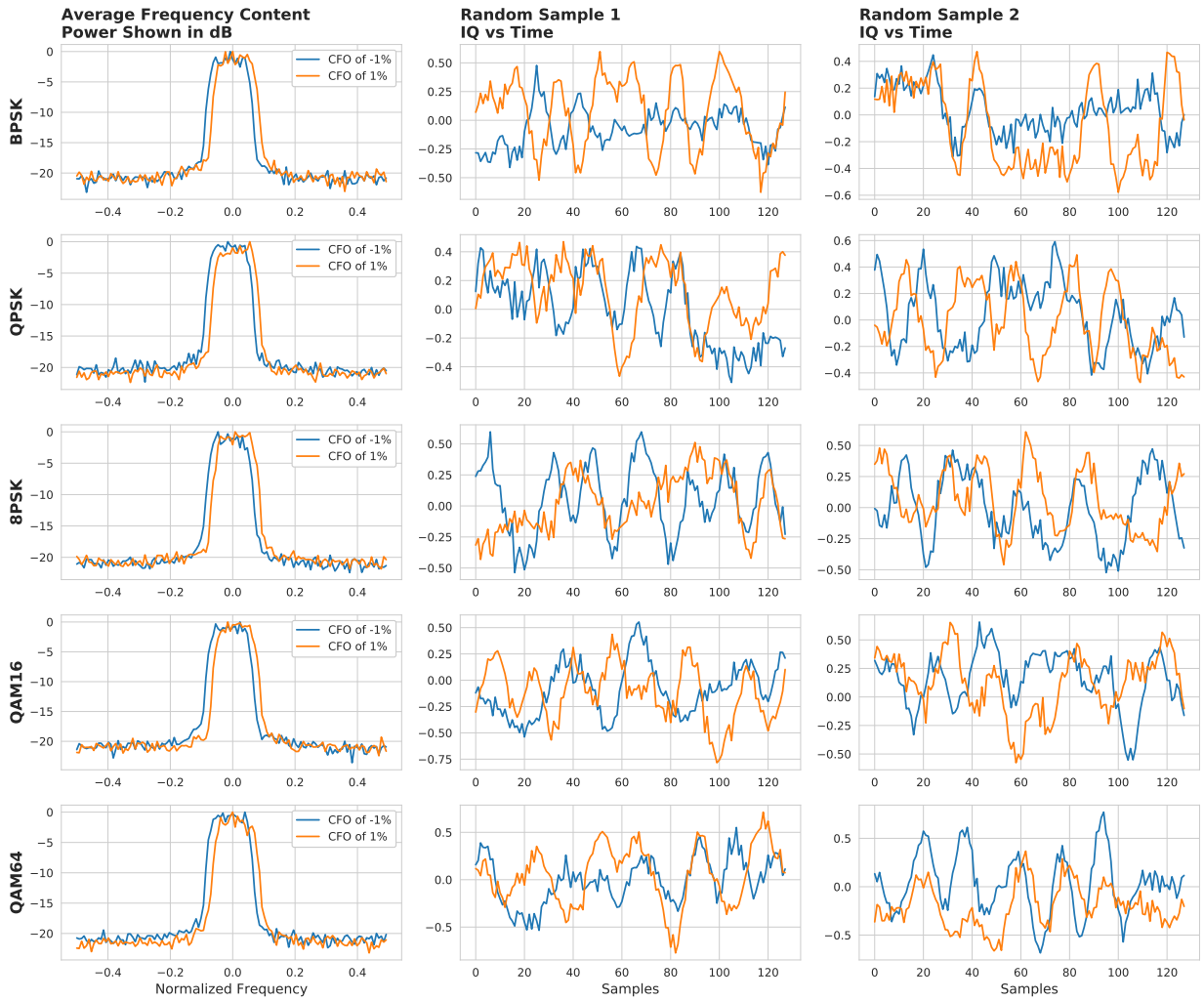


**Figure 3.7:** Random samples from the 128 sample version of Dataset B. The $E_s/N_0$ was restricted to 20 dB but the specific examples in time were selected randomly from that subset. The center frequency offset is shown by averaging the frequency content across all samples with $-1\%$ offset and $1\%$ offset.

trained on Dataset B use Batch Normalization as this increased training stability for the larger example sizes. For all models, the learning rate is set to 0.001 and early stopping is employed with a patience of 5.

During training, 30% of the dataset was withheld as a test set. The remaining 70% of the data is used in the training sequence with 5% of the training set used as a validation set. All data is split randomly with the exception that modulation classes and SNR are kept balanced for all sets. Each of the models is then evaluated at each SNR in the test set for overall accuracy and the results are shown, for Dataset A, in Figure 3.8, and for Dataset B in Figure 3.9. As expected, increasing the input size lead to increasing accuracy across all SNR ranges for Dataset B. The peak accuracy for Dataset B is higher, even for the equivalent 128 sized input network, than Dataset A, likely because there are more modulations to confuse the network with in Dataset A. While the specific SNR cannot be compared between the two test results in Figure 3.8 and 3.9 because the underlying dataset use a different measure of SNR, all Figures in future chapters present results in terms of $E_s/N_0$.

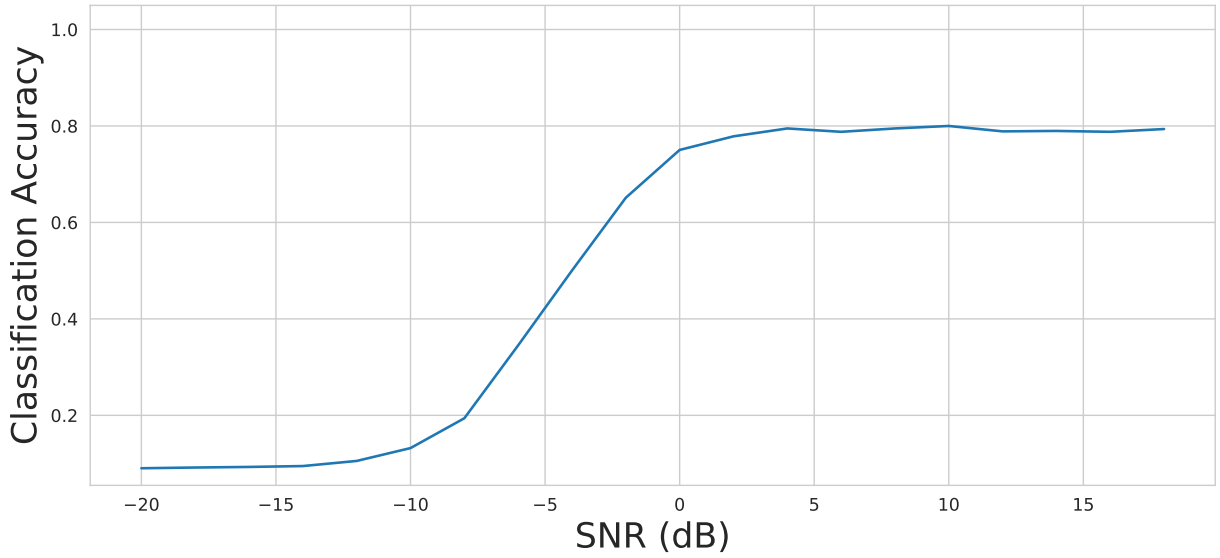


**Figure 3.8:** Dataset A test accuracy vs SNR.

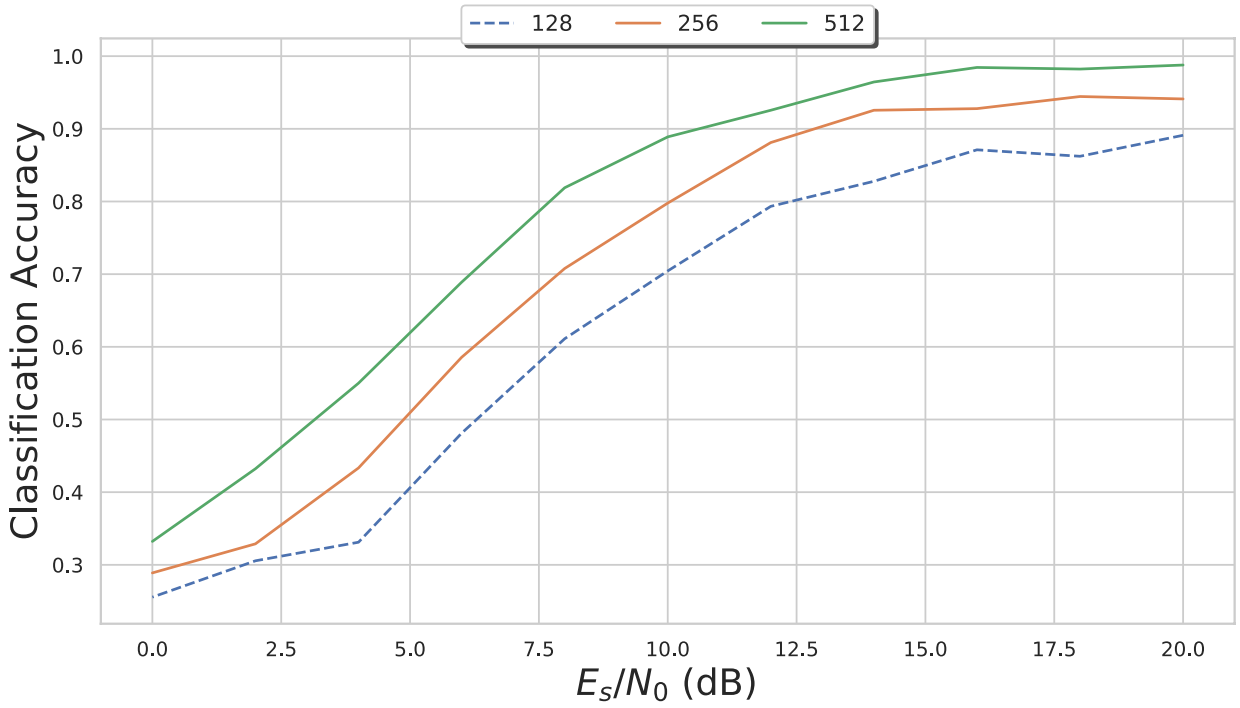


**Figure 3.9:** Dataset B test accuracy vs SNR for three different DNNs input sizes. In this Figure, there are no center frequency offsets during evaluation. As expected, increasing the input size results in increasing test accuracy over the entire SNR range studied.

# Chapter 4

# Direct Access Evasion Attacks

The focus of the current work is to develop an OTA adversarial evasion attack against RFML. This chapter makes a step towards that goal by studying the effectiveness of the FGSM attack, which is one methodology for adversarial evasion attacks presented in CV, in an environment that is the closest analogy to that presented in [39], with direct access to the classifier.

This chapter first introduces the concept of adversarial machine learning for untargeted evasion attacks and describes how FGSM creates adversarial examples. It then adapts FGSM to be bounded by a power ratio, as is common in wireless communications, instead of by a distance in the feature space, as is common in CV. Using this adaptation, a baseline evaluation is performed to confirm that FGSM is effective against a RFML model. The remainder of the chapter is dedicated to studying the effect that input sizes and noise have on the adversarial attack. By studying randomly drawn individual examples in Section 4.5, and the effect that AWGN has on them, the current chapter offers a fine grained look at some effects that could be encountered in an OTA attack. This chapter then concludes and these effects are studied more broadly for an FGSM attack in Chapter 5.

## 4.1 Introduction to Adversarial Machine Learning

Most raw IQ based signal classifiers seek to take in a signal snapshot, $\boldsymbol{x}$, and output the most probable class $\boldsymbol{y}$. Traditionally, $\boldsymbol{x}$ would represent a single channel of complex samples, with little pre-processing performed, and could therefore be represented as a two-dimensional ma-

trix [IQ, number of samples]. Specifically, RFML systems, which use DNNs, learn a mapping from the data by solving

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{L}(f(\boldsymbol{\theta}, \boldsymbol{x}), \boldsymbol{y})), \tag{4.1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ represent the training inputs and target labels respectively and $f$ represents the chosen network architecture. A loss function ($\mathcal{L}$), such as categorical cross entropy, is generally used in conjunction with an optimizer, such as stochastic gradient descent or Adam [81], to train the DNNs and thus learn the network parameters $\boldsymbol{\theta}$. While training the model, the dataset is fixed (assuming no data augmentation) and is assumed to be sampled from the same distribution that will be seen during operation of the RFML system.

Untargeted adversarial machine learning is simply the inverse of this process. By seeking to maximize the same loss function, an adversary can decrease the accuracy of a system. Therefore, the adversary is also solving an optimization problem that can be defined by the following [1].

$$\underset{\boldsymbol{x}^*}{\operatorname{argmax}}\, \mathcal{L}(f(\boldsymbol{\theta}, \boldsymbol{x}^*), \boldsymbol{y})) \tag{4.2}$$

In this case, the parameters, $\boldsymbol{\theta}$, of the classifier are fixed but the input, $\boldsymbol{x}^*$, can be manipulated. Many approaches exist to solve this problem [27, 33, 35, 39, 82]. In particular,

[1] Adversarial machine learning is generally constrained such that $||\boldsymbol{x}^* - \boldsymbol{x}||_p \leq \epsilon$ where $p$ can be $0, 1, 2, \infty$. In this thesis, the constraint is formulated in terms of the power ratio between the perturbation and the underlying transmission.

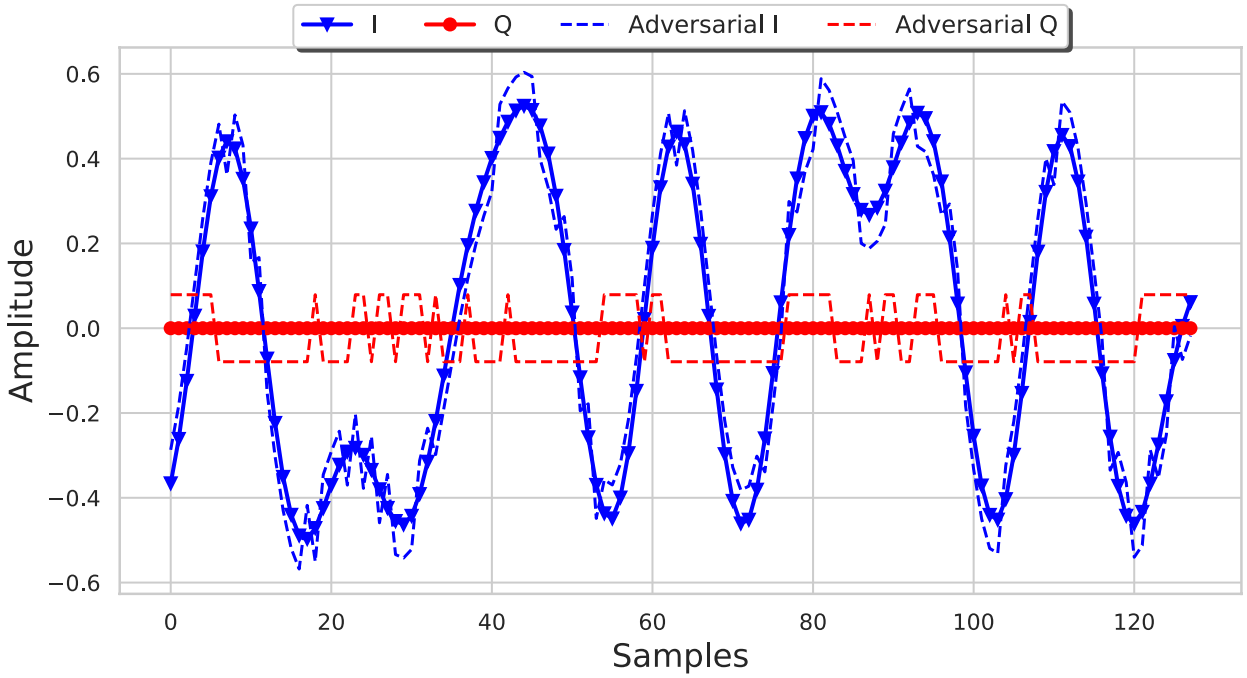


**Figure 4.1:** BPSK adversarial example with a 10 dB ($E_s/E_j$) perturbation, created with the FGSM [39] algorithm, applied.

FGSM [39] creates untargeted adversarial examples using

$$\boldsymbol{x}^* = \boldsymbol{x} + \epsilon \times \text{sign}(\nabla_x \mathcal{L}(f(\boldsymbol{\theta}, \boldsymbol{x}), \boldsymbol{y})), \tag{4.3}$$

where $\boldsymbol{y}$ represents the true input label and $\nabla_x$ represents the gradient of the loss function with respect to the original input, $\boldsymbol{x}$. This methodology creates adversarial examples constrained by a distance, $\epsilon$, in the feature space in a single step. $\boldsymbol{x}^*$ is referred to as an adversarial example. One adversarial example used in the current work is presented in Figure 4.1, where the source modulation is BPSK and a perturbation has been applied to achieve untargeted evasion for a direct access attack.

In the context of wireless communications, the absolute value of the signal is generally less important than the relative power of the signal with respect to some other signal such as noise. Therefore, similar to [43], the current work reformulates the perturbation constraint, $\epsilon$, from a distance bounding in the feature space to a bounding of power ratios in the following section.

## 4.2 Adapting FGSM

The average energy per symbol ($E_s$) of a transmission can be computed using

$$\mathbb{E}[E_s] = \frac{\text{sps}}{N} \sum_{i=0}^{N} |s_i|^2, \tag{4.4}$$

where sps represents samples per symbol, $N$ is the total number of samples, and $s_i$ represents a particular sample in time. Without loss of generality, the current work assumes the average energy per symbol of the modulated signal, $E_s$, is 1. Therefore, the power ratio of the underlying transmission to the jamming/perturbation signal[2] ($E_j$) can be derived as

$$\begin{aligned} \frac{E_s}{E_j} &= \frac{1}{E_j} \\ &= 10^{-E_j(\text{dB})/10} \end{aligned} \tag{4.5}$$

Since the input of $\text{sign}(\nabla_x)$ in (4.3) is complex, the output is also complex, and is therefore a vector whose values are $[\pm 1, \pm 1j]$. Therefore, the magnitude of each sample of the jamming

[2]Because the perturbation is an electronic signal deliberately crafted to impair the successful operation of the eavesdropper, the current work uses jamming signal and perturbation signal interchangeably.

signal can be computed as

$$\begin{aligned} |\operatorname{sign}(\nabla_x)| &= |\operatorname{sign}(z)| \\ &= \sqrt{(\pm 1)^2 + (\pm 1)^2} \\ &= \sqrt{2} \end{aligned} \tag{4.6}$$

Thus the energy per symbol of $\operatorname{sign}(\nabla_x)$ can be computed by plugging (4.6) into (4.4) resulting in

$$\begin{aligned} E_{\operatorname{sign}(\nabla_x)} &= \frac{\text{sps}}{N} \sum_{i=0}^{N} |\operatorname{sign}(\nabla_x)|^2 \\ &= 2 \times \text{sps} \end{aligned} \tag{4.7}$$

Because sps is fixed throughout transmission, a closed form scaling factor, $\epsilon$, can be derived to achieve the desired energy ratio $(E_s/E_j)$ by using

$$\begin{aligned} \epsilon &= \sqrt{\frac{\frac{E_j}{E_s}}{E_{\operatorname{sign}(\nabla_x)}}} \\ &= \sqrt{\frac{10^{\frac{E_j(\text{dB})}{10}}}{2 \times \text{sps}}} \end{aligned} \tag{4.8}$$

Plugging $\epsilon$ into (4.3) allows the creation of adversarial examples constrained by $E_s/E_j$ and can be succinctly defined as

$$\boldsymbol{x}^* = \boldsymbol{x} + \sqrt{\frac{10^{\frac{E_j(\text{dB})}{10}}}{2 \times \text{sps}}} \times \operatorname{sign}(\nabla_x \mathcal{L}(f(\boldsymbol{\theta}, \boldsymbol{x}), \boldsymbol{y})) \tag{4.9}$$

Constraining the power ratio in this way can be useful for evaluating system design trade-offs. Typically, a transmitter has a fixed power budget and the current chapter considers an adversarial machine learning technique which is not aware of the underlying signal; therefore, power which is used for the jamming signal subsequently cannot be used for the underlying transmission[3].

## 4.3 Baseline Evaluation

In order to first characterize the effectiveness of adversarial machine learning on raw IQ based AMC, a baseline study of average classification accuracy against $E_s/E_j$ was performed using the model trained on Dataset A. This attack was performed with no noise added to the adversarial examples, either before or after the perturbation was added, and thus assumes direct access to the classifier input. This represents the best case scenario for the classification network because, since there was no noise added, the SNR is infinite and therefore the model would be most accurate when it is not under attack. Additionally, this also shows the best case scenario for the adversary because the perturbation is also not distorted by noise. Ergo, this is the most ideal environment for both parties. Further, to more closely shadow the results shown in Chapter 5 and 6, which will be studied both in terms of classification accuracy and BER in a simulated OTA environment, the modulations are restricted to BPSK, QPSK, 8PSK, QAM16, and QAM64.

As can be seen in Figure 4.2, even at 30 dB, the FGSM attack is more effective than

[3] Methodology that takes into account BER is explored in Chapter 6.

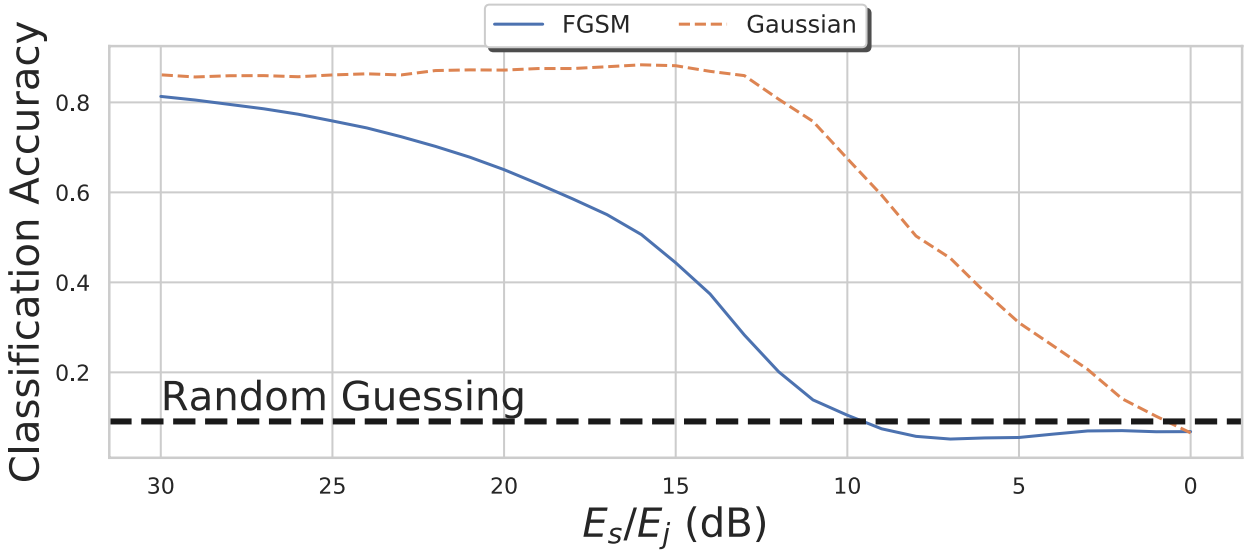


**Figure 4.2:** Classification accuracy of a model trained on Dataset A for a direct access attack. This plot compares the average classification accuracy for BPSK, QPSK, 8PSK, QAM16, and QAM64 when FGSM in used to apply a specific adversarial perturbation to the accuracy when "jammed" with a Gaussian noise signal at the same power ratio.

simply adding Gaussian noise (AWGN). At 10 dB, the FGSM attack is effective enough to degrade the classifier below the performance of random guessing. This represents an 8 dB improvement over the same degradation using Gaussian noise.

## 4.4 Attack Effectiveness versus NN Input Size

Increasing the DNNs input size has been empirically shown to improve the performance of raw IQ AMC in [48] as well as the current work's reproduction of similar results in Figure 3.9. While it is intuitive that viewing longer time windows of a signal will allow for higher

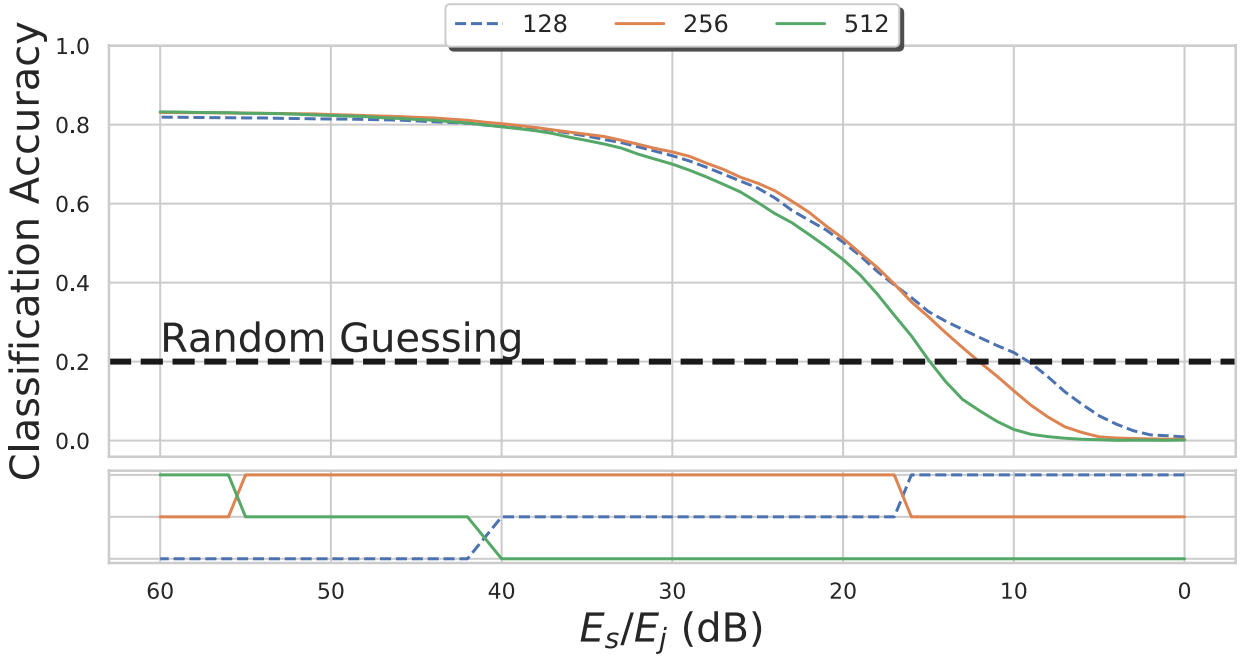


**Figure 4.3:** (top) Overall classification accuracy of models trained on Dataset B in the presence of a direct access FGSM attack for different input sizes. (bottom) The relative classification accuracy ranking of the three different models for each $E_s/E_j$.

classification accuracy (for static channels where psuedo-stationarity can still be assumed), it is also intuitive that allowing more adversarial jamming energy to enter the algorithm will have adverse effects. Therefore, the current work presents an experiment used to verify this intuition. Three copies of the same network, that differ only in input size, are trained on Dataset B. The analysis from the previous section is then repeated and shown in Figure 4.3.

As expected, at very high $E_s/E_j$, where the adversarial energy is low, the network with the largest input size is the most accurate. However, it is quickly supplanted by the second largest input size when $E_s/E_j$ drops below 55 dB ($\epsilon \approx 0.00044$). Once $E_s/E_j$ drops below 15 dB, the classification accuracy ranking inverts from the initial rankings, with the smallest input size

being the most accurate and the largest input size being the least accurate. Therefore, when developing a RFML system for use in adversarial environments, the benefits of increasing input size must be balanced against the cost of increasing the attack surface.

## 4.5 Analyzing Individual Adversarial Examples

While the earlier sections presented macro-level results, this section presents results at a micro-level by analyzing the fine grained effect of the adversarial machine learning method on individual examples rather than the average effect across multiple examples. The current work considers a single machine learning example from each of the source modulations[4]. For each example, $E_s/E_j$ is swept from 40 to 0 dB with a step size of 1 dB. At each $E_s/E_j$, the outputs of the DNN before the softmax function (as was shown in [39]) are captured.

One adversarial example for BPSK is shown in Figure 4.1. It can be seen in the $Q$ samples that, due to the sign operation in (4.9), the perturbation applied to the signal has a square shape. Therefore, the perturbation alone is easily identifiable; however, in the $I$ samples, where the underlying modulated signal also lies, it is less distinguishable. Notably, the differences are most apparent around the symbol locations (note that this signal has 8 samples per symbol), which could indicate that the symbol transitions are an important feature to the classifier.

[4]While random individual examples are analyzed for simplicity, the conclusions drawn are further explored in Chapter 5.

### 4.5.1 Difference in Logits

While the full output of the DNNs provides ample information, it is multi-dimensional and therefore hard to visualize. One metric that is often used is a confusion matrix, which captures the relationships among classes. However, confusion matrices are generally only presented as an average across multiple examples and do not provide any notion of the confidence with which a classifier made a single prediction. Therefore, a confusion matrix would not fully capture the variance of the DNNs because the outputs would not change unless the input examples were moved across a decision boundary. Another metric that could be used is to apply the softmax function to the output and report the confidence associated with the source class. This metric shows the variance of the classifier output but does not provide any indication of the Top-1 accuracy score because even a low confidence output could still be the highest and therefore the predicted class. The Top-1 accuracy score can be described as

$$\mathbb{E}[y_p = y_t] \tag{4.10}$$

where $y_p$ is the classification label obtained by taking the index of the maximum network output of the model.

$$y_p = \text{argmax}(\boldsymbol{y}) \tag{4.11}$$

Under the classification policy presented in (4.11), and the accuracy defined by (4.10),

Top-1 accuracy can also be described as the probability of correct classification.

The current work presents an additional metric, which we term the "difference in logits" ($\Delta_{logits}$), that simultaneously captures the accuracy of the classifier as well as the variance in outputs. "Logits" refers to the DNNs output before the softmax function has been applied. The maximum output of all incorrect classes is subtracted from the source (true) class output, which can be described by the following Equation.

$$\Delta_{logits} = y_s - \max(y_i \forall i \neq s) \tag{4.12}$$

The difference in logits can be visualized as the shaded region in the top of Figures 4.4 and 4.5. When $\Delta_{logits}$ is positive, the example is correctly classified and a negative $\Delta_{logits}$ therefore indicates untargeted adversarial success.

### 4.5.2 Classifier Output versus Attack Intensity

The output of the classifier for the BPSK example, across multiple $E_s/E_j$ is shown in Figure 4.4. At an $E_s/E_j$ of 10 dB, the jamming intensity present in Figure 4.1, untargeted misclassification is achieved because the BPSK output is not the highest output of the classifier; this result is also indicated by viewing that $\Delta_{logits}$ is negative. However, even though misclassification is achieved, the signal is still classified as a linearly modulated signal, with the predicted modulation order increasing as $E_s/E_j$ increased. Linearly modulated signals have symbols which exist in the IQ plane (distinguished as solid lines in Figure 4.4) versus a FSK

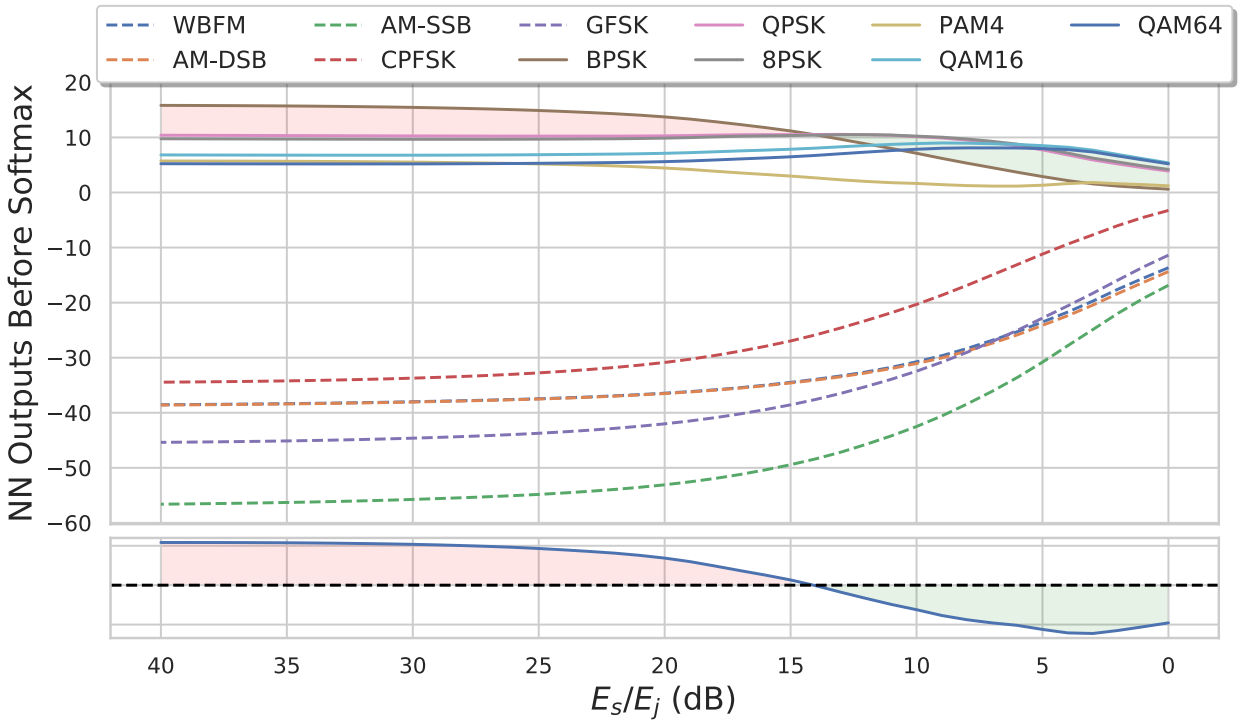


**Figure 4.4:** Output of the model trained on Dataset A for a direct access FGSM attack using a single, randomly selected, BPSK adversarial example across varying $E_s/E_j$ (top) and the corresponding difference in logits (bottom). The areas shaded red represent regions where a correct classification occurred (therefore the adversary was unsuccessful) while the areas shaded green represent an incorrect classification (therefore the adversary was successful). Note that the regions are only shaded to visualize (4.12).

or continuous signal (distinguished as dashed lines) whose symbols exist in the frequency domain or do not have discrete symbols at all, respectively. Therefore, while the adversarial machine learning method was able to achieve untargeted misclassification by causing the classifier to misinterpret the specific linearly modulated signal, the classifier still captured the hierarchical family of the human-engineered modulation. This reinforces the natural notion that the difficulty of targeted adversarial machine learning varies based on the specific source and target modulations used.

Figure 4.5 shows the output of the classifier for a single QAM16 example. As was observed in Figure 4.4, at very low $E_s/E_j$, where the attack intensity is the highest, the example is again classified as QAM (though untargeted misclassification is narrowly achieved because the model believes it is QAM64). Further, the QAM16 example required much lower energy ($E_s/E_j < 30$ dB) than the BPSK example ($E_s/E_j < 15$ dB) to achieve untargeted misclassification. Therefore, increasing the perturbation energy does not always provide advantageous effects from the evasion perspective, as can be observed from the difference in logits of Figure 4.5, and the optimal attack intensity varies between source modulations.

### 4.5.3 Mutation Testing with AWGN

In an OTA environment, an adversary would (almost) never be able to ensure that a transmitted signal isn't corrupted by some source of noise. Noise can come from many sources, including an adjacent transmitter, non-linearities in the eavesdropper or transmitter's RFFE, errors in the eavesdroppers signal detection stage, or simply the motion of electrons in the

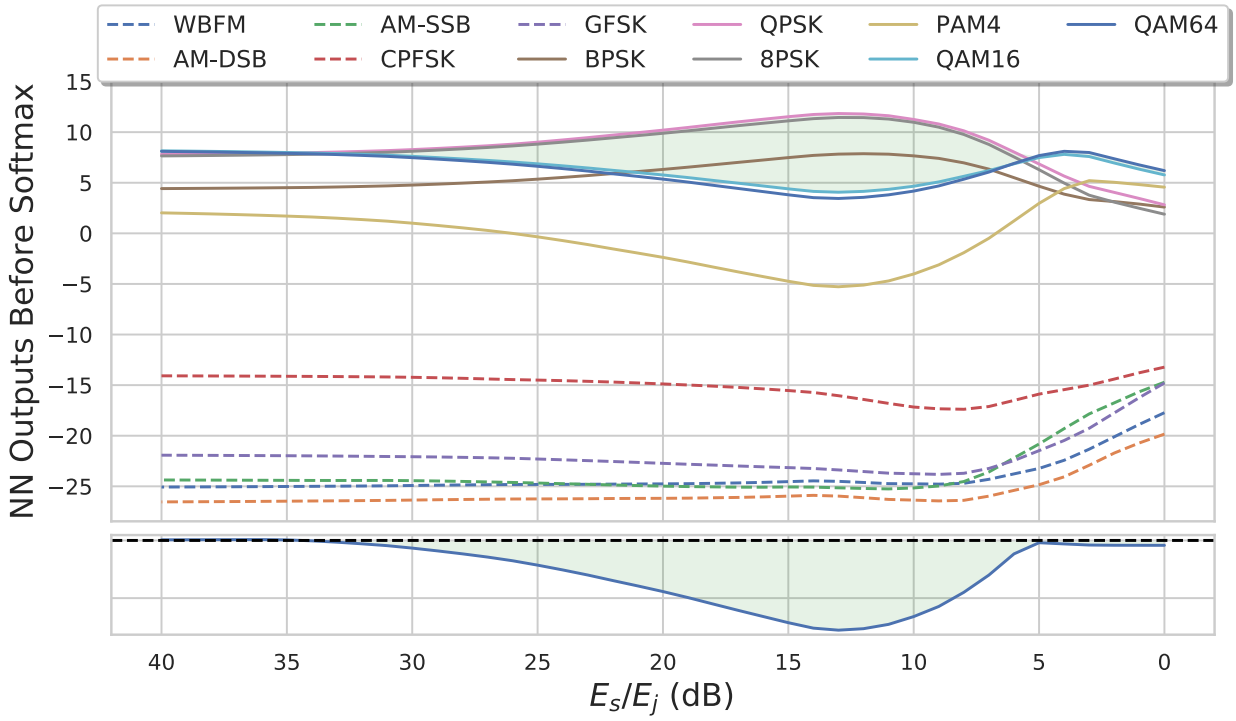


**Figure 4.5:** Output of the model trained on Dataset A for a direct access FGSM attack using a single, randomly selected, QAM16 adversarial example across varying $E_s/E_j$ (top) and the corresponding difference in logits (bottom). The areas shaded red represent regions where a correct classification occurred (therefore the adversary was unsuccessful) while the areas shaded green represent an incorrect classification (therefore the adversary was successful). Although it is a low confidence prediction, the classification is narrowly correct when $E_s/E_j > 35$(dB) and narrowly incorrect when $E_s/E_j < 5$(dB). With the model having trouble distinguishing between QAM16 and QAM64. Note that the regions are only shaded to visualize (4.12).

RFFE.

While having the adversarial example corrupted by noise is a certainty for OTA attacks, it was actually proposed as a defensive strategy in [42], named mutation testing, where the authors repeatedly applied domain specific noise to a machine learning example and calculated the input's sensitivity, with respect to the classifier output, in the presence of this noise. The authors of [42] found that adversarial examples were more sensitive to noise than examples contained in the initial training distribution and therefore mutation testing could be used to detect adversarial examples. It is important to note that the noise is added after the perturbation is applied and thus corrupts both the original example as well as the adversarial perturbation.

The current work presents a study of the effect of AWGN, one of the most prevalent models of noise in RFML, on randomly selected individual adversarial examples. For each $E_s/E_j$, AWGN is introduced to the signal at varying $E_s/N_0$ (SNR). $E_s/N_0$ is swept from 20 to 0 dB with a step size of 1 dB. For each of the SNRs considered, 1000 trials are performed. While $E_s/E_j$ and $E_s/N_0$ are the parameters swept in this experiment, the jamming to noise ratio ($E_j/N_0$) can be quickly inferred by

$$\begin{aligned} \frac{E_j}{N_0} &= \frac{E_s/N_0}{E_s/E_j} \\ &= \frac{E_s}{N_0}\text{dB} - \frac{E_s}{E_j}\text{dB} \end{aligned} \tag{4.13}$$

Again, results are presented in Figure 4.6 from the BPSK example originally shown in

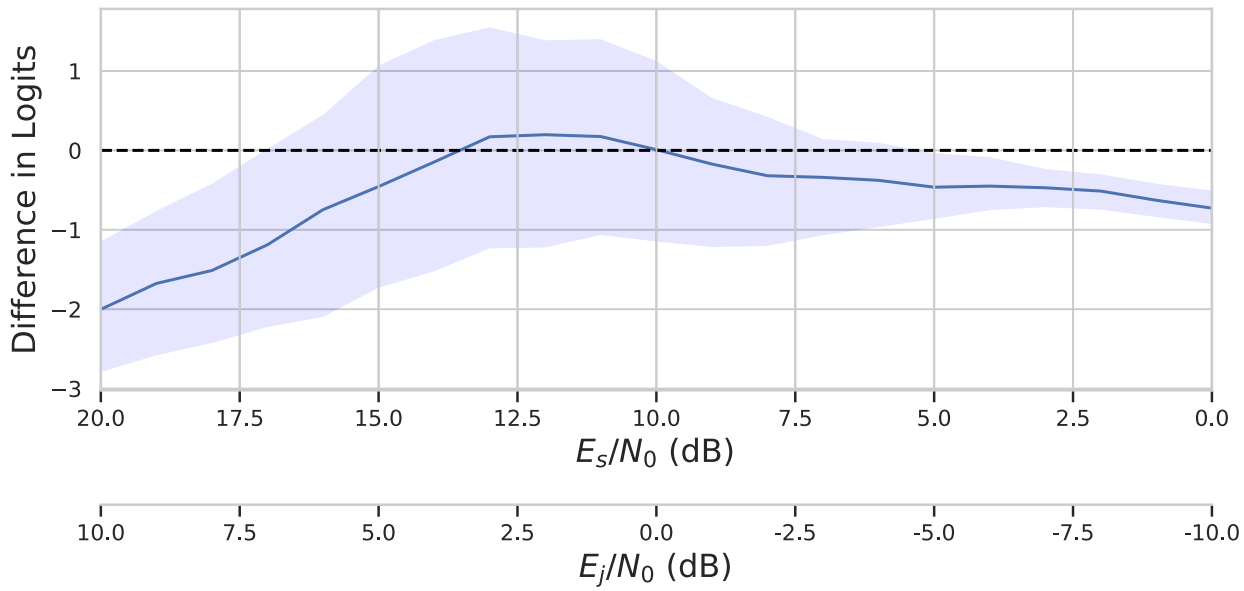


**Figure 4.6:** The effect of noise on the output of the model trained on Dataset A for a single, randomly selected, BPSK adversarial example with an $E_s/E_j$ of 10 dB. The line represents the mean of the difference in logits, at a specific $E_s/N_0$, while the shaded region represents the 25th and 75th percentiles in order to show the variance of the output.

Figure 4.1, where $E_s/E_j$ is 10 dB. The mean of the difference in logits is shown with the 25th and 75th percentiles shaded to show the variance in the output of the classifier at each SNR. With even a small amount of noise ($E_s/N_0$ of 17 dB) the 75th percentile of the difference in logits becomes positive indicating that the example was classified correctly in some iterations. Increasing the noise power to roughly half that of the applied perturbation ($E_j/N_0$ of 3 dB) results in the classification, on average, being correct.

This effect was not observed across all adversarial examples tested. In Figure 4.7 it is shown that, while the increased sensitivity of the classifier output is observed in the same range of $E_j/N_0$, it does not result in a correct classification. Therefore, while [42] presented

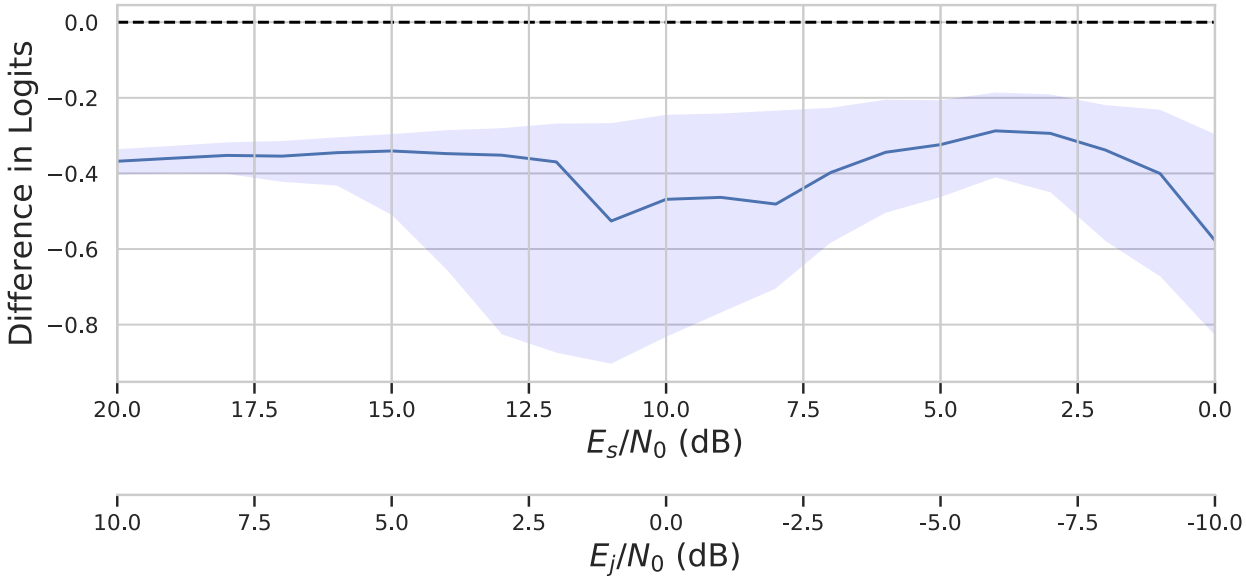


**Figure 4.7:** The effect of noise on the output of the model trained on Dataset A for a single, randomly selected, QPSK adversarial example with an $E_s/E_j$ of 10 dB. The line represents the mean of the difference in logits, at a specific $E_s/N_0$, while the shaded region represents the 25th and 75th percentiles in order to show the variance of the output.

general conclusions that all adversarial examples were sensitive to noise, these results show that this effect is most pronounced when the adversarial perturbation and noise have similar power. Additionally, these effects were not observed at all in the individual 8PSK and QAM16 examples studied.

Although only four random examples were studied in this section, it is clear that AWGN can greatly impact the classifier's output when applied to adversarial examples and, in some cases, that can lead to a reduction in adversarial success when the perturbation and noise power are at similar levels. Therefore, the current Chapter does not conclude that the vulnerabilities found in direct access attacks, which have deterministic access to the

classifier's inputs, automatically transfer to OTA attacks, where an adversary would only have access to the classifier's inputs through a stochastic wireless channel.

## 4.6 Conclusion

This chapter has shown a baseline result that deep learning based raw IQ AMC is vulnerable to untargeted adversarial examples when the adversary has direct access to the classifier's input. Further, it was shown that although increasing the DNNs input size can improve accuracy in non-adversarial scenarios, it can make a classifier more susceptible to deception for a given $E_s/E_j$. However, these results assumed direct access to the classifier input, which is unrealistic in an operational environment. A more realistic attack would only have access to the classifier's input through a stochastic wireless channel; therefore, this chapter studied the impact, on a fine grained level, of AWGN on the adversarial examples crafted using the FGSM algorithm and found that noise can have a negative impact on adversarial success. Therefore, the evaluations presented in this Chapter are not indicative of the adversarial success rates that would be achieved in an OTA attack. The following chapter performs an evaluation of identical methodology, but, in an OTA environment where sources of noise can negatively impact adversarial success and the perturbation applied to the transmission can negatively impact the communication to a cooperative receiver.

# Chapter 5

# Self Protect Evasion Attacks

All OTA attacks must consider the impact of receiver effects on adversarial success; furthermore, self protect attacks must balance the secondary goal of evading an adversary with the primary goal of transmitting information across a wireless channel. These effects have traditionally been ignored in prior work and therefore, while the previous chapter studied adversarial success in near perfect conditions, this chapter studies the impact to adversarial success when the examples are evaluated in the presence of three specific receiver effects, which would likely occur during an OTA attack: AWGN, sample time offsets, and center frequency offsets. While these effects are not exhaustive of all noise sources that could occur, the study of additional effects[1] is left to future work.

The work presented in this chapter does not consider the impact of noise on adversarial success or the impact of the perturbation to the intended receiver in the adversarial machine learning methodology. This chapter therefore uses the same FGSM method that was studied in Chapter 4, but, examines how that methodology would break down in an OTA attack at the eavesdropper or impact the intended communication at the receiver. In Chapter 6, methodology will be developed that specifically accounts for AWGN and sample time offsets at the eavesdropper, as well as the BER at the intended receiver.

[1] Additional sources of noise that would occur in a real system include: non-linearities in the transmitter's or eavesdropper's RFFE (due to amplifier distortion or inter-modulation effects), multi-path effects due to the propagation environment, quantization error (from the DAC on the transmitter or the ADC on the eavesdropper), or a fast fading channel that changes quick enough to impact the short time window of the signal that the eavesdropper uses for classification.

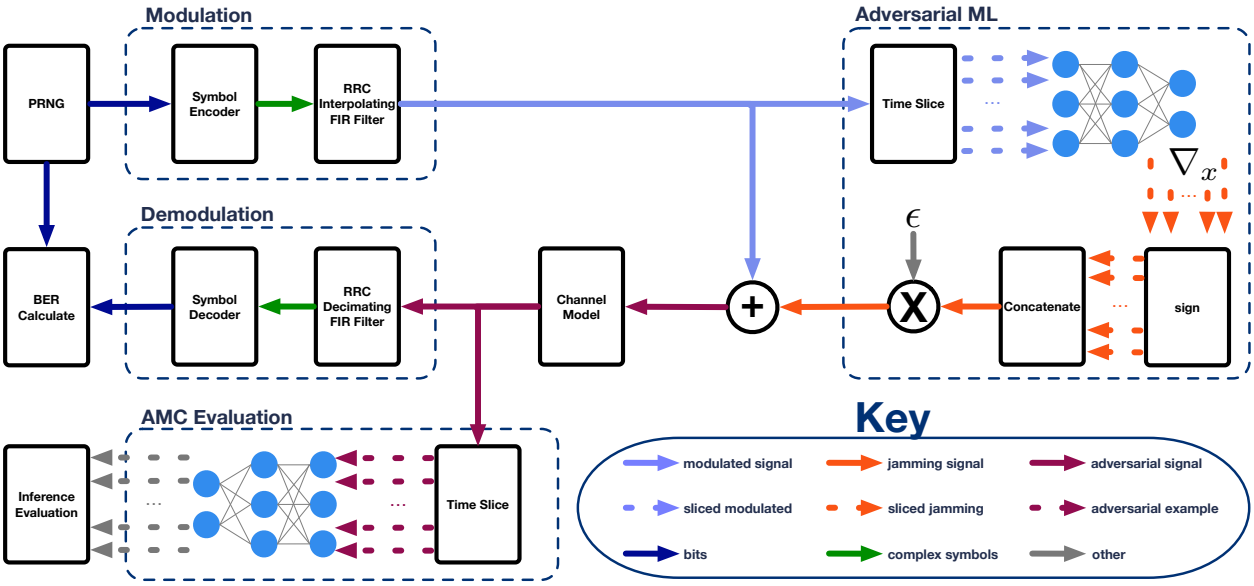


**Figure 5.1:** Block diagram of the evaluation methodology developed for the current work. The current work assumes perfect knowledge of the target DNNs and therefore the DNNs shown in the AMC Evaluation and Adversarial ML blocks are identical and simply separated for clarity.

## 5.1 Simulation Environment

The high level overview of the simulation environment used in the current chapter is shown in Figure 5.1 and each major block is described below. Full evaluation in the context of wireless communications requires the interfacing of both a DSP and ML framework. The current work uses GNU Radio and PyTorch respectively; however, the methodology is not dependent upon use of those frameworks in any way.

### 5.1.1 Modulation

The initial modulated signal is generated by a simple flow graph in GNU Radio. Unless otherwise stated, the parameters for transmission can be summarized as follows. The symbol constellations used are BPSK, QPSK, 8PSK, and QAM16. The root raised cosine filter interpolates to 8 samples per symbol using a filter span of 8 symbols and a roll-off factor of 0.35. 1000 examples[2] modulation scheme are created using a random bit stream.

The FGSM methodology used in the current (and previous) chapter does not depend on the properties of the signal in any way, but, the signals should closely resemble the training distribution of the AMC model under test in order to isolate the effects of adversarial machine learning from the effects of changing the test distribution.

### 5.1.2 Adversarial ML

In order to craft the jamming signal using adversarial machine learning techniques, it is necessary to first slice the signal into discrete examples matching the DNNs input size. Before feeding these examples into the DNNs, dithering is employed to add small amounts of noise to the examples. While dithering is a standard process in signal processing, it was specifically used in the current work because the sign operation in PyTorch is defined such

[2]1000 examples per class corresponds to 16000 random symbols per modulation in the current Chapter. This is because the AMC model studied operates on 128 complex samples which are 8 times over sampled, and therefore view 16 (128/8) symbols per inference. While 1000 examples are created as a baseline signal, due to sweeping parameters for the channel model and performing multiple trials at each characterization in order to sample the random process, the actual number of examples evaluated is much higher.

that sign(0) = 0. Therefore, for a BPSK signal, sign($\nabla_x$) would always be 0 for quadrature samples, which are always 0 by definition of BPSK. Restricting the perturbation only to the in-phase samples was not the goal of the current work and dithering eliminated this issue. After dithering, the FGSM algorithm is then used to create the perturbations which are concatenated back together to form the jamming signal. For each $E_s/E_j$ studied, the jamming signal is scaled linearly using (4.8) and added to the modulated signal. Unless otherwise stated, $E_s/E_j$ is swept from 0 to 20 dB with a step size of 4 dB.

### 5.1.3 Channel Model

The current work considers a simple, and static, channel model with AWGN and center frequency offsets. While the channel model is fixed for each observation, the parameters of this model is swept throughout the current chapter in order to evaluate the effectiveness of adversarial machine learning in multiple scenarios.

AWGN models thermal noise in the receiver and implicitly considers the path loss between the transmitter and eavesdropper by varying the height of the noise floor (as $E_s$ is fixed at 1 in the current work). Center frequency offsets model errors in the signal detection and isolation stage that would occur because the eavesdropper is performing blind signal classification and is therefore not synchronized[3] to the underlying transmission.

[3]The channel model encapsulates errors due to the eavesdropper not being synchronized to the transmitter in the frequency domain. An additional effect, due to not being synchronized in time, is explored in Chapter 5.4 but is not represented in the channel model because it only occurs when the signal is split into discrete examples.

The received signal can be characterized as follows:

$$s_{\text{rx}}(t) = e^{-j2\pi f_o t} s_{\text{tx}}(t) + \mathcal{CN}(0,\, \sigma^2) \tag{5.1}$$

Where $f_o$ is the normalized frequency offset and $\sigma^2$ is given by the desired $E_s/N_0$. The channel model is implemented using a GNU Radio flow graph.

### 5.1.4 Demodulation

Demodulating the received signal, at the intended receiver, consists of match filtering, down-sampling to one sample per symbol, and decoding the symbols back into a bit stream to verify the data received matches the data transmitted. The demodulation is also implemented as a GNU Radio flow graph and assumes both symbol and frame synchronization.

### 5.1.5 Automatic Modulation Classification Evaluation

Top-1 accuracy is the metric used for eavesdropper classifier evaluation in [8], [9], and [48] and is the metric used for evaluation in the current work. For untargeted adversarial machine learning, adversarial success is defined as a lower Top-1 accuracy as opposed to a higher accuracy.

## 5.2 Impact of Additive White Gaussian Noise

AWGN has been shown to negatively impact both BER and classification accuracy. Additionally, as discussed in Section 4.5, AWGN can have a negative effect on adversarial success. While section 4.5 studied this impact on individual examples, this section further evaluates these negative effects with a larger scale study in order to validate that the intuition gained from section 4.5 generalizes across all examples. In some cases, such as in "rubbish examples" [39] or "fooling images" [38], the primary goal of adversarial machine learning may simply be to create an input that is classified with high confidence as some target class starting from a noise input. However, in most practical applications, fooling a classifier is a secondary goal that must be balanced against the primary objective. In CV, this primary objective is to preserve human perception of the image. In the current work, the primary objective of self protect attacks is to transmit information to a friendly receiver using a known modulation while the secondary objective is to avoid recognition of that modulation scheme by an eavesdropper. Therefore, this section presents results showing the compounding impacts of adversarial machine learning and AWGN on BER as well as the effect of AWGN on adversarial success rates.

Using the model trained on Dataset A, a range of $E_s/N_0$ and $E_s/E_j$ are considered. For each $E_s/N_0$ considered, ten thousand trials are executed to provide averaging of the random processes present in the channel model for a given random signal. The current work considers both the BER and classification accuracy for BPSK in Figure 5.2, QPSK in Figure

5.3, 8PSK in Figure 5.4, and QAM16 in 5.5.

Unsurprisingly, increasing the adversarial perturbation energy has positive effects on adversarial success rates (also shown previously in Chapter 4) and negative effects on BER. In order to directly compare the trade space between the two across a range of SNRs, BER versus classification accuracy is plotted for each $E_s/E_j$ considered. At high SNR, extremely low probabilities of bit error, such as those seen in BPSK at $E_s/N_0 = 20$ dB, are hard to characterize empirically due to computation constraints. Therefore, in the BER versus classification accuracy plots, all results with lower than $10^{-6}$ BER have been omitted for clarity. Conversely, not all BERs are attainable in the SNR range studied when $E_s/E_j$ becomes small; therefore, when comparing BER vs classification accuracy not all BERs will necessarily be defined.

By looking at Figure 5.2, one can observe that classification accuracy can be degraded to $\approx 0\%$ with no noticeable effect to BER for BPSK when using a white-box adversarial attack with an $E_s/E_j$ of 4 dB. While this is a very strong result, it only occurs at high SNRs ($> 15$ dB). A more reasonable result to compare to would be the baseline result at 10 dB. In order to achieve the same BER as the baseline of no attack (shown as a dashed line), an adversary must increase their SNR, and therefore their transmission power, by $\approx 2$ dB when performing an adversarial attack at an $E_s/E_j$ of 8 dB. A similar analysis can be performed for QPSK (Figure 5.3) where a 4 dB increase to SNR is required to maintain the same BER while reducing classification accuracy to $< 20\%$.

As stated in Chapter 4, AWGN can have negative effects on adversarial success. There-

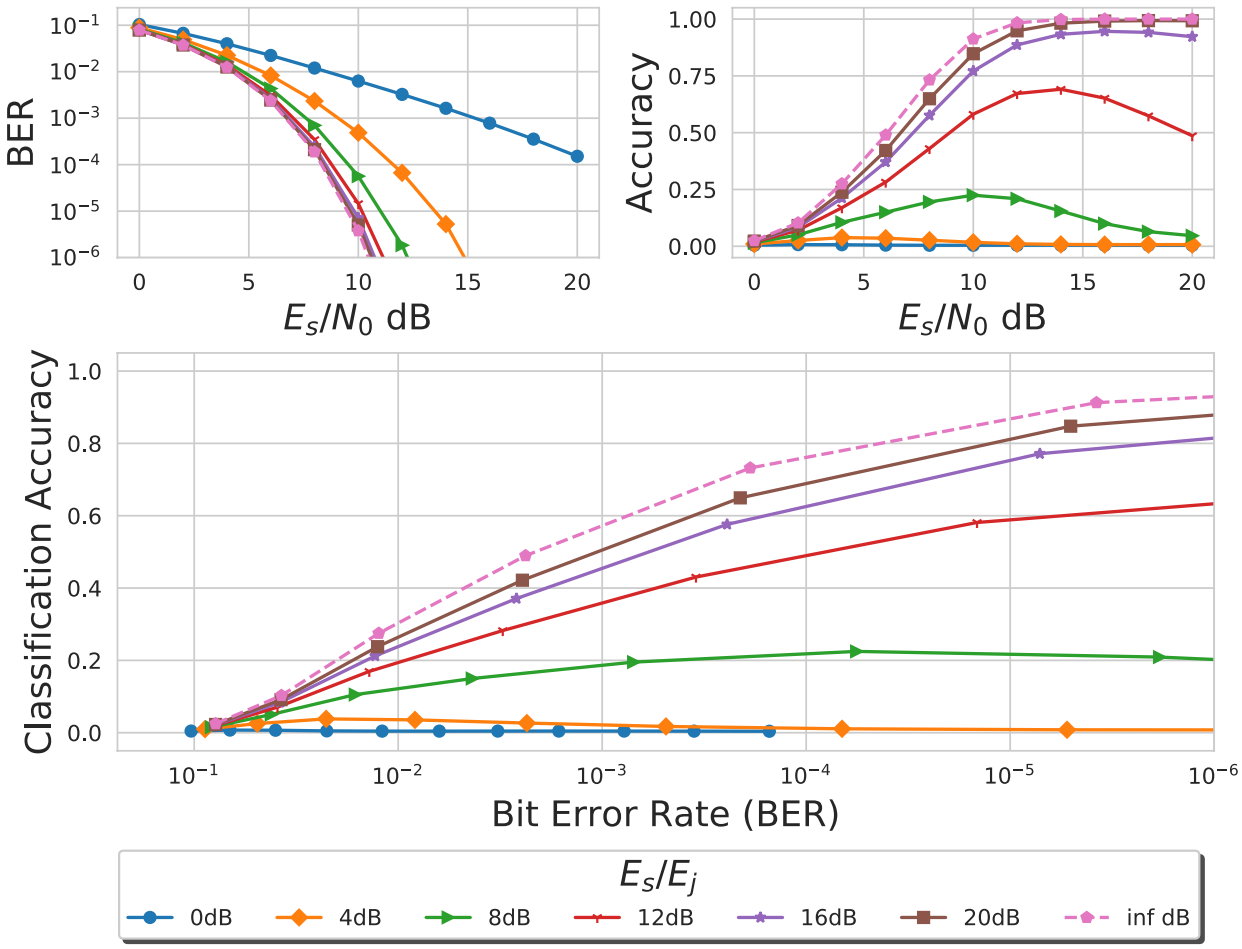


**Figure 5.2:** Classification accuracy and BER at varying $E_s/E_j$ and $E_s/N_0$ for self protect untargeted adversarial attacks using FGSM on the model trained with Dataset A and a source modulation class of BPSK. Note that when using a high powered perturbation, where $E_s/E_j$ is a small value, not all BERs are attainable in the SNR range studied. In all plots, the adversary desires to have lower curves because lower BER implies a lower impact to the underlying communication and a lower accuracy implies untargeted adversarial success. Conversely, the eavesdropper desires to have higher curves which indicate a larger increase in BER for an adversary to evade signal classification.

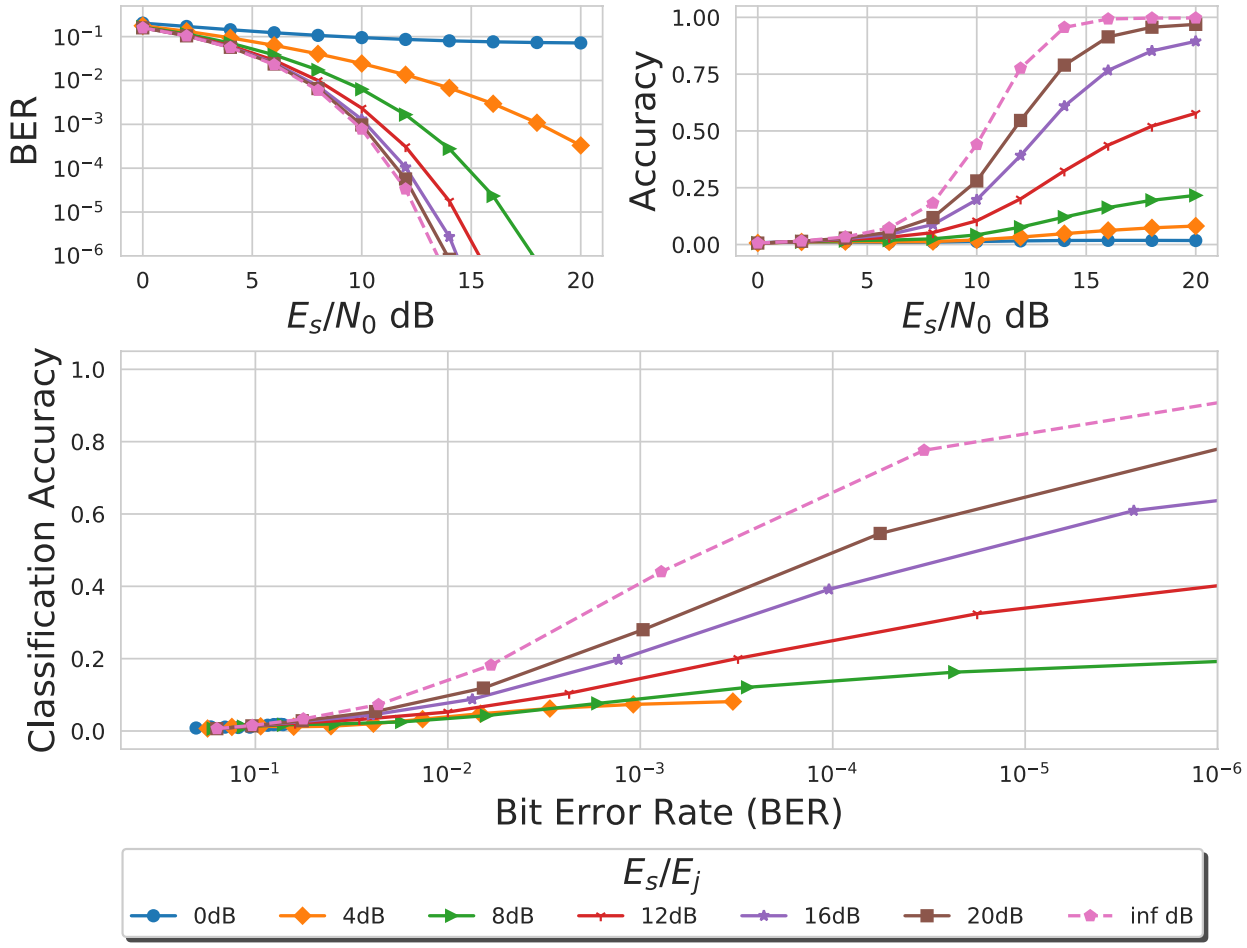


**Figure 5.3:** Classification accuracy and BER at varying $E_s/E_j$ and $E_s/N_0$ for self protect untargeted adversarial attacks using FGSM on the model trained with Dataset A and a source modulation class of QPSK. Note that when using a high powered perturbation, where $E_s/E_j$ is a small value, not all BERs are attainable in the SNR range studied. In all plots, the adversary desires to have lower curves because lower BER implies a lower impact to the underlying communication and a lower accuracy implies untargeted adversarial success. Conversely, the eavesdropper desires to have higher curves which indicate a larger increase in BER for an adversary to evade signal classification.

fore, while an eavesdropper with a high SNR would be fooled nearly all of the time by a BPSK transmission with an $E_s/E_j$ of 8 dB, an eavesdropper with an $E_s/N_0$ of 10 dB would still classify this signal correctly 20% of the time. If an adversary wished to attain 0% classification more generally for BPSK using FGSM, then they would need to transmit with an $E_s/E_j$ of 4 dB. This attack intensity would require an SNR increase of $\approx 4$ dB to maintain the same BER. The increased accuracy, at lower SNRs, observed previously in Figure 4.4 can also be observed in Figure 5.2 and therefore generalizes across BPSK examples. This effect can also be observed, to a lesser extent, in the results of 8PSK (Figure 5.4) and QAM16[4] (Figure 5.5). Thus, even when evaluating adversarial success purely in terms of accuracy, the adversary can be significantly impaired by an AWGN channel and must contribute more power to the perturbation in order to achieve the same success rates as a Direct Access Evasion Attack (Chapter 4).

As previously mentioned, reducing an eavesdropper's classification accuracy is a secondary goal that must be balanced against the ability to successfully transmit information. Therefore, as modulation order increases, BER can become prohibitive for an adversary's success. For instance, FGSM attacks using source modulations of 8PSK and QAM16, with $E_s/E_j \leq 8$ dB, already contain bit errors without any added noise. Therefore, degrading classification accuracy of 8PSK below 20%, outside of the eavesdropper receiving the signal

[4]Although QAM16 shows an accuracy increase at lower SNRs for adversarial examples, this is also observed in the baseline case (although it is less pronounced). Therefore, while it can't be conclusively stated that accuracy increases at lower SNRs is unique to adversarial examples for QAM16, it can clearly be concluded that an adversary would be less successful when the eavesdropper has a low SNR.

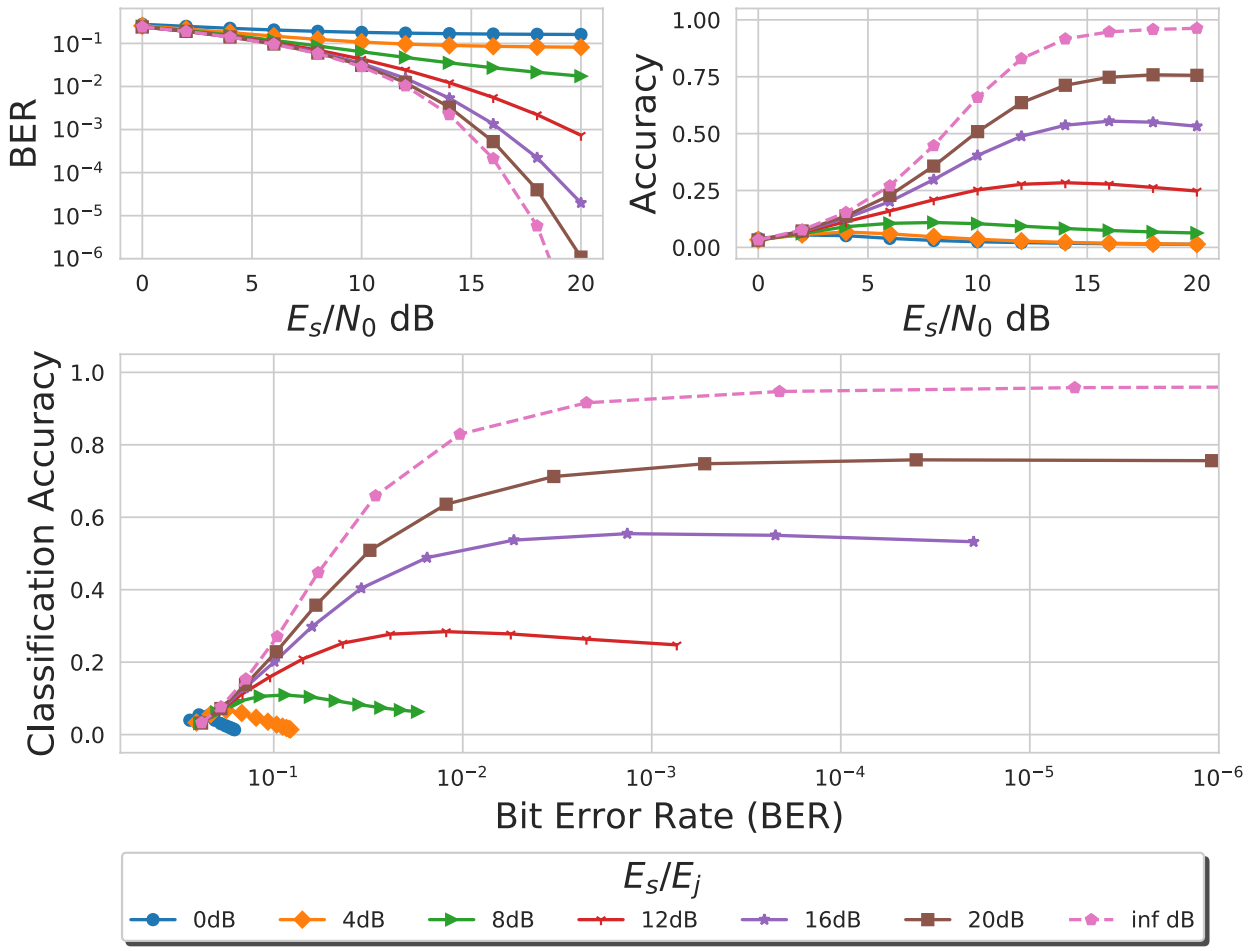


**Figure 5.4:** Classification accuracy and bit error rates at varying $E_s/E_j$ and $E_s/N_0$ for self protect untargeted adversarial attacks using FGSM on the model trained with Dataset A and a source modulation class of 8PSK. Note that when using a high powered perturbation, where $E_s/E_j$ is a small value, not all BERs are attainable in the SNR range studied. In all plots, the adversary desires to have lower curves because lower BER implies a lower impact to the underlying communication and a lower accuracy implies untargeted adversarial success. Conversely, the eavesdropper desires to have higher curves which indicate a larger increase in BER for an adversary to evade signal classification.

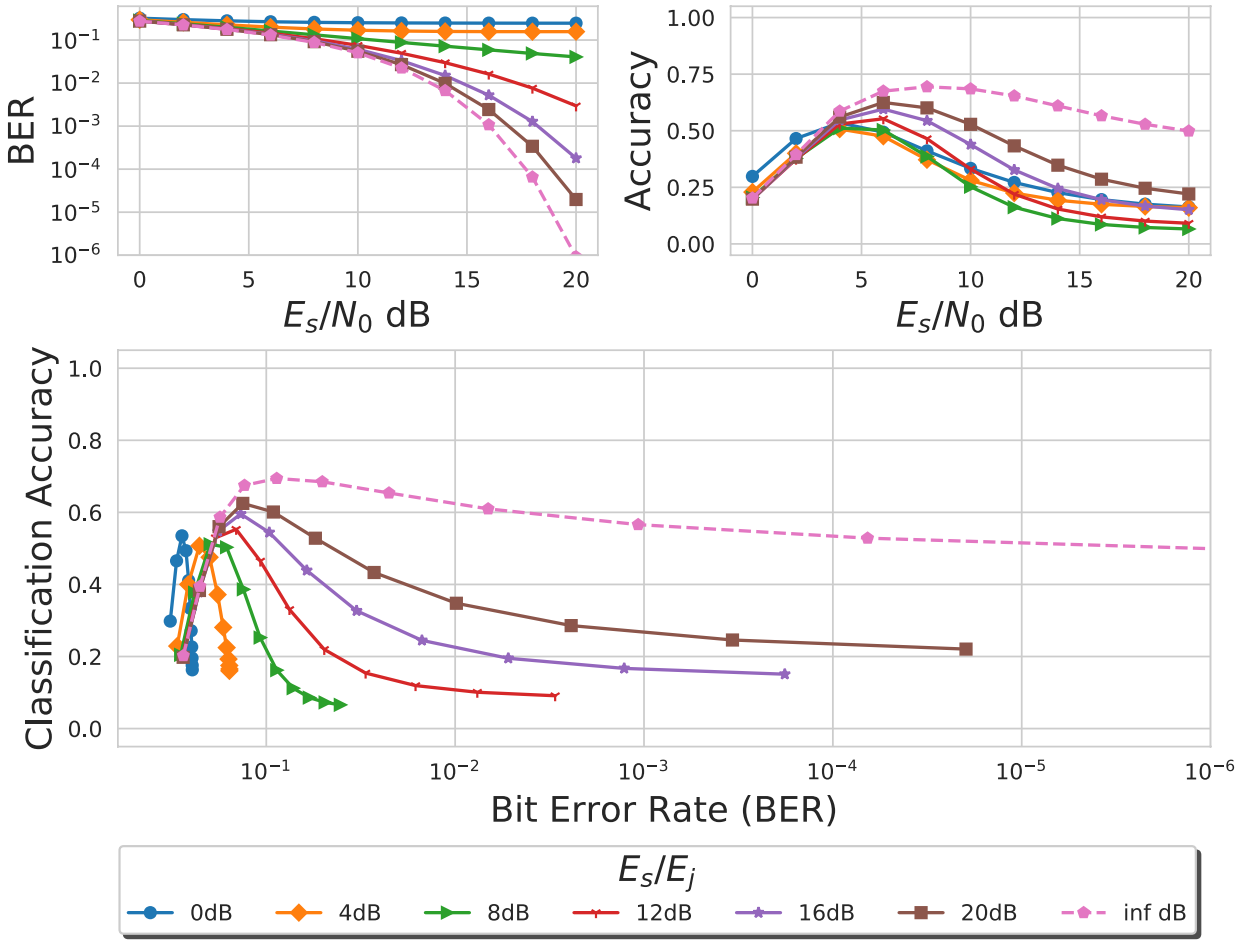


**Figure 5.5:** Classification accuracy and bit error rates at varying $E_s/E_j$ and $E_s/N_0$ for self protect untargeted adversarial attacks using FGSM on the model trained with Dataset A and a source modulation class of QAM16. Note that when using a high powered perturbation, where $E_s/E_j$ is a small value, not all BERs are attainable in the SNR range studied. In all plots, the adversary desires to have lower curves because lower BER implies a lower impact to the underlying communication and a lower accuracy implies untargeted adversarial success. Conversely, the eavesdropper desires to have higher curves which indicate a larger increase in BER for an adversary to evade signal classification.

at low SNR, would require forward error correction to account for the errors in transmission. In the case of QAM16, attacks using $E_s/E_j \leq 4$ dB would impact the receiver more than the eavesdropper in many scenarios. Specifically, QAM16 has a BER of $\approx 16\%$ and $\approx 25\%$ when $E_s/E_j$ is 4 and 0 dB respectively even when there is no additive noise. Therefore, when evaluated as a function of BER, the classification accuracy is actually lower in the baseline case than under the presence of these high intensity attacks.

These results conclude that adversarial machine learning is effective across multiple modulations and SNRs to achieve the goal of untargeted misclassification because, for a given BER, classification can be greatly reduced in many scenarios. However, avoiding signal classification may require sacrificing spectral efficiency or increasing transmission power to maintain the same BER. Additionally, AWGN was shown to have a negative impact on adversarial success rates in 3 out of 4 source modulations tested and therefore adversarial machine learning can be the most effective at high SNRs.

## 5.3 Impact of Center Frequency Offsets

Signal classification systems typically do not know when and where a transmission will occur. Therefore, they must take in a wideband signal, detect the frequency bins of the signals present, as well as the start and stop times of transmission, and bring those signals down to baseband for further classification. However, this process is not without error. One effect shown in [48] was the consequences of errors in center frequency estimation, resulting in

frequency offset signals. The authors of [48] found that raw IQ based AMC only generalized over the training distribution it was provided and therefore if additional frequency offsets outside of the training distribution were encountered, the classification accuracy would suffer. Because these estimations are never exact, adversarial examples transmitted over the air must also generalize over these effects. This section, as well as the following section for a different effect, simply evaluates whether an FGSM attack does in fact generalize over these effects and does not modify the methodology.

In order to evaluate the impact of center frequency offsets to adversarial examples, it is necessary to use a model that has been trained to generalize over these effects. Therefore, this experiment uses Dataset B, which has a training distribution consisting of $\pm 1\%$ frequency offsets, which have been normalized to the sample rate. An input size of 128 is used for closer comparison to other results using Dataset A, which only has 128 as an input size. The frequency offsets are swept between $-2.5\%$ and $2.5\%$ with a step size of $0.1\%$. $E_s/N_0$ is evaluated at 10 and 20 dB. At each SNR, 100 trials are performed to average out the effects of the stochastic process. The results of this experiment are shown in Figure 5.6.

It can be observed that the baseline classifier has learned to generalize over the effects of frequency offsets within its training range of $\pm 1\%$; however, the adversarial examples are classified with $\approx 10\%$ higher accuracy even at the lowest evaluated frequency offsets of $\pm 0.1\%$. This effect is observed at both 20 and 10 dB SNR. Therefore, even minute errors in frequency offset estimation can have negative effects on adversarial machine learning and must be considered by adversarial generation methods. Yet, even though frequency offsets

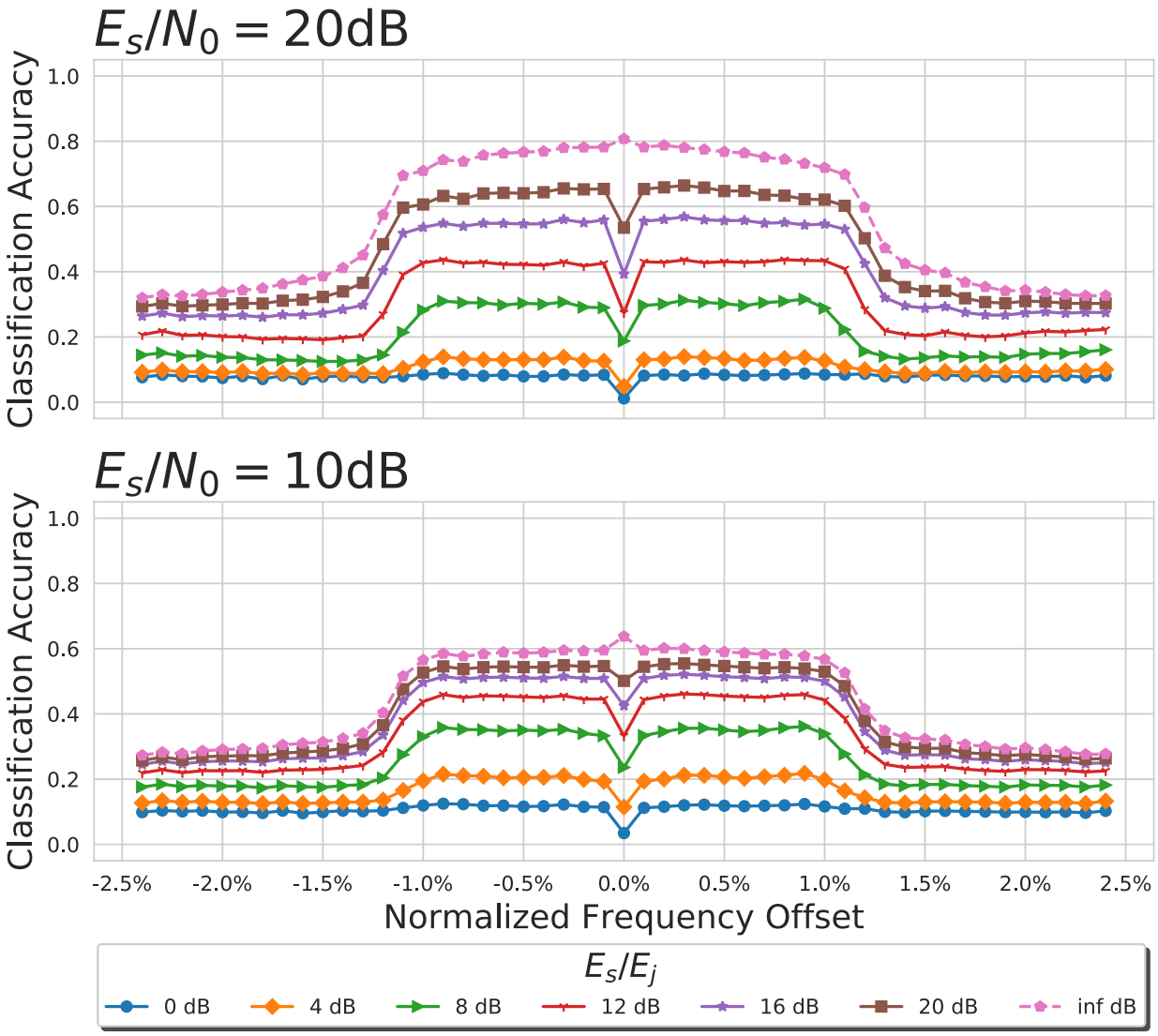


**Figure 5.6:** Classification accuracy vs normalized center frequency offset at varying $E_s/E_j$ for self protect untargeted adversarial attacks using FGSM. The model used is trained on Dataset B with an input size of 128. This dataset has a training distribution of $\pm 1\%$ frequency offset that has been normalized to the sample rate.

impede adversarial success, an FGSM attack is still more successful at evasion than the baseline case of not modifying the signal at all.

## 5.4 Impact of Timing Offsets

An additional effect that could be encountered is sample time offsets. In the context of communications, sample time offsets can be thought of as a rectangular windowing function, used for creating discrete machine learning examples, not aligning between the adversarial perturbation crafting, at the transmitter, and signal classification, at the eavesdropper. As previously mentioned, the signal classification system must estimate the start and stop times of a transmission; one way to estimate these times is to use an energy detection algorithm where the power of a frequency range is integrated over time and then thresholded to provide a binary indication of whether a signal is present. A low threshold could have a high false alarm rate and a high threshold could induce a lag in the estimation of the start time. Furthermore, signal classification systems could use overlapping windows for subsequent classifications to increase accuracy through the averaging of multiple classifications of different "views" of a signal or use non-consecutive windows due to real-time computation constraints. Therefore, this effect is a near certainty. The current section simply evaluates whether an FGSM attack generalizes over this effect and does not modify the adversarial methodology. Chapter 6 will present methodology that does not rely on a discrete window for crafting a perturbation in order to generalize over this effect.

This experiment uses the model trained on Dataset A and again evaluates the effect at an $E_s/N_0$ of 10 and 20 dB. At each SNR, 100 trials are performed. The time offset is modeled as a shift in the starting index, from the starting index used at the transmitter for creating the adversarial perturbations, that is used when slicing the signal for evaluating the signal classification performance and non-overlapping/consecutive windows are still used. The time offset was swept from 0 to 127 (because the input size is 128 and this effect is periodic in the input size); however, only the results from 0 to 10 are shown for simplicity. Time offsets higher than 8 samples, the symbol period, did not present any significant additional impairments beyond those seen at 8. The results are shown in Figure 5.7.

As expected, the network is not heavily effected in the baseline case. However, the adversarial examples can be significantly impacted. In the case of an $E_s/E_j$ of 12 dB, simply shifting the time window to the right by four samples can increase the classification accuracy by 20%. Additional energy can be dedicated to the adversarial perturbation to partially overcome this effect. In Figure 5.7 it can be observed that attacks with an $E_s/E_j$ of 4 dB or 0 dB have less than 10% accuracy increase from this effect. However, as stated in prior sections, increasing the perturbation power of an FGSM attack negatively impacts BER and therefore becomes prohibitive for higher order modulations. While some adversarial perturbations have been shown to be agnostic to these time shifts, such as the UAP [33] attack considered in [43], all evaluations of adversarial machine learning in the context of RFML, that seek to model OTA attacks, must assume this effect exists and generalize over it.

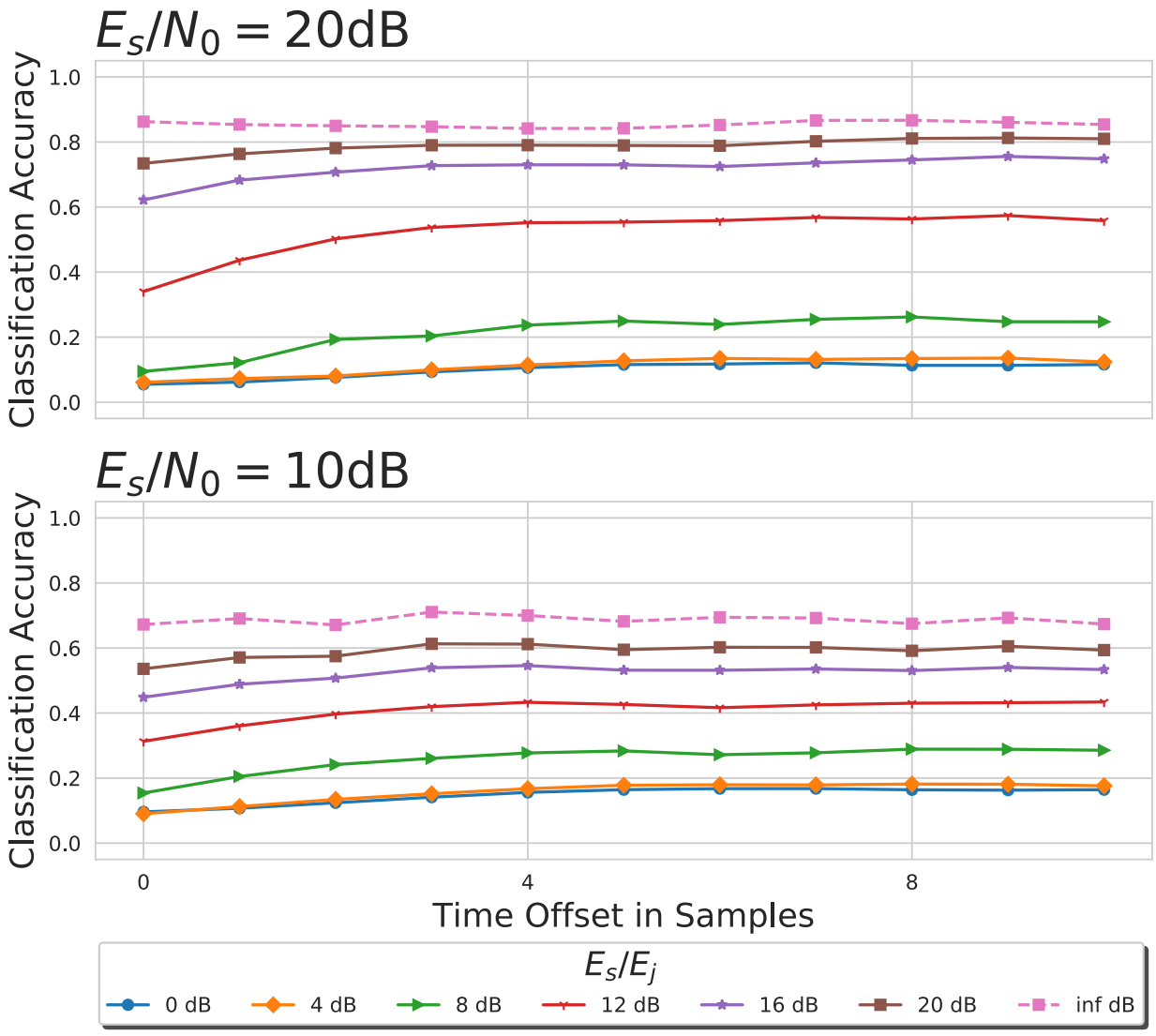


**Figure 5.7:** Classification accuracy vs time window offsets at varying $E_s/E_j$ for self protect untargeted adversarial attacks using FGSM. The model used is trained on Dataset A.

## 5.5 Conclusion

The current chapter has demonstrated that RFML systems are vulnerable to OTA adversarial evasion attacks; however, these attacks are not as effective as an attack with Direct Access to the classifier, as was demonstrated in Chapter 4. This Chapter has proven this by evaluating multiple example attacks against a raw IQ deep learning based modulation classifier and examined the effectiveness of an FGSM attack in the presence of three RFML domain specific effects that would occur at an eavesdropper: AWGN, sample time offsets, and center frequency offsets. When evaluating OTA attacks, evading an eavesdropper is generally a secondary goal and must be balanced against the primary goal of transmission, which is to communicate information across a wireless channel. Therefore, the current chapter showed that these attacks harmed the eavesdropper more than the adversary by demonstrating that, for a given BER, classification accuracy could be lowered for the majority of the OTA attacks considered. Given these results, it is logical to conclude that similar vulnerabilities exist in all RFML systems when the adversary has white-box knowledge of the classifier.

Future OTA adversarial evasion attacks must consider their ability to generalize over RFML domain specific receiver effects as well as their their impact to the underlying transmission. The current chapter has demonstrated that all three sources of noise considered can degrade the adversary's ability to evade classification. Furthermore, the current chapter has shown that, while current adversarial methodology can be used for evading classification, especially when using a lower order source modulation such as BPSK, it may require

sacrificing spectral efficiency or increasing transmission power to maintain the same BER. The following chapter describes methodology that incorporates these effects and wireless communications goals directly into the adversarial methodology in order to create strong adversarial examples that generalize over receiver effects and have limited impact to the underlying transmission.

# Chapter 6

# Communications Aware Evasion Attacks

The previous chapters have demonstrated the use of adversarial RFML for untargeted OTA evasion attacks on raw IQ based AMC. However, the prior methodologies only consider the effect of adversarial machine learning on the underlying transmission as an evaluation metric (Chapter 5), don't consider it at all (Chapter 4 and [4], [43]), or actively seek to disrupt communication [44, 46]. Recent research has shown the promise of directly including the BER in the loss function [45], but, the proposed techniques require gradient computation for each example in order to craft an adversarial perturbation which requires a machine learning framework and thus makes deployment of these adversarial methodologies difficult. Further, the methodology in [45] operated on discrete blocks of signals and thus assumed time synchronization.

This chapter addresses both of the shortcomings of [45]. First, methodology is independently developed that directly accounts for the underlying transmission in the adversarial optimization problem. Second, the methodology presented in the current chapter directly accounts for sample time offsets and therefore does not depend on time synchronization. Finally, the learned model for perturbation creation is encapsulated in a fully convolutional adversarial residual network. Once the parameters of this network are learned, in an offline training process, they could be easily deployed as a complex non-linear filter in a communications system.

This chapter is organized as follows. First, a system model is presented, which describes the necessary modifications to a transmit chain. Second, the methodology is described, including the custom loss functions, the training procedure used to minimize those loss

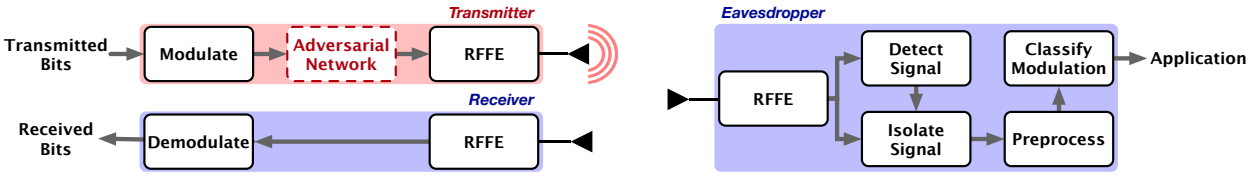


**Figure 6.1:** Overview of the system model assumed in the current chapter where the contributions are encapsulated in the "adversarial network" block.

functions, and the specific neural network architecture that is used to encapsulate the learned parameters during training. Finally, the results are provided, specifically showing the tradeoff between the ability to communicate with the ability to deceive, the BER and accuracy vs SNR, as well as a discussion on the additional bandwidth used by this methodology.

## 6.1 System Model

An overview of the system model for the current work is provided in Figure 6.1. The current work considers three major components of the system: an intelligent and mutable transmitter, which seeks to communicate with an immutable receiver implementation, while avoiding AMC by a deep learning enabled eavesdropper.

### 6.1.1 Transmitter

The primary goal of the transmitter is considered to be the communication of information to the receiver and BER is the pertinent metric used to evaluate fulfilment of this goal. The secondary goal of the transmitter is to evade classification of its unintended emission

to an eavesdropper. A methodology is developed for applying adversarial machine learning techniques to the signal before transmission.

Many adversarial techniques, such as the well known FGSM [39] require the computation of both a forward and backward pass (iterated variants such as [35] require multiple computations) in order to compute an added perturbation that causes a misclassification. Adversarial Transformation Network (ATN) are separate DNNs that can transform any input into an adversarial example. The authors of [30] presented two variants: Adversarial Auto-Encoder (AAE) and Perturbation - Adversarial Transformation Network (P-ATN). AAEs create adversarial examples through

$$\boldsymbol{x}^* = g(\boldsymbol{\theta}, \boldsymbol{x}), \tag{6.1}$$

where $g(\cdot)$ represents the AAE, $\boldsymbol{x}$ represents the initial input, and $\boldsymbol{\theta}$ represents the parameter matrices learned during the training of the AAE. The second variant of ATN, P-ATN, is very similar to the residual block introduced in [83] and creates adversarial transformations through

$$\boldsymbol{x}^* = \boldsymbol{x} + g(\boldsymbol{\theta}, \boldsymbol{x}) \tag{6.2}$$

Residual networks can more easily learn the identity function because it is easier to push a residual to 0 than to learn to directly replicate $\boldsymbol{x}$ on the output of the network (as would be needed in AAEs). Furthermore, the current work considers that the transmission is already optimal for the primary goal of communicating information (ergo the identity function would be optimal if only this goal was considered) and thus uses a P-ATN architecture. The current work will refer to P-ATN as an Adversarial Residual Network (ARN) for simplicity.

In order to provide the greatest control over the signal, the ARN is chosen to be applied at the highest sample rate in the system; thus, the ARN is the final step in the transmit chain and occurs after the pulse shaping filter.

### 6.1.2 Receiver

The receiver is assumed to be fixed, and therefore no modifications are made in the receive chain. The current work assumes that the receiver is synchronized to the transmitter and is demodulating the signal using a known modulation scheme, such as PSK or QAM, to extract the transmitted information.

### 6.1.3 Eavesdropper

The eavesdropper is modeled as a deep learning enabled blind modulation classifier operating on raw IQ with minimal pre-processing [8, 9]. Therefore, the eavesdropper has very limited *a priori* information about the transmission and must first detect when, in time, and where, in frequency, a transmission has occurred. It must then isolate that signal and bring it down to baseband before classifying the modulation. This process can introduce errors such as center frequency offsets or sample time offsets [48].

### 6.1.4 Threat Model

The current work describes a physical attack where the signal is perturbed at the transmitter and propagates through a wireless channel to a cooperative receiver and unintended eaves-

dropper. In order to trick the eavesdropper, the current work assumes full knowledge of the eavesdropper's neural network and signal processing chain in order to evade classification. Therefore, in the threat model provided by Figure 3.1, the current chapter would have a goal of self protect untargeted misclassification with knowledge of the target network architecture and parameter matrices.

Further, the current work is only concerned with modeling attacks on the classifier in Figure 6.1 and considers all other portions of the eavesdropper to be static and not react to the attack. Therefore, evaluating the cascading effects that the perturbation would have on the signal detection and isolation stage of the eavesdropper are left as future work.

## 6.2 Methodology

This section first describes the methodology needed to create and train an ARN for evading signal classification while maintaining the ability to communicate. First, the loss, or objective, function is presented. Then, the specific training procedure used to minimize that loss function is described. Finally, the exact ARN architecture used in the current work is presented. After training the ARN, its performance is then evaluated using a reference AMC model and receiver implementation. The reference model as well as the procedure used to perform the evaluations seen in Section 6.3 is described at the end of this section.

### 6.2.1 Loss Function

The objective of the ARN can be succinctly described as "minimize BER at the receiver and minimize classification accuracy at the eavesdropper." However, because the ARN in this work is trained using Adam [81], which is a gradient based method, a surrogate loss function needs to be used that is (at least pointwise) differentiable. Further, as there are multiple objectives for the ARN, the full loss function used is therefore a balance between the adversarial loss, $\mathcal{L}_{\text{adv}}$, that seeks to evade classification, and the communications loss, $\mathcal{L}_{\text{comm}}$, that seeks to minimize BER. This balance is controlled by a hyper-parameter, denoted in the current work as $\alpha$ [1]. Additionally, while the current work considers instantaneous transmit power to be a hard limit and thus the perturbation power is constrained before transmission, it also includes a perturbation power term in the loss function as a regularizer, $\mathcal{L}_{\text{pwr}}$. This regularization term is chosen to be a part of the communications loss because, as previously mentioned, the original communications is considered to be optimal, and therefore a lower power perturbation would undoubtedly correlate with lower communications loss because the symbols would not be perturbed. The hyper-parameter describing the trade off between true communications loss and power regularization is described as $\beta$ in the current work and

[1] As $\alpha$ is a hyper-parameter of the methodology presented in this Chapter, it should be chosen by the operator based on their high level objectives. For instance, an operator that places a high priority on communications would choose $\alpha$ to be a value close to 100% where as an operator that places a high value on evading signal classification would choose $\alpha$ to be a value close to 0%. The current Chapter later chooses 50% as a middle ground for further analysis.

is always 0.99. The full loss function can thus be described by

$$\mathcal{L}(\cdot) = (1-\alpha)\mathcal{L}_{\text{adv}}(\cdot) + \alpha[(\beta)\mathcal{L}_{\text{comm}}(\cdot) + (1-\beta)\mathcal{L}_{\text{pwr}}(\cdot)] \tag{6.3}$$

and each component of this loss function is described below. As most deep learning frameworks are designed to minimize a loss function, all elements of $\mathcal{L}(\cdot)$ are formulated as minimization problems. For ease of interpretation, all loss functions have been formulated to asymptotically approach 0; therefore, the most optimal ARN would achieve a loss, $\mathcal{L}(\cdot)$, of 0.

#### 6.2.1.1 Adversarial Loss

Traditional adversarial machine learning would seek to maximize the cross-entropy loss of target DNNs with respect to the true class of the adversarial example. In practice, this results in DNNs having lower confidences in the true class and thus misclassifications when the model becomes more confident in another class than the true one. However, this can't be used in the current work because cross-entropy loss becomes unstable and approaches $\infty$ as the confidence in the true class approaches 0, which is the region that is being optimized in the current work. Therefore, the current work does not use cross-entropy loss and instead uses the square of the model's confidence for the true (source) class, denoted as $p_s$, as a proxy for the same goal.

$$\mathcal{L}_{\text{adv}}(p_s) = p_s^2 \tag{6.4}$$

The confidence, $p_s$, is obtained by applying a softmax function to the output of DNNs. The objective of the ARN, in the current work, is to lower the confidence in the source class ($p_s$)

and thus to minimize $\mathcal{L}_{\text{adv}}$.

#### 6.2.1.2 Communications Hinge Loss

The primary objective of wireless communications is to transmit information without error. This information can be thought of as a random bit stream; however, the information is first encoded into a symbol space before being transmitted over the air where each symbol could encode multiple bits in order to increase data rates. Therefore, when defining a loss function for a physical layer wireless communication, it would be logical to simply use the mean squared error in the symbol space. However, a typical wireless receiver would use a hard decision, based on the nearest possible symbol, to decode each symbol back to its corresponding bits. Thus, if the error in the symbol space did not result in an incorrect hard decision, it would have no impact on receiver performance. If one assumes an AWGN channel and a synchronized receiver, as the current work does, then the probability of incorrectly decoding a symbol for any given SNR depends only on the distance between the transmitted symbol and its decision boundary. Naturally, the decision boundaries are different for each source modulation. Higher order modulations have lower distances between symbols and are thus more sensitive to AWGN.

The current work does not directly use distance in the symbol space for deriving the probability of bit error in order to remain agnostic of the channel model. Instead, the current work introduces what is termed a "Communications Hinge Loss" that can be used to empirically penalize bit errors when they occur during training. First, an indicator variable

is created to describe symbol error.

$$I_s = \begin{cases} 0 & \text{HD}(S_{rx}) = \text{HD}(S_{tx}) \\ 1 & \text{otherwise} \end{cases} \tag{6.5}$$

In (6.5), $\text{HD}(S_{rx})$ represents the hard decision made by the receiver to decode the received symbol, $S_{rx}$, and $\text{HD}(S_{tx})$ represents the correct interpretation of the transmitted symbol, $S_{tx}$. Thus, $I_s$ is only non-zero when a bit error would occur. In higher order modulation schemes, a single symbol can encode multiple bits, and therefore an error in decoding a symbol can result in multiple bit errors. For simplicity, $I_s$ does not encode the number of bit errors that occurred. As gray coding is used in the current work, the most likely number of bit errors, per symbol error, is 1; thus $\mathbb{E}[I_s]$ is related to BER through

$$\frac{\mathbb{E}[I_s]}{\log_2 M} \approx \text{BER} \tag{6.6}$$

where $M$ is the order of the modulation and $\log_2 M$ denotes the number of bits per symbol.

Note that $S_{rx}$ and $S_{tx}$ are complex valued. The current work denotes their error vector magnitude (EVM) as $|S_{rx} - S_{tx}|$. Further, the current work uses normalized constellations that have an average energy per symbol, $E_s$, of 1. Therefore, the EVM of received symbols will generally be much smaller than 1 and $\text{EVM}(\cdot)^2 \leq \text{EVM}(\cdot)$. Additionally, the current work uses the observation that $\mathbb{E}[S_{rx}] = S_{tx+j}$ where $S_{tx+j}$ is the transmitted symbol (after the perturbation has been applied) and uses $\text{EVM}(S_{tx}, S_{tx+j})$ as a noiseless proxy for

$\text{EVM}(S_{tx}, S_{rx})$. Finally, the communications hinge loss is defined as

$$\begin{aligned}\mathcal{L}_{\text{comm}}(S_{tx}, S_{tx+j}, I_s) = \text{EVM}(S_{tx}, S_{tx+j})^2 \\ + I_s \times \text{EVM}(S_{tx}, S_{tx+j}),\end{aligned} \tag{6.7}$$

which will heavily penalize bit errors that occur during training while only lightly penalizing the movement of the transmitted symbol if it does not disrupt communication. Intuitively (6.7) is a *nearly* ideal loss function for minimizing BER. First, as previously mentioned in (6.6), $I_s$ is almost perfectly correlated with BER. However, as its derivative is always 0, it cannot directly be used in most gradient descent algorithms. Thus, the addition of EVM ensures that $\mathcal{L}_{\text{comm}}$ is smoother, has a derivative that is typically non-zero, and provides an easily implemented surrogate for maximizing the distance between symbols. Specifically, for the interior symbols of QAM, minimizing EVM is equivalent to maximizing the distance between symbols in the constellation. For PSK and the outer symbols of QAM, EVM provides a good approximation of maximizing the distance, but, as power can be added through the addition of the perturbation, the symbols could be moved further from the origin and thus further from all other symbols. This potential behavior is not encapsulated by (6.7) and should be considered in future work.

#### 6.2.1.3 Perturbation Power Regularization

As will be discussed in Sections 6.2.2 and 6.2.5, the instantaneous power of the ARN is always hard limited. However, a loss term that penalizes the average perturbation power, $E_j$, in terms of its power ratio, $E_s/E_j$, is included for regularization of the ARN output.

Each ARN is allowed a power budget, $L$, and the perturbation power regularization seeks to greatly penalize exceeding that limit with no penalty for using less power. This regularizing loss term is provided by

$$\mathcal{L}_{\text{pwr}}(\frac{E_s}{E_j}, L) = \max(0, L - \frac{E_s}{E_j})^2 \tag{6.8}$$

### 6.2.2 Training Implementation

An overview of the training implementation is provided in Figure 6.2 with each major component, as well as the specific procedure, described below. All elements of the training procedure are implemented in PyTorch and, with the exception of the initial AMC training, executed on a CPU (a GPU was not used purely to allow parallel training of multiple networks).

#### 6.2.2.1 Transmitter

The transmit chain in training consists of sampling random symbols from a constellation, upsampling the symbol stream to achieve the desired samples per symbol (sps), which is 8 in the current work (due to the usage of an open source dataset which only contains eight times over sampled signals), and then pulse shaping the resulting signal.

#### 6.2.2.2 Root Raised Cosine Filter

Both the pulse shaping and match filter are implemented using a convolution operation with the taps of the convolution set as a root raised cosine filter with a half-sided span of 8 symbols

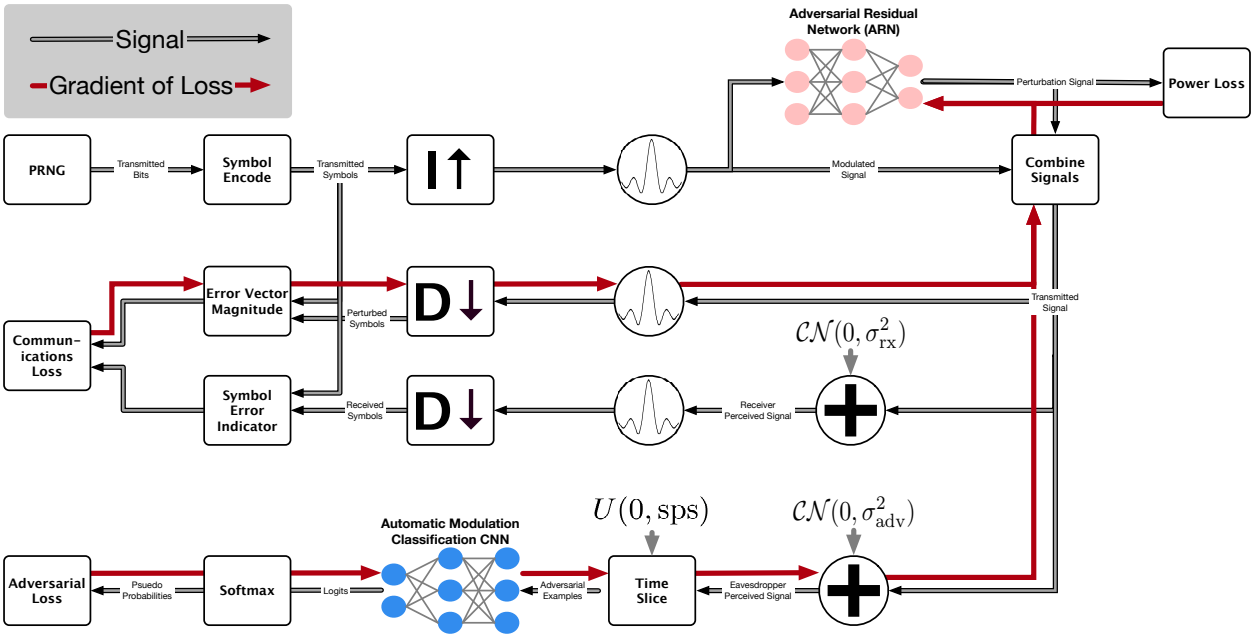


**Figure 6.2:** Training procedure used in the current work. All elements are implemented in PyTorch. The forward pass, or the "signals", are shown in black. The backward pass, or the "gradient of the loss", is shown in red. In order to extract the symbols, $S_{tx+j}$, the transmitted signal is match filtered and then downsampled to achieve one sample per symbol. The symbol error indicator, $I_s$, is created by applying AWGN to the signal before performing the same filtering and downsampling process to extract the received symbols, $S_{rx}$, which can then be compared with the transmitted symbols, $S_{tx}$, to compute $I_s$. The signal used for AMC is also passed through an AWGN channel but the power of the noise, $\sigma^2_{\text{adv}}$, is varied independently of the power of the noise at the receiver, $\sigma^2_{\text{rx}}$. When creating discrete adversarial examples to pass through the AMC model, the starting index is varied uniformly to ensure that adversarial success does not depend on time synchronization.

and excess bandwidth of 0.35. Using convolutions with separate upsampling/downsampling was due simply to ease of implementation in PyTorch.

#### 6.2.2.3 Combining Signals

To ensure that the perturbation remained below a specified power limit, $L$, the perturbation was first normalized to have a maximum component of 1.

$$\boldsymbol{s}_j^* = \frac{g(\boldsymbol{\theta}, \boldsymbol{s}_{tx})}{\max(|\,\mathbb{R}(g(\boldsymbol{\theta}, \boldsymbol{s}_{tx}))|, |\,\mathbb{C}(g(\boldsymbol{\theta}, \boldsymbol{s}_{tx}))|)} \tag{6.9}$$

In (6.9), $\boldsymbol{s}_{tx}$ represents the signal that would have been transmitted if the ARN was not present, $g(\cdot)$ represents the outputs of the ARN, and $\boldsymbol{s}_j^*$ represents an intermediate, unscaled, perturbation signal. The final perturbation signal, $\boldsymbol{s}_j$, could then be created with the assurance that $E_s/E_j > L$ through

$$\boldsymbol{s}_j = \boldsymbol{s}_j^* \sqrt{\frac{10^{\frac{-E_s/E_j(\text{dB})}{10}}}{2 \times 8}}, \tag{6.10}$$

by leveraging the fact that the samples per symbol are 8 throughout transmission and $\mathbb{E}[E_s]$ is 1 for all signals. The final combined signal, $s_{tx+j}$ is then simply the addition of the original signal and perturbation created by the ARN, $s_{tx} + s_j$.

#### 6.2.2.4 Receiver

The receiver is implemented by first match filtering the signal and then downsampling to one sample per symbol. Two different receiver implementation are used. The first does not simulate a channel and therefore extracts the true transmitted symbols $S_{tx+j}$. The second

simulates an AWGN channel and extracts $S_{rx}$, where $S_{rx}$ is $S_{tx+j} + \mathcal{CN}(0, \sigma_{rx}^2)$ ($\mathcal{CN}$ denotes a complex normal distribution). This noisy estimate of the received symbol is then used to compute $I_s$ by determining if the nearest symbol in the constellation to $S_{rx}$ matches $S_{tx}$. The noise power, $\sigma_{rx}^2$, is set to achieve a desired SNR. In the current work, the receiver SNR is uniformly sampled from 10 to 20 dB.

#### 6.2.2.5 Eavesdropper Channel and Signal Isolation Model

The eavesdropper receives the signal through an AWGN channel with the noise power, $\sigma_{adv}^2$, set to achieve a desired SNR (note that $\sigma_{adv}^2$ and $\sigma_{rx}^2$ are independent as the eavesdropper and receiver are not necessarily co-located). In the current work, this SNR is uniformly sampled from 10 to 20 dB. Although there are multiple effects due to errors in the signal detection and isolation stage [48], the one that is modeled in this work is a simple time offset. This effect occurs because a RFML system will typically only look at a small window in time and it is unknown when that window starts. In Chapter 5.4, it was found that time offsets (from the window used to craft the adversarial perturbation) higher than the symbol period did not provide a significant detriment to adversarial success for FGSM; therefore, this effect is modeled, during training, as a discrete uniform distribution from 0 to 8 samples. Non-overlapping and consecutive windows are used to create examples for the eavesdropper.

#### 6.2.2.6 Training Procedure

As mentioned in Section 6.2.2.1, the input data to the ARN consists of random signal streams. A batch, in the current work, refers to one iteration of training where 1000 random symbols are sent. During a batch, all parameters of the environment such as SNRs and time offsets at the eavesdropper are static; however, they are re-sampled at the beginning of each new batch. Each batch consists of a single back propagation and ARN parameter update. An epoch, in the current work, refers to 1000 batches. Because the input data, as well as the noise realizations, are modeled as random variables, instead of a fixed dataset, a separate validation procedure is not used to reduce computational complexity. Instead, the mean training loss per epoch is used for early stopping with a patience of 5. The maximum number of training epochs is set as 10.

### 6.2.3 Adversarial Residual Network Architecture

The current work considers an ARN architectures that does not change the sample rate of the signal and therefore produces 1 complex sample on the output for every 1 complex sample on the input. In the context of neural networks, this can be easily implemented as a fully convolutional neural network. This network takes in one channel on the input and in the current work, the batch dimension is not used for the ARN (though it is used for AMC). Therefore, the input to the network is a tensor of the form $[1, 1, 2, N]$, where $N$ represents a variable number of samples for the input signal. The output of the ARN is thus the same dimensions since there is no change in sample rate. There are 256 kernels with $1 \times 7$ filters

in the first layer, 160 kernels with $2 \times 7$ filters in the second layer, and 2 kernels with $1 \times 129$ filters in the final layer which are then concatenated to form $I$ and $Q$ on the output of the ARN. Batch Normalization [84] is used for regularization and ReLU is used as the activation function on all but the final layer.

### 6.2.4 Automatic Modulation Classification

The reference AMC implementation used is the same as in Chapter 4 and 5. The current Chapter uses Batch Normalization [84] instead of dropout as the regularization method. The AMC network was trained on Dataset A which contains 11 modulations with all signals eight times over sampled.

### 6.2.5 Evaluation Procedure

For each attack, a signal consisting of 8000 random symbols is created and passed through the ARN to craft perturbations. In order to evaluate the impact to communications, the combination of signal and perturbation is subjected to AWGN at SNRs varying from 0 to 20 dB ($E_s/N_0$). At each SNR, 100 trials are performed and a BER is calculated. To evaluate the impact to AMC, the combination of signal and perturbation is again subjected to AWGN in the same SNR range. At each SNR, sample time offsets are introduced varying from 0 to 8 (the symbol period). 10 trials are performed for each characterization and classification accuracy is computed.

For the evaluation, the initial signal is created, and AWGN is applied to the signal and

perturbation, using GNU Radio. The sample time offsets are implemented in Python and all machine learning components are implemented using PyTorch.

## 6.3 Results

In order to validate the methodology described in prior sections, 60 ARNs were trained to evade AMC. Four source modulations were used: BPSK, QPSK, 8PSK, and QAM16. Multiple configurations were tried, with $\alpha$ values of 50%, 70%, and 90%, and $E_s/E_j$ limits were swept from $-10$ to 10 dB with a step size of 5 dB.

### 6.3.1 Trading Off Communication for Deception

Logically, the ability to evade signal classification will require sacrificing some communications ability. This tradeoff can be directly observed in Figure 6.3 which shows the correlation between adversarial loss, which measures the ability to deceive, and communications loss, which measures the ability to communicate. The loss values are extracted from the mean training loss of the "best" epoch, where "best" is described as the lowest total training loss and is therefore the epoch used for early stopping. As expected, the groupings closely follow the $\alpha$ parameter used with some outliers for an $E_s/E_j$ limit of 10 dB. When the ARN has a small power budget, it is not able to perturb the signal and therefore incurs a lower communications loss but loses the ability to evade classification.

This tradeoff can also seen in the convergence plots, which represent the training loss

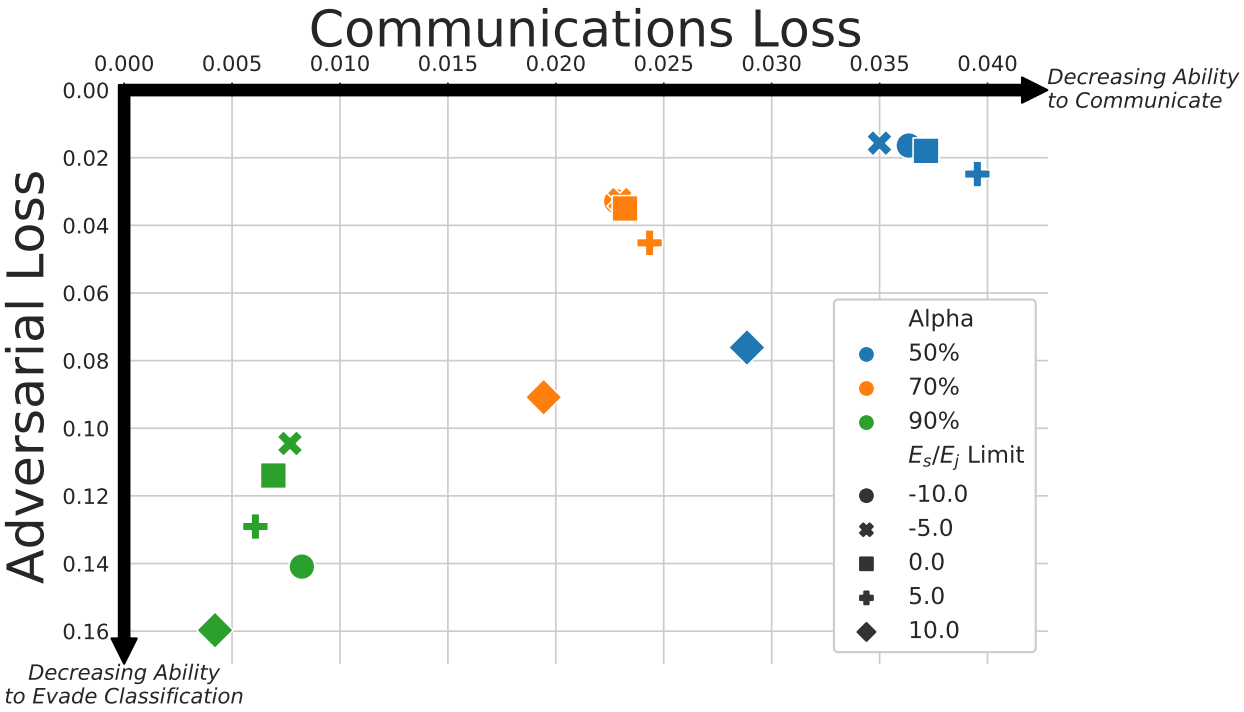


**Figure 6.3:** Mean communications and adversarial loss values of the best epoch for multiple trained ARNs with a source modulation of 8PSK. The optimal location would be the upper left of the plot.

over time. In Figure 6.4, the mean training loss for each of the loss functions is provided (before being scaled using $\alpha$ and $\beta$) for each batch during training. Note that each epoch, in the current work, consists of 1000 batches. During the second epoch, it can be seen that the ARN learns to sacrifice communications for evasion because the communications loss rises while the adversarial loss lowers.

### 6.3.2 Evading Modulation Recognition

In order to evaluate the ARN's ability to evade modulation recognition results are presented in Figure 6.5 for BPSK, Figure 6.6 for QPSK, Figure 6.7 for 8PSK, and Figure 6.8 for QAM 16. These results are presented as an average accuracy across uniformly sampled time offsets from 0 to 8 (the symbol period) to show that the ARN does not depend on time synchronization with the eavesdropper. As with FGSM attacks (Chapter 5.2), the ARN is most effective when the eavesdropper perceives the signal with a high SNR, but, is able to greatly lower the classification accuracy across the board (discussion of the impact to communications is reserved for Section 6.3.3).

Recall that the perturbation power was hard limited by (6.10) such that $E_s/E_j$ is guaranteed to be greater than a specified limit, but, in practice, the average perturbation power was much lower than the limit, and therefore $\mathbb{E}[E_s/E_j] \gg L$ (this is further discussed in section 6.3.4). Although the exact location of the highest classification accuracy cannot be directly predicted from $\mathbb{E}[E_s/E_j]$, as was found for FGSM attacks (Chapter 5.2), the relative locations are preserved. For instance, in Figure 6.5, $\mathbb{E}[E_s/E_j]$ for the $-10$ dB attack (blue)

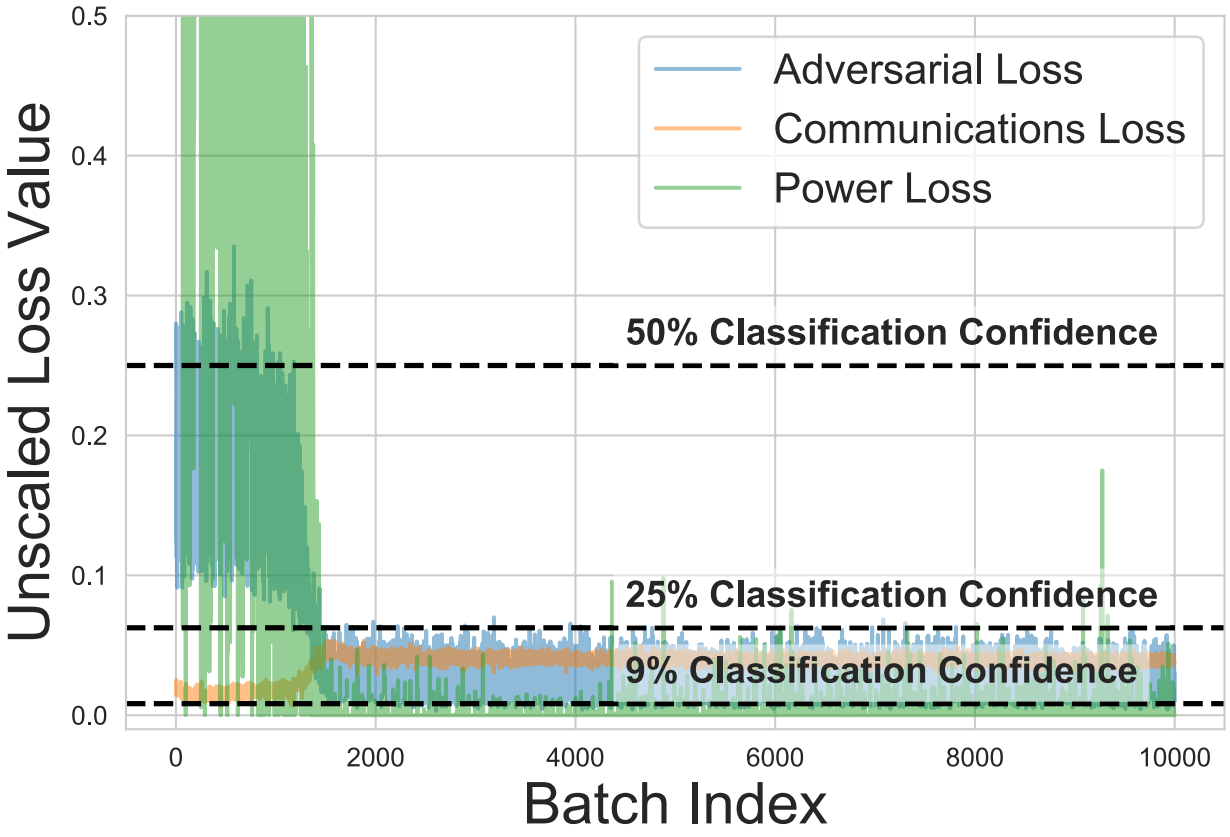


**Figure 6.4:** Mean training loss per batch, before scaling with $\alpha$ and $\beta$, for an ARN being trained on a source modulation of 8PSK, $\alpha$ parameter of 50%, and an $E_s/E_j$ limit of 5 dB. At the beginning of training, the ARN trades off communication's ability (orange) for deceiving the eavesdropper (blue). For reference, horizontal dashed lines are added to represent the mean classification confidence in the true class that would produce a specific adversarial loss (blue).

is 9 dB while $\mathbb{E}[E_s/E_j]$ for the other ARN attacks (orange and green) in Figure 6.5 are 10 dB. Further, observe that the highest classification accuracy occurs at $\approx 7$ dB and $\approx 8$ dB for blue and the others, respectively, reinforcing that higher perturbation powers can aid in generalizing adversarial evasion attacks across larger SNR ranges for the ARN, just as they did for FGSM in Chapter 5.2. The same result can be observed in Figure 6.7 where, counter-intuitively, $\mathbb{E}[E_s/E_j]$ is 1 dB higher for the $-5$ dB limited attack (orange) than the 0 dB attack (blue).

### 6.3.3 Maintaining the Communications Link

Figure 6.5 shows that the ARN can achieve similar results, in terms of BER increase required to achieve a lower classification accuracy as FGSM. The ARN was able to do this without requiring the computation of a gradient to craft the adversarial perturbation, which makes it a more suitable methodology for a real time communication system due to the lower computational complexity. Further, as was shown in Chapter 5.2, the BER penalty for high intensity FGSM attacks can quickly become prohibitive for higher order modulations such as 8PSK and QAM16. Figure 6.7 shows that incorporating communications objectives into the loss function of the ARN allows for lower BER increases to achieve this evasion which makes the ARN much more effective when higher order source modulations are used than a traditional adversarial evasion attack which only considers classifier evasion as an objective.

Despite incorporating a communications loss, the ARN did not completely eliminate BER in all modulations tested. With an $\alpha$ of 50%, four of five QAM16 ARNs contained BERs

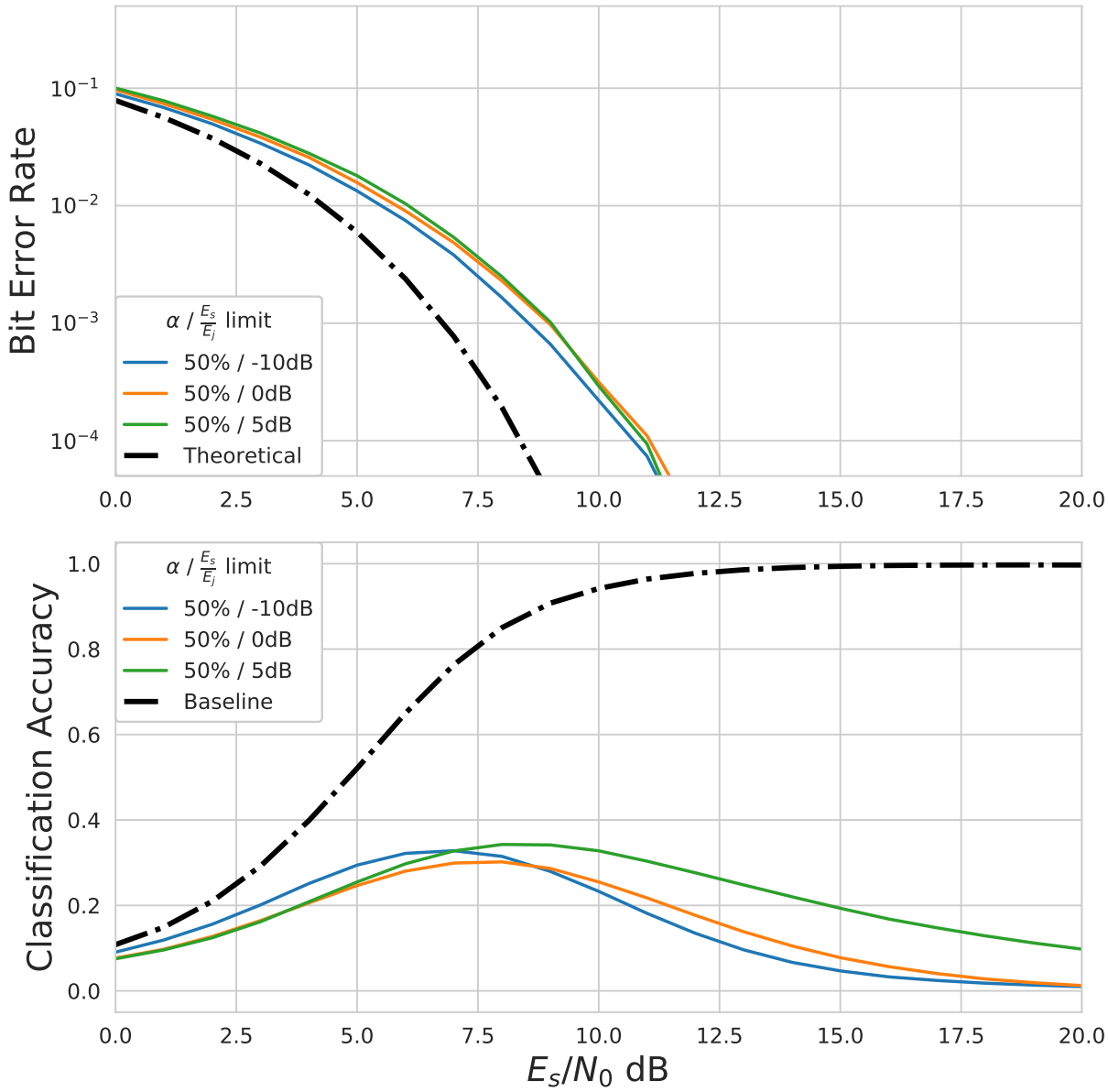


**Figure 6.5:** (top) BER for adversarial attacks at differing $E_s/E_j$ for a source modulation of BPSK as well as the theoretical BER for comparison. (bottom) Classification accuracy for the same adversarial attacks with the baseline classification accuracy, which would occur when no adversarial machine learning is performed, shown. The classification accuracy is presented with uniformly distributed time offsets.

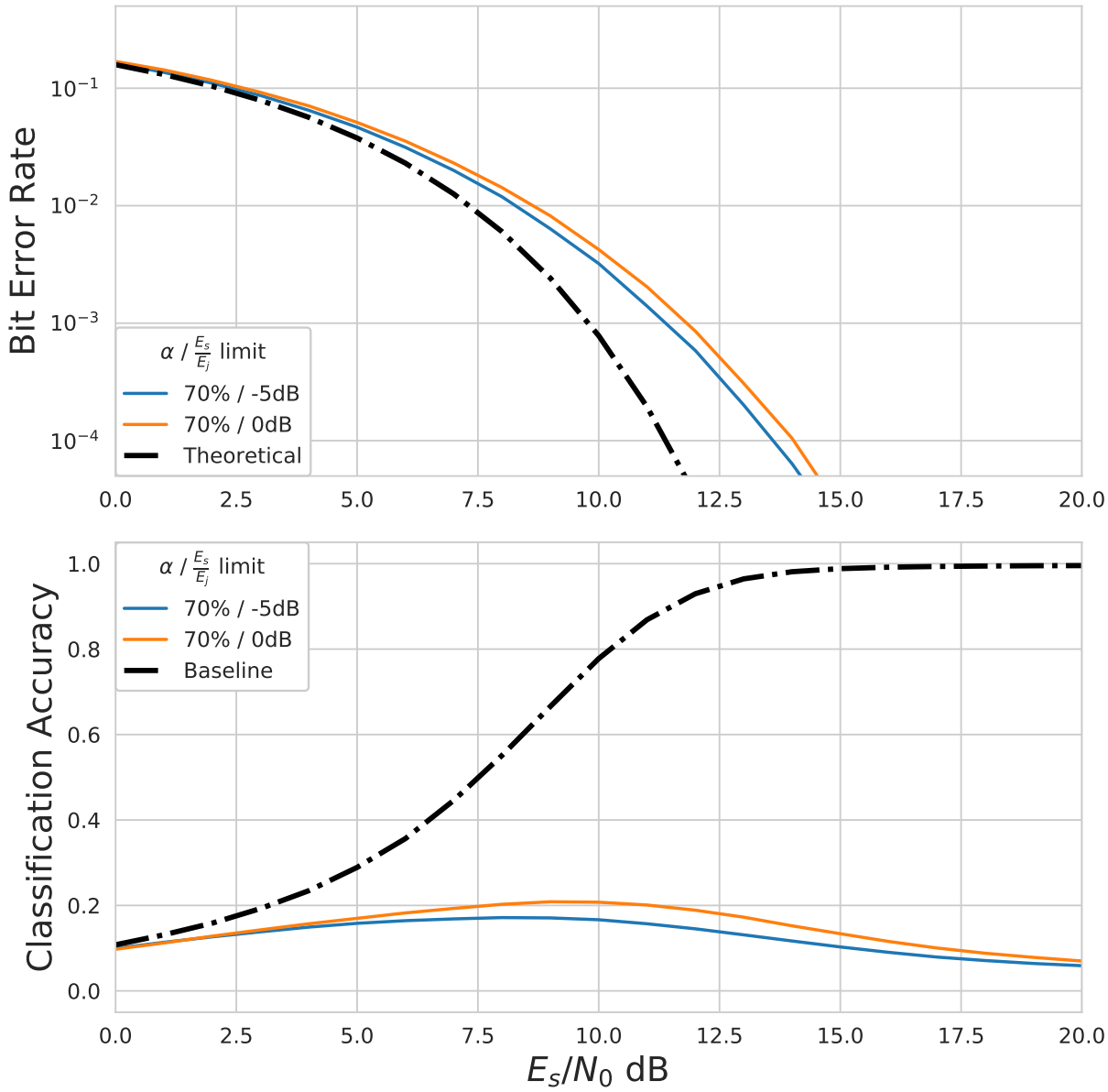


**Figure 6.6:** (top) BER for adversarial attacks at differing $E_s/E_j$ for a source modulation of QPSK as well as the theoretical BER for comparison. (bottom) Classification accuracy for the same adversarial attacks with the baseline classification accuracy, which would occur when no adversarial machine learning is performed, shown. The classification accuracy is presented with uniformly distributed time offsets.

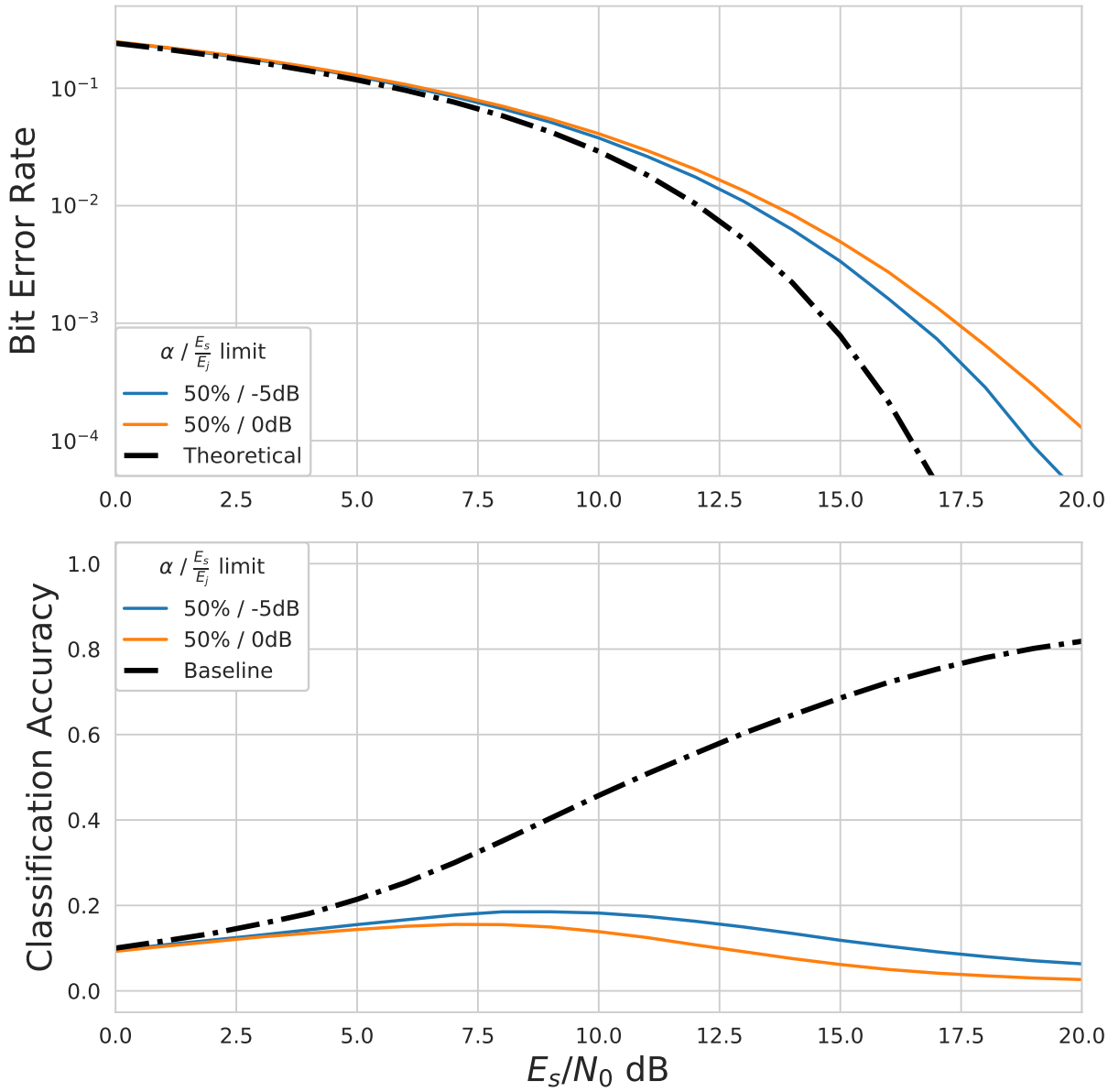


**Figure 6.7:** (top) BER for adversarial attacks at differing $E_s/E_j$ for a source modulation of 8PSK as well as the theoretical BER for comparison. (bottom) Classification accuracy for the same adversarial attacks with the baseline classification accuracy, which would occur when no adversarial machine learning is performed, shown. The classification accuracy is presented with uniformly distributed time offsets.

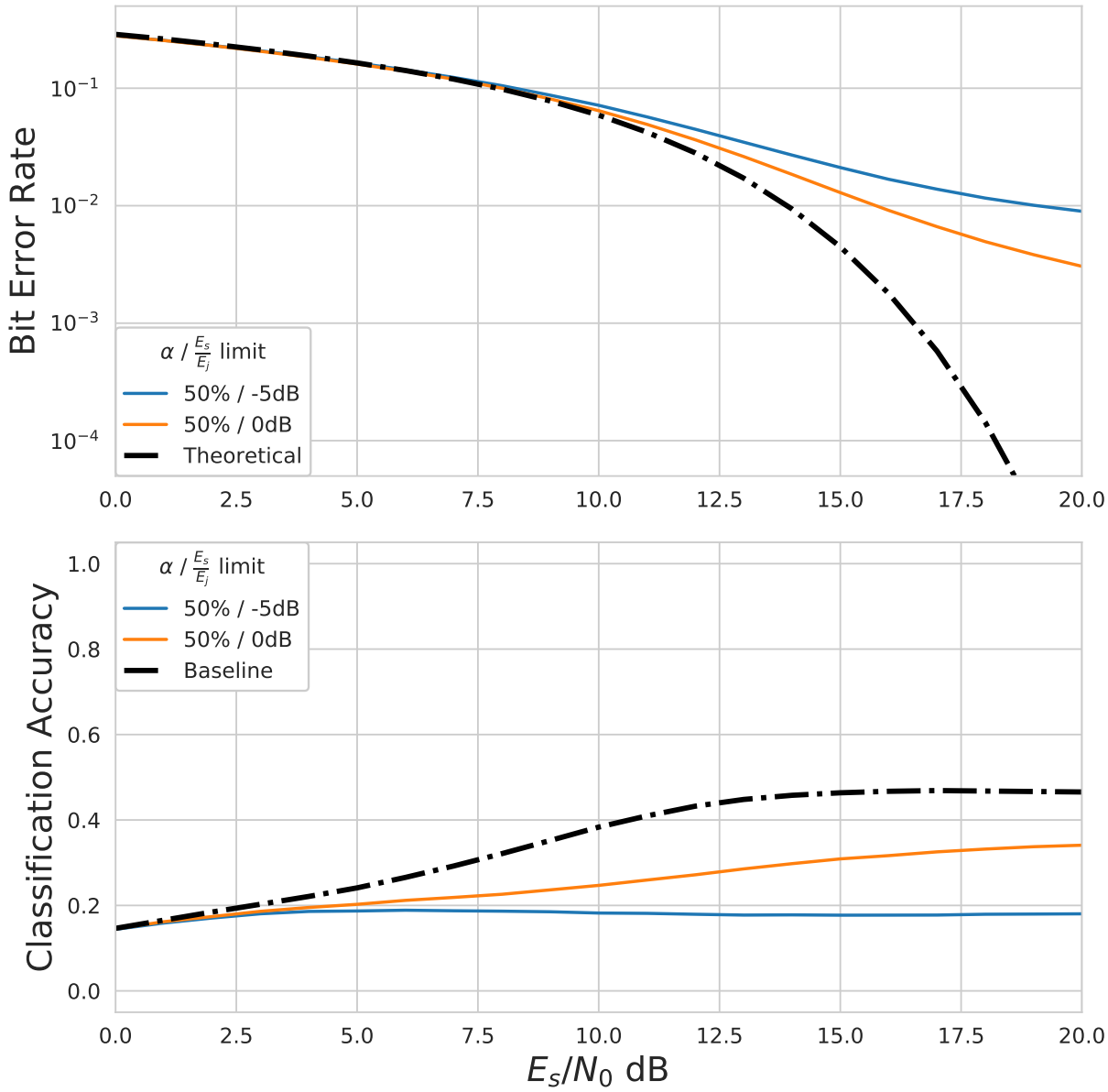


**Figure 6.8:** (top) BER for adversarial attacks at differing $E_s/E_j$ for a source modulation of QAM16 as well as the theoretical BER for comparison. (bottom) Classification accuracy for the same adversarial attacks with the baseline classification accuracy, which would occur when no adversarial machine learning is performed, shown. The classification accuracy is presented with uniformly distributed time offsets.

even without noise. However, while Chapter 5.2 found that high intensity attacks could cause BERs of up to 30% without noise, rendering the communication largely useless, the ARN developed in the current work only had BERs less than $5 \times 10^{-3}$ for QAM16. Therefore, while the BERs have not been completely eliminated for higher order modulations by the current work, they have been greatly reduced.

Further, counter to the FGSM attack presented in Chapter 5.2, increasing the perturbation power output by the ARN does not necessarily increase the BER. As can be seen in Figure 6.5, the ARN with a $-10$ dB $E_s/E_j$ limit had a lower BER than the other presented attacks which had lower perturbation powers. This can be attributed to two things. First, as the ARN is aware of the underlying communications methodology, the perturbations crafted are not simply interference to the intended receiver and can be used to positively affect the interpretation of the transmitted signal. Second, as the ARN operates on an oversampled signal, due to the precedent set for AMC by the RML2016.10A dataset [80], the effective $E_s/E_j$ when the receiver is making the hard decision on which symbol was sent is in fact higher because portions of the perturbation do not make it through the match filter.

### 6.3.4 Adversarial Spectrum Usage

One example of the spectrum usage by the ARN is provided in Figure 6.9. The ARN increases the bandwidth of the signal by placing some of the perturbation energy out of band. In the current results, a match filter provided up to a 6 dB gain in $E_s/E_j$; specifically, the signal shown in Figure 6.9 can be filtered to provide a 2 dB gain. While this is a positive

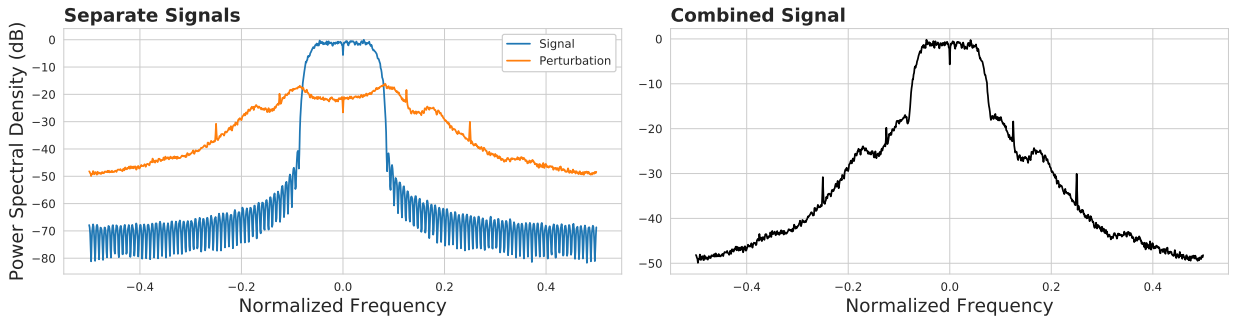


**Figure 6.9:** Normalized power spectral density for a 8PSK source modulation with a perturbation crafted by an ARN with an $\alpha$ of 0.5 and $E_s/E_j$ limit of 0 dB. (left) Frequency content of each signal separately. (right) Frequency content of the combination of both signals, which represents what would be transmitted.

from the communications perspective, it may be detrimental from an evasion perspective. While the current work considered an AMC model which was operating on an eight times over sampled signal, a more ideal signal detection and isolation stage could also filter the out of band perturbation which would likely increase classification accuracy. Further, the current work has considered the signal detection and isolation stage to be fixed and not react to the attack; however, this is unrealistic as an eavesdropper would not know *a priori* the parameters of the signal and therefore the perturbation would have cascading effects on the signal detection and isolation stage. However, further exploration of the compounding impact on the signal detection and isolation stage is left as future work.

## 6.4 Conclusion

The current chapter has developed and tested methodology for evading signal classification with minimal impact to the underlying communication. The developed methodology was shown to be equivalent or better than an FGSM attack when evaluated in terms of both BER and classification accuracy. Further, an ARN, once trained, has encapsulated the knowledge needed to craft the perturbation into the parameter matrices of a fully convolutional neural network. While prior work in adversarial RFML has required the solving of an optimization problem via gradient descent using one (Chapter 5 and [43]) or more [4], [45] iterations for each block of signals that are to be transmitted, the ARN only requires a single inference by a pre-trained model. Thus, an ARN has reduced computational complexity during operation and is more suitable to high data rate systems.

Most of the work in adversarial RFML [4], [45], including much of the current work, operates on the RML2016.10A (Dataset A) dataset [80] and thus considers perturbations that can use eight times the bandwidth of the underlying signal. The current work has shown that while untargeted adversarial evasion can be achieved with low power perturbations, there is still a large power difference between in band and out of band. Ergo, a more optimal signal detection and isolation stage would likely filter out some of the adversarial perturbation before subsequent classification by DNNs. Future research should consider joint attacks on the signal detection and isolation stage as well as the AMC stage in order to ensure that the cascading effects of the perturbation on other subsystems is modeled. Further, all future

adversarial RFML attacks must report the bandwidth requirements of the perturbation as a hard constraint just as power constraints are reported in order for comparison between attacks.

While the ARN presented in the current chapter greatly reduced the impact to the underlying communications, ARNs trained on QAM16 for high intensity attacks still contained bit errors even without noise. Thus, future work should explore forward error correction techniques, such as block coding, that will correct for the small BER present in QAM16. If the ARN also operates on a block, it will implicitly be able to learn to sacrifice certain symbols, which would be corrected by the error correction technique, to evade signal classification.

# Chapter 7

# Conclusion

The current work has investigated the vulnerabilities of current RFML systems, specifically raw IQ based AMC, to adversarial evasion attacks. It has done this by first developing a threat model that enhances the vocabulary used to describe the adversarial goals that are unique to RFML (Chapter 3). The current work then evaluated the closest analogy to most of the literature surrounding adversarial machine learning in CV [35, 37, 39], a direct access attack using an existing, well-known, adversarial machine method, FGSM, from CV (Chapter 4). Chapter 4 found that RFML signal classification DNNs are just as susceptible to adversarial examples as in CV by demonstrating that an FGSM attack offered a 10 dB improvement, in perturbation power needed to achieve a given accuracy reduction, over simply adding Gaussian noise to the signal. However, a direct access attack assumes that an additional exploit has been used to compromise the signal processing chain of the eavesdropper, and therefore, it is unrealistic in many scenarios. Therefore, the current work then investigated the vulnerabilities of RFML systems to self protect attacks, which are launched OTA, where the perturbation is subject to multiple channel and signal detection effects and has an impact on the intended receiver of the signal (Chapter 5). Chapter 5 found that channel and signal detection effects can significantly (by up to a 20% increase in accuracy) reduce the effectiveness of adversarial machine learning in OTA attacks. While these effects can be overcome by higher intensity attacks, Chapter 5 showed that high intensity attacks, which significantly corrupt the underlying transmission, quickly become prohibitive for a rational adversary whose primary objective is to communicate information to the intended receiver and evading signal classification is a secondary goal. In order to envision future threats, the

current work developed a "communications aware" attack that directly incorporated BER in the adversarial methodology, did not depend on time synchronization with the eavesdropper, and did not require solving an optimization problem for every communications block that was transmitted (Chapter 6). Chapter 6 found that the developed attack performed equivalently or better than an FGSM attack when evaluated in terms of BER and classification accuracy. The current work now concludes with this chapter.

This chapter first summarizes the current vulnerabilities of RFML systems that were discovered through the current work, outlines what it will take to secure against these threats, and then summarizes the proposed future work beyond this Thesis.

## 7.1 Current RFML Vulnerabilities

Similar to DNNs in other modalities, RFML DNNs are severely vulnerable to direct access attacks. In practicality, these vulnerabilities are partially mitigated in OTA attacks through channel and signal detection effects, which would traditionally be considered an impediment to signal classification, but, also impede the adversarial machine learning techniques used. The perturbation applied to the signal must overcome the noise power for maximum effectiveness, therefore, lower SNR captures of an adversarial signal can result in higher classification accuracy. Further, FGSM has reduced effectiveness when the assumption of time synchronization is removed (up to 20% higher accuracy) and a more realistic model of the signal detection and isolation stage [48] is used that induces center frequency offsets (up to

10% higher accuracy). All of these effects can be overcome by increasing the adversarial perturbation energy but this quickly becomes prohibitive for a rational adversary whose primary goal is to transmit information to an intended receiver, using a known modulation, and secondary goal is to avoid recognition of that modulation by an unintended eavesdropper. The underlying interpretation of the signal is central to successful evasion because, without that constraint, these attacks quickly devolve into simply transmitting a known modulation with a random data stream or replaying a signal that is observed but the exact waveform is unknown and therefore cannot be replicated. Therefore, while high intensity attacks may still be beneficial when using a low order modulation, such as BPSK, because the BER, even without noise, can approach 30% for higher order modulations such as QAM16, there is limited benefit to a rational adversary in applying a direct translation of adversarial machine learning from CV.

Further, while the current work focused on untargeted attacks, the work in [4] extended the methodology from Chapter 4 into a targeted attack and found that significant perturbation energy was required, even for a direct access attack, to reliably move between differing signal categories such as digital and analog waveforms. Although it was not evaluated in [4], these high energy perturbations will undoubtedly corrupt the underlying transmission, just as the work in Chapter 5, because they are also not considering the underlying communication during the crafting of the perturbation. Therefore, while current adversarial machine learning techniques may be sufficient to masquerade a QPSK signal as an 8PSK or QAM16 signal, they would not be sufficient to masquerade a QPSK signal as an analog transmission

because of the large impact on the communication to an intended receiver. Thinking more broadly, while the current abilities of these adversarial machine learning techniques could be used to evade automatic demodulation (by simply being mistaken as any other modulation than the true class), they are likely not sufficient for Primary User Emulation [23], where a digital signal would need to masquerade as a radar transmission or analog broadcast.

In order to envision future threats, beyond a direct translation of methodology from CV, Chapter 6 showed that the underlying impact to the transmission can be greatly reduced by incorporating BER in the adversarial methodology. It did this, while encapsulating the underlying perturbation creation procedure into a fully convolutional neural network and therefore could easily be deployed to a real time communications system due to the lower computational complexity. This improved methodology both lowered BER and did not depend on time synchronization and thus performed equivalently (for low order modulations) or better (for high order modulations) than the FGSM attack. While BER was greatly reduced by the ARN presented in Chapter 6, small amounts of bit errors were still induced in the QAM16 source modulations tested when under high intensity attacks. While the FGSM attack had BERs up to 30%, the ARN was able to reduce this to $5 \times 10^{-3}$. Regardless, this means that an adversary would still require forward error correction techniques, such as block coding, to account for these errors but the rate could be greatly increased over an FGSM based attack.

In summary, adversarial RFML is a credible and evolving threat to RFML signal classification systems and therefore defenses must begin to be investigated; however, a direct

translation of adversarial machine learning from CV is generally insufficient to cause serious immediate concern due to its high impact on the underlying communication, assumption of time synchronization, and inability to generalize over channel and signal detection effects.

## 7.2 Hardening RFML

A RFML system will never be completely invulnerable to attack, but, the benefits of launching that attack can be greatly reduced. Even minimal defenses can increase the perturbation power, bandwidth used, computational complexity, and bit error rate of an adversarial transmitter. However, this must be done without significant increases to the RFML system's classification latency, computational complexity, and threat surface.

Hardening RFML is beginning to be investigated in [47] and [85] where two threads are investigated: detecting that an attack occurs and being robust to that attack. Detecting the presence of an adversarial evasion attack is likely sufficient for many civilian applications of RFML, such as DSA, where, sensing that a transmitter is masquerading as a primary user is sufficient, even if the specific signal format they're using is unknown. However, when military applications are considered, detection of the attack becomes of limited benefit. Considering the case of automatic demodulation, where an eavesdropper seeks to passively collect the data of transmitters in its proximity, detecting that they are camouflaging their transmission does not allow for successful demodulation because the underlying modulation scheme being used is still not known. Therefore, while detection of the attack is a good first step for

defenses, being robust to the attack must also be investigated.

## 7.3 Limitations of the Current Work and Suggested Future Directions

All work in adversarial RFML so far, including the current work, has ignored the cascading impact that these perturbations will have on both the intended receiver and the signal processing of the eavesdropper. While the current work evaluates BER, synchronization between the receiver and transmitter is assumed. While this is a reasonable starting assumption, due to oscillator drifts, a phase-locked loop would typically be used to maintain synchronization and adversarial perturbations would undoubtedly have an effect on this which could impact the interpretation of all signal types. Further, automatic gain control would typically be used to adjust for varying received signal strength and adversarial perturbations would affect this as well. Neither of these cascading impacts on the receiver have been modeled in the current work and should be investigated in the future.

The current work modeled errors in the signal detection and isolation stage but assumed that it was static and thus did not react to the attacks. While this assumption is necessary for preliminary work in adversarial RFML, it is not a valid assumption that could be made on a real system. The current work, as well as many others, operate on oversampled signals when creating the perturbation and thus, as shown in Chapter 6, increase the bandwidth of the signal. While increasing the bandwidth of the signal is a valid design decision for adversarial

RFML, assuming that it will not be filtered out in the preliminary signal processing before signal classification is not. Therefore, future work should consider a joint attack on both the signal detection and isolation stage, as well as the signal classification model, in order to ensure that these adversarial evasion attacks generalize across more optimal initial signal processing.

Considering the previous cascading impact, more perturbation energy will likely be concentrated into the same frequency range as the underlying signal, removing the benefit the intended receiver had in the current work of being able to filter out parts of the perturbation. This, combined with the already small amount of bit errors present in higher order modulations, means that forward error correction must be considered in future work. Future work should consider applying the ARN concept from Chapter 6 to communications blocks that are protected with a block code in order to lower the BER.

The current work has considered multiple channel and signal detection effects; however, it only performed a simulated analysis and therefore the effects are not exhaustive of all that would be seen in a real world scenario. Future work, particularly those seeking to establish the vulnerabilities of a specific RFML device, should consider empirical analysis of these attacks by transmitting and receiving these adversarial signals using SDRs such as USRPs. By physically transmitting the signal, additional effects will be included in the evaluation such as non-linearities in power amplifiers, quantization error due to ADCs and DACs, and interference from other transmitters in the environment. However, each of these hardware impairments and interferers are unique to the specific RFFEs used and environments oper-

ated in. In general, these impairments and noise cannot be directly controlled for. Therefore, it becomes more difficult to isolate the impact of a specific source of noise, such as the time or center frequency offsets discussed in the current work, on adversarial success (and to overcome that specific impairment as was done for sample time offsets in Chapter 6). Further, parameters of the noise, such as SNR, have to be estimated when performing an experimental validation; thus, reporting adversarial success as a function of any of these parameters loses accuracy even when they can be more directly controlled. For these reasons, while an empirical analysis could provide a good estimate of the vulnerabilities of a specific RFML system, the current work has focused on a simulation based analysis in order to: stay more general as an analysis of the algorithms used and not the specific hardware used, isolate the impact of each individual source of noise, and provide estimations of adversarial success across a range of parameters.

Finally, the current work has only considered attacks against RFML signal classification, but, that is not indicative of all of RFML. Investigating the vulnerabilities of the auto-encoder based waveforms [6,12] has begun and have been found to be especially vulnerable to adversarial machine learning as well [46]. Future work in adversarial RFML should expand further into attacks against Deep Q Network (DQN) agents that control the RFFE [7]. Evaluating the vulnerabilities of all applications of RFML will help to define the limitations of this technology in adversarial environments and serves as the first step to developing techniques to make the technology more robust.

# Bibliography

[5] Cisco Systems, "Cisco visual networking index: Forecast and trends, 2017-2022 white paper," Tech. Rep. 1551296909190103, Cisco Systems, Februrary 2019.

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

[7] T. J. O'Shea and T. C. Clancy, "Deep reinforcement learning radio control and signal detection with kerlym, a gym rl agent," *CoRR*, vol. abs/1605.09221, 2016.

[8] T. J. OShea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Int. conf. on engineering app. of neural networks*, pp. 213–226, Springer, 2016.

[9] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *IEEE Int. Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–6, IEEE, 2017.

[10] K. Karra, S. Kuzdeba, and J. Petersen, "Modulation recognition using hierarchical deep neural networks," in *IEEE Int. Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–3, 2017.

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

[12] T. OShea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Commun. and Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[13] L. J. Wong, W. C. Headley, S. Andrews, R. M. Gerdes, and A. J. Michaels, "Clustering learned cnn features from raw i/q data for emitter identification," in *IEEE Military Commun. Conf. (MILCOM)*, 2018.

[14] J. Mitola, "Cognitive radio for flexible mobile multimedia communications," in *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99) (Cat. No.99EX384)*, pp. 3–10, 1999.

[15] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[16] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Commun.*, vol. 1, no. 2, pp. 137–156, 2007.

[17] W. C. Headley, J. D. Reed, and C. R. C. M. d. Silva, "Distributed cyclic spectrum feature-based modulation classification," in *IEEE Wireless Commun. and Netw. Conf.*, pp. 1200–1204, 2008.

[18] D. T. Kawamoto and R. W. McGwier, "Rigorous moment-based automatic modulation classification," *Proc. of the GNU Radio Conf.*, vol. 1, no. 1, 2016.

[19] A. Hazza, M. Shoaib, S. A. Alshebeili, and A. Fahad, "An overview of feature-based methods for digital modulation classification," in *1st Int. Conf. on Commun., Signal Proc., and their App.*, pp. 1–6, 2013.

[20] M. M. T. Abdelreheem and M. O. Helmi, "Digital modulation classification through time and frequency domain features using neural networks," in *2012 IX Int. Sym. on Telecommun.*, pp. 1–5, 2012.

[21] M. Bari, A. Khawar, M. Doroslovaki, and T. C. Clancy, "Recognizing fm, bpsk and 16-qam using supervised and unsupervised learning techniques," in *49th Asilomar Conf. on Signals, Systems and Computers*, pp. 160–163, 2015.

[22] P. Tilghman, "Radio frequency machine learning systems (rfmls)." https://www.darpa.mil/program/radio-frequency-machine-learning-systems. Accessed: 2019-04-15.

[23] R. Chen, J.-M. Park, and J. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Commun.*, vol. 26, no. 1, pp. 25–37, 2008.

[24] K. I. Talbot, P. R. Duley, and M. H. Hyatt, "Specific emitter identification and verification," *Technology Review*, vol. 113, 2003.

[25] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[26] H. Kwon, Y. Kim, K.-W. Park, H. Yoon, and D. Choi, "Friend-safe evasion attack: An adversarial example that is correctly recognized by a friendly classifier," *Computers & Security*, vol. 78, pp. 380–397, 2018.

[27] J. Hayes and G. Danezis, "Learning universal adversarial perturbations with generative models," *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 43–49, 2018.

[28] I. Goodfellow, P. McDaniel, and N. Papernot, "Making machine learning robust against adversarial inputs," *Commun. ACM*, vol. 61, no. 7, pp. 56–66, 2018.

[29] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[30] S. Baluja and I. Fischer, "Learning to attack: Adversarial transformation networks," *Proc. of AAAI-2018. http://www. esprockets. com/papers/aaai2018. pdf*, 2018.

[31] W. Wang and Q. Zhu, "On the detection of adversarial attacks against deep neural networks," in *Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense*, SafeConfig '17, (New York, NY, USA), pp. 27–30, ACM, 2017.

[32] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. of the 2017 ACM on*

*Asia Conf. on Computer and Commun. Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pp. 506–519, 2017.

[33] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 86–94, 2017.

[34] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *CoRR*, vol. abs/1611.01236, 2016.

[35] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[36] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2016.

[37] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, IEEE, 2016.

[38] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recog.*, pp. 427–436, 2015.

[39] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Int. Conf. on Learning Representations*, 2015.

[40] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[41] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. D. McDaniel, "Ensemble adversarial training: Attacks and defenses," *CoRR*, vol. abs/1705.07204, 2017.

[42] J. Wang, J. Sun, P. Zhang, and X. Wang, "Detecting adversarial samples for deep neural networks through mutation testing," *CoRR*, vol. abs/1805.05010, 2018.

[43] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Commun. Letters*, pp. 1–1, 2018.

[44] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, "Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies," in *IEEE Int. Conf., on Commun. Workshops (ICC Workshops)*, pp. 1–6, 2018.

[45] M. Z. Hameed, A. Gyorgy, and D. Gunduz, "Communication without interception: Defense against deep-learning-based modulation detection," *arXiv preprint arXiv:1902.10674*, 2019.

[46] M. Sadeghi and E. G. Larsson, "Physical adversarial attacks against end-to-end autoencoder communication systems," *CoRR*, vol. abs/1902.08391, 2019.

[47] S. Kokalj-Filipovic and R. Miller, "Adversarial examples in RF deep learning: Detection of the attack and its physical robustness," *CoRR*, vol. abs/1902.06044, 2019.

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

[49] J. H. Reed, J. T. Bernhard, and J. Park, "Spectrum access technologies: The past, the present, and the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1676–1684, 2012.

[50] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, pp. 26–38, May 1995.

[51] J. Mitola, "Software radio architecture: a mathematical perspective," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 514–538, Apr. 1999.

[52] "GNU Radio." https://www.gnuradio.org, 2019.

[53] "LiquidDSP." http://liquidsdr.org, 2019.

[54] "Redhawk." https://redhawksdr.github.io/, 2019.

[55] E. Research, "Usrp software defined radio (sdr)." http://www.ettus.com/products/, 2019.

[56] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[57] S. M. Dudley, W. C. Headley, M. Lichtman, E. Y. Imana, X. Ma, M. Abdelbar, A. Padaki, A. Ullah, M. M. Sohul, T. Yang, and J. H. Reed, "Practical issues for spectrum management with cognitive radios," *Proc. of the IEEE*, vol. 102, no. 3, pp. 242–264, 2014.

[58] R. Girshick, "Fast r-cnn," in *International Conference on Computer Vision (ICCV)*, 2015.

[59] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[60] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[61] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017.

[62] T. J. O'Shea, S. D. Hitefield, and J. Corgan, "End-to-end radio traffic sequence recognition with recurrent neural networks," *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 277–281, 2016.

[63] J. H. Reed, *Software radio: a modern approach to radio engineering*. Prentice Hall Professional, 2002.

[64] J. Erickson, *Hacking: The Art of Exploitation, 2nd Edition*. San Francisco, CA, USA: No Starch Press, second ed., 2008.

[65] S. Hitefield, V. Nguyen, C. Carlson, T. O'Shea, and T. Clancy, "Demonstrated llc-layer attack and defense strategies for wireless communication systems," in *2014 IEEE Conference on Communications and Network Security*, pp. 60–66, Oct 2014.

[66] S. Hitefield, M. Fowler, and T. C. Clancy, "Exploiting buffer overflow vulnerabilities in software defined radios," in *2018 IEEE International Conference on Computer and Information Technology (CIT)*, Aug 2018.

[67] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. of the 4th ACM Workshop on Security and Artificial Intelligence*, AISec '11, (New York, NY, USA), pp. 43–58, ACM, 2011.

[68] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

[69] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, (Berkeley, CA, USA), pp. 601–618, USENIX Association, 2016.

[70] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *IEEE Symposium on Security and Privacy (SP)*, pp. 19–35, 2018.

[71] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proceedings of the 29th International Conference on International Conference on Machine Learning*, ICML'12, (USA), pp. 1467–1474, Omnipress, 2012.

[72] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu, H. Zhu, and B. Li, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach," *Computers & Security*, vol. 73, pp. 326–344, 2018.

[73] Y. Shi, T. Erpek, Y. E. Sagduyu, and J. H. Li, "Spectrum data poisoning with adversarial deep learning.," in *IEEE Military Commun. Conf. (MILCOM)*, 2018.

[74] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25nd Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-221, 2018*, The Internet Society, 2018.

[75] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *eprint arXiv:1605.07277*, p. arXiv:1605.07277, 2016.

[76] N. Papernot, N. Carlini, I. Goodfellow, R. Feinman, F. Faghri, A. Matyasko, K. Hambardzumyan, Y.-L. Juang, A. Kurakin, and R. Sheatsley, "cleverhans v2. 0.0: an adversarial machine learning library," *arXiv preprint arXiv:1610.00768*, 2016.

[77] Y. Shi and Y. E. Sagduyu, "Evasion and causative attacks with adversarial deep learning," in *Military Commun. Conf. (MILCOM)*, IEEE, 2017.

[78] T. Newman and T. Clancy, "Security threats to cognitive radio signal classifiers," in *Virginia Tech Wireless Personal Commun. Symp.*, 2009.

[79] T. C. Clancy and A. Khawar, "Security threats to signal classifiers using self-organizing maps," in *4th Int. Conf. on Cognitive Radio Oriented Wireless Netw. and Commun.*, pp. 1–6, 2009.

[80] T. J. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in *Proc. of the GNU Radio Conf.*, vol. 1, 2016.

[81] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[82] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," *arXiv:1703.09387*, 2017.

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

[84] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456, JMLR.org, 2015.

[85] S. Kokalj-Filipovic, R. Miller, N. Chang, and C. L. Lau, "Mitigation of adversarial examples in rf deep classifiers utilizing autoencoder pre-training," *arXiv preprint arXiv:1902.08034*, 2019.