Enhancing Communications Aware Evasion Attacks on RFML Spectrum Sensing Systems

Matthew D. DelVecchio

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

Electrical Engineering

William C. Headley, Chair

Richard M. Buehrer

Ryan K. Williams

Joseph D. Gaeddert

July 30, 2020

Blacksburg, Virginia

Keywords: Spectrum Sensing, Adversarial Machine Learning, Radio Frequency Machine Learning, Communications Security

Enhancing Communications Aware Evasion Attacks on RFML Spectrum Sensing Systems

Matthew D. DelVecchio

(ABSTRACT)

Recent innovations in machine learning have paved the way for new capabilities in the field of radio frequency (RF) communications. Machine learning techniques such as reinforcement learning and deep neural networks (DNN) can be leveraged to improve upon traditional wireless communications methods so that they no longer require expertly-defined features. Simultaneously, cybersecurity and electronic warfare are growing areas of focus and concern in an increasingly technology-driven world. Privacy and confidentiality of communication links are both more important and more difficult than ever in the current high threat environment. RF machine learning (RFML) systems contribute to this threat as they have been shown to be successful in gleaning information from intercepted signals, through the use of learning-enabled eavesdroppers. This thesis focuses on a method of defense against such communications threats termed an adversarial evasion attack in which intelligently crafted perturbations of the RF signal are used to fool a DNN-enabled classifier, therefore securing the communications channel.

One often overlooked aspect of evasion attacks is the concept of maintaining intended use. In other words, while an adversarial signal, or more generally an adversarial example, should fool the DNN it is attacking, this should not come at the detriment to it's primary application. In RF communications, this manifests in the idea that the communications link must be successfully maintained with friendly receivers, even when executing an evasion attack against malicious receivers. This is a difficult scenario, made even more so by the nature of

channel effects present in over-the-air (OTA) communications, as is assumed in this work. Previous work in this field has introduced a form of evasion attack for RFML systems called a communications aware attack that explicitly addresses the reliable communications aspect of the attack by training a separate DNN to craft adversarial signals; however, this work did not utilize the full RF processing chain and left residual indicators of the attack that could be leveraged for defensive capabilities. First, this thesis focuses on implementing forward error correction (FEC), an aspect present in most communications systems, in the training process of the attack. It is shown that introducing this into the training stage allows the communications aware attack to implicitly use the structure of the coding to create smarter and more efficient adversarial signals. Secondly, this thesis then addresses the fact that in previous work, the resulting adversarial signal exhibiting significant out-of-band frequency content, a limitation that can be used to render the attack ineffective if preprocessing at the attacked DNN is assumed. This thesis presents two novel approaches to solve this problem and eliminate the majority of side content in the attack. By doing so, the communications aware attack is more readily applicable to real-world scenarios.

Enhancing Communications Aware Evasion Attacks on RFML Spectrum Sensing Systems

Matthew D. DelVecchio

(GENERAL AUDIENCE ABSTRACT)

Deep learning has started infiltrating many aspects of society from the military, to academia, to commercial vendors. Additionally, with the recent deployment of 5G technology, connectivity is more readily accessible than ever and an increasingly large number of systems will communicate with one another across the globe. However, cybersecurity and electronic warfare call into question the very notion of privacy and confidentiality of data and communication streams. Deep learning has further improved these intercepting capabilities. However, these deep learning systems have also been shown to be vulnerable to attack. This thesis exists at the nexus of these two problems, both machine learning and communication security. This work expands upon adversarial evasion attacks meant to help elude signal classification at a deep learning-enabled eavesdropper while still providing reliable communications to a friendly receiver. By doing so, this work both provides a new methodology that can be used to conceal communication information from unwanted parties while also highlighting the glaring vulnerabilities present in machine learning systems.

Dedication

To my family and friends who have shown unwavering support.

Acknowledgments

I would like to thank Dr. William "Chris" Headley for his help and support over the years. Throughout both my graduate and undergraduate studies he helped cultivate and grow my interest in research. I would not be here without his guidance and am highly grateful for the countless hours he spent advising me. He pushed me to be a better researcher and student, something I will carry with me throughout the rest my career and life. I would also like to thank Bryse Flowers for his guidance the last two years. Without his his continued support, my work would not have been possible. I could always look to him for help. I also thank the Hume Center for giving me the ability to study such interesting work and providing the support necessary to do so. I will miss working with everyone there.

I would also like to acknowledge and thank Dr. Joseph Gaeddert, Dr. Ryan Williams, and Dr. R. Michael Buehrer for serving on my committee and offering extremely helpful feedback and advice. I appreciate you taking the time to help me. Finally, I would like to thank all of my family and friends who have supported me throughout my studies and made even the hard times manageable and enjoyable. Without you, I would not be where I am today.

This material is based upon work supported by the National Science Foundation under Grant Number 1303297. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Contents

List of Figures

List of Abbreviations

AAE Adversarial Auto-Encoder

AI Artificial Intelligence

AMC Automatic Modulation Classification

AMN Adversarial Mutation Network

ARN Adversarial Residual Network

ATN Adversarial Transformation Network

AWGN Additive White Gaussian Noise

BER Bit Error Rate

CNN Convolutional Neural Network

CV Computer Vision

DNN Deep Neural Network

EVM Error Vector Magnitude

FEC Forward Error Correction

FFT Fast Fourier Transform

FGSM Fast Gradient Sign Method

IFFT Inverse Fast Fourier Transform

IQ In-phase and Quadrature

MAE Mean Absolute Error

MI-FGSM Momentum Interative - FGSM

MSE Mean Squared Error

OFDM Orthogonal Frequency Division Multiplexing

OTA Over-the-Air

P-ATN Perturbation - Adversarial Transformation Network

PSD Power Spectral Density

RF Radio Frequency

RFFE Radio Frequency Front End

RFML Radio Frequency Machine Learning

RNN Recurrent Neural Network

RRC Root-Raised-Cosine

SJR Signal-to-Jamming Ratio

SNR Signal-to-Noise Ratio

Chapter 1

Introduction and Motivation

Deep learning and other forms of artificial intelligence (AI) have experienced extreme growth and focus in recent years. Numerous industries and fields of study have scrambled to adapt machine learning techniques within their own applications. Coupled with the recent development in areas such as Internet of Things (IoT), 5G technology, and computing capabilities, machine learning looks to be at the forefront of cutting edge technology ranging from self driving cars [1] to network intrusion detection [2, 3].

One field that has benefited greatly from the rise in machine learning capabilities is that of wireless communication and radio frequency (RF) engineering. While image recognition and Computer Vision (CV) have been the predominant focus of research ([4, 5, 6]) in the field of AI, communication systems have found ways to leverage these concepts to enhance their own functionality in a field often referred to as radio frequency machine learning (RFML) [7, 8, 9, 10, 11, 12]. Cognitive radios (CR), for example, utilize deep learning strategies to more quickly and efficiently adapt to the wide range of changing and complex environment in which they are deployed [13, 14]. Machine learning has also paved the way for improvements in areas such as direction of arrival calculations [15, 16], spectrum allocation [13, 17], and spectrum sensing [18].

One such area of improvement for spectrum sensing is in the area of automatic modulation classification (AMC) [19, 20, 21, 22, 23, 24]. In this application, a deep neural network (DNN) typically operates on raw in-phase and quadrature (IQ) data to perform modulation

classification of the RF communications signal. Traditional methods for signal classification required expert knowledge and extraction of features, often necessitating a human operator and strong assumptions of *a priori* knowledge [25, 26, 27, 28, 29]; however, by leveraging the learning capabilities of a DNN, an AMC can perform this task more autonomously by first automatically pulling features, without required feedback from an expert, and then determining the underlying modulation scheme based on learned understandings of these features. The ability to apply a DNN, and more generally any machine learning approach, to raw communications data opens the door to a large number of improvements in the RF communication environment.

Given the potential widespread deployment of not only RFML systems but deep learning-enabled applications in general, the security of these networks and techniques has been the focus of significant research [30, 31, 32, 33, 34, 35, 36]. The study of adversarial machine learning encompasses a variety of attack scenarios on deep learning systems. For instance, privacy attacks may look to glean information about the structure and architecture of a DNN [37, 38], while an evasion attack may aim to evade correct detection by a classifier [7, 31, 35]. In the context of RFML, adversarial machine learning can be used to evade classification by an AMC network, although this is just one example [39, 40, 41, 42] . These forms of attack have all been shown to be extremely successful in degrading or hindering the performance and execution of machine learning applications.

The study of evasion attacks against RFML systems aids in two extremely important areas. The first is in understanding the security limitations of machine learning and therefore paving the way for research into defense and hardening schemes, a growing area of focus. The second lies in the attack's applicability to securing communication links against RFML-enabled systems. Recently, the intertwined fields of electronic warfare and cybersecurity have grown in importance and prominence as digital communications and networking become a larger

driving force in global operations and connectivity. The need for secure communications is larger than ever given both the dependence on this form of infrastructure and the increased capabilities that adversaries can employ against it, such as deep learning-enabled systems. For example, RFML applications have shown to be successful in detecting, disrupting, and extracting information from communication transmissions [43, 44, 45]. For instance, correctly classifying the modulation scheme of a transmission using an AMC can be the first stage in a cyber-attack as it can help extract underlying features such as data rate [45, 46]. Evasion attacks on RFML systems can therefore be seen as a method for securing communications against malicious actors.

One of the most difficult aspects of evasion attacks on RFML systems is the underlying trade-off between degrading the classifier's performance and maintaining reliable communications with an intended receiver [7, 12, 47]. This is a problem that isn't as pertinent in image recognition where the translation of this trade-off manifests in the idea that a human should still successfully perceive the image. Ensuring an un-witting receiver is still able to perceive an adversarial RF signal is a much more complicated process. This assumption of a simplistic receiver is made envisioning a scenario where the receiver either has limited processing power, is a legacy system, or cannot be made adaptable to a changing form of transmission, all things extremely applicable within a military context. This complicated process is made even more so by the existence of channel effects inherent in over-the-air (OTA) communications [40, 41]. Additionally, as research in the field of RFML evasion attacks continues to grow, deployed eavesdroppers will become more privy to attacks against them, potentially leveraging preprocessing stages to limit the effectiveness of the attack. Given this, it is becoming increasingly important for evasion attacks in the RFML domain to adapt and improve to both take advantage of features inherent in the communication chain as well as address their own limitations.

1.1 Research Contributions

This thesis makes the following contributions:

- Chapter 2 and [48] present an improvement over traditional fast gradient sign method (FGSM) evasion attacks on an RFML system through implementation of a momentum iterative - FGSM (MI-FGSM) attack.
- Chapter 3 and [47] provide updates to the communications aware attack framework introduced in [12] that incorporates forward error correction (FEC) into the attack process. Using these updates, it then shows that the attack framework can leverage FEC redundancy both implicitly (no knowledge of the coding) and explicitly (information of the coding is built into the architecture) to improve attack success.
- Chapter 4 and [49] introduce a new metric, called spectral integrity, to be considered alongside communication and evasion success for RF evasion attacks to address out-of-band frequency content seen in the adversarial signals of prior work. It then presents two novel solutions to this problem, one that assumes access to the intermediate transmission stages, and one that does not.

1.2 Thesis Outline

Chapter 2 provides background on the concept of RFML which serves as the starting point for all of the work presented in this thesis. It further defines the concept of adversarial machine learning and illustrates common attack approaches such as evasion attacks. With this in mind, a discussion of traditional evasion attack methods, used not only in RFML but also other communities within the field of machine learning, is presented. More specifically, the

gradient-based evasion attack algorithms of FGSM and MI-FGSM are highlighted as well as their limitations when considering the communications link required in RF. This limitation lies in that these methods do not provide ways to explicitly prioritize and improve communications success other than just weakening the attack. Given this, Chapter 2 concludes with a description of a recently developed framework called a communications aware evasion attack, presented in [12], that more explicitly considers the communication link existent in RFML. This attack uses a separate DNN that learns to craft a perturbation to be added to the signal to create an adversarial signal used to execute the evasion attack. The work presented in this thesis builds off this framework.

Chapter 3 considers the adoption of Forward Error Correction (FEC) coding in the context of RFML evasion attacks. It starts by defining the attack environment and actors present throughout this work. Details about the configuration of the system model are provided. The chapter then considers the communications aware attack of [12] and provides updates to the training process such that the attack is better optimized for FEC-enabled systems, as most RF applications use this feature. This is followed by illustrating the chapter's main contribution of utilizing the attack methodology to implicitly learn the FEC coding scheme to then craft more intelligent adversarial signals for the evasion attack. In this work, implicitly learning the FEC means that no information of the FEC structure, or even that it exists, is provided in the attack process. Alternatively, explicit knowledge of the FEC means that information about the coding structure is built into the architectural configuration of the attack. Results are presented that show that the communication reliability increases when the attack network is trained with FEC-enabled signals. Further, the results then show that the trade-off between communication and evasion success is improved, meaning that improved evasion ability does not degrade communications as significantly as in prior work. It is then shown that the new adaptation of this attack framework unintentionally

creates adversarial signals that hold better spectral characteristics. More specifically, the resulting adversarial signals are more in-band than the previous work and exhibit similar properties in the frequency domain to clean signals. This helps avoid detection and limits the ability for defensive mechanisms against the attack that utilize preprocessing of the signal to eliminate out-of-band perturbations. Finally, an additional update to the attack framework is presented to allow for providing explicit information regarding the code scheme to the attacking network during training. This is done through architectural changes to the network and is shown to further improve evasion success.

Chapter 4 focuses on the limitation of the communications aware framework in which the adversarial signal exhibits significant side content in the frequency domain. While the updates in Chapter 3 showed improvements in this area of focus, the perturbations still contained noticeable out-of-band content. This could allow the perturbation to be removed with preprocessing and render the attack ineffective. Additionally, communications in commercial environments are typically allotted specific frequency bands and the out-of-band perturbation would make the resulting adversarial signal non-compliant. Two solutions are presented in this chapter. The first considers perturbing the symbols of the communication link rather than the samples, as was previously done. This is shown to successfully shape the adversarial signal to be in-band, similar to a clean signal, without much detriment to the communication link and evasion success. This method provides improvement over the previous attack that perturbed the samples; however, this assumes modifications to the intermediate stages in the transmission process, something that may not be feasible in a scenario where the attack is plugged in at the end or is combined with the clean signal over the air. Therefore, a second approach is presented that considers a substituted loss function to use when training the attack, coined the spectral deception loss. This loss looks to deceive an eavesdropper by prioritizing the spectral characteristics of the adversarial signal during training such that it

looks spectrally like a benign signal. A variety of methods for this loss are proposed, implemented, and analyzed. The deception loss is shown to be successful in removing out-of-band perturbation, however this is at the detriment of the communication link as the communication success decreases when the spectral shape is more highly prioritized. It is shown that the training configuration for the attack can be adapted to allow for an adjustable trade-off between communication success and in-band frequency content.

The thesis then concludes in Chapter 5 with a summary of the results and ideas introduced throughout this thesis. This is followed by a discussion on avenues for future work, both to roll back assumptions made, and to expand and improve on key points of this work.

1.3 Relevant Publications

Thesis Publications

- Matthew DelVecchio, Bryse Flowers, and William C. Headley. "Effects of forward error correction on communications aware evasion attacks." *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (accepted PIMRC)*, September 2020.
- Matthew DelVecchio, Vanessa Arndorfer, and William C. Headley. "Investigating a spectral deception loss metric for training machine learning-based evasion attacks." *ACM Workshop on Wireless Security and Machine Learning (accepted WiseML)*, July 2020.

Relevant Publications

- 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 first provides an overview on how improvements in machine learning have led to a wide variety of radio frequency (RF) use cases in which applying machine learning techniques has been shown to offer improvement over traditional methods. This is followed by more closely examining a subset of the radio frequency machine learning (RFML) field, in which a convolutional neural network (CNN) is used to perform modulation classification, forming the basis of the work presented in this thesis. This is then proceeded with a discussion on adversarial machine learning, a subset of this field that looks to degrade the performance of the machine learning applications, and more specifically for this work, decrease the modulation classification success. A variety of attack approaches are briefly described. Details are provided on gradient-based evasion attacks on modulation classification, a class of attacks that are the predecessor to the attack frameworks used in this work. More specifically, FGSM and MI-FGSM attacks are presented along with their limitations. Finally, a relatively new attack framework referred to as a communications aware attack, is discussed that more explicitly addresses and improves the communications success required in RF evasion attacks. This provides the starting point for the novel attack methodologies that are then developed in more detail in Chapters 3 and 4.

2.1 Radio Frequency Machine Learning

In recent years, the field of computing has seen increases in data storage, processing power, GPU capabilities, and architectural components. These enhancements have allowed machine learning techniques to grow in usage and success in a wide variety of applications. Machine learning requires a large quantity of input data and computational resources and while this once created a barrier to entry for their use, these are no longer restrictions. Machine learning, and similar focuses such as Artificial Intelligence (AI), can be applied to a wide variety of problems ranging from image recognition [4, 5], to medical diagnosis [50, 51, 52], to housing market predictions [53, 54]. While these are all valid and beneficial use cases, the research area that forms the focus of this thesis is that of RFML.

RFML, as defined in this work, focuses on using machine learning techniques to improve upon traditional methods used for communication and sensing in the radio frequency (RF) spectrum. Just as the general field of machine learning is very broad and applicable to many focus areas, RFML has been shown to have great success in numerous communication applications, utilizing a great number of different machine learning techniques. Techniques that have seen great success in the field of RFML include the CNN [20, 21, 22, 23], reinforcement learning [55, 56], the recurrent neural network (RNN) [57, 58], and auto-encoders [24, 59], among others. As the benefits of machine learning become more understood and the advancements in techniques are adapted for use in the communications community, this list will continue to grow. These methods have been shown in many cases to be more successful than more traditional methods. While the reason for improvement varies, it is often because traditional methods rely on expert features generated by a human whom may be unable to easily understand and make use of underlying patterns, or necessitates *a priori* knowledge or understanding of the operating environment that is not feasible. Machine learning approaches are good at being able to extract underlying features or patterns that make RF

solutions more feasible and generalized, [21, 22].

One such application of these techniques is using reinforcement learning to control radar usage such that collisions are avoided [55]. Results show that when using reinforcement learning to dictate the best plan of action when operating in a congested environment, there is a significant decrease in the number of collisions observed than when using a traditional method such as Sense-and-Avoid. This is due to the reinforcement learning algorithm's ability to make predictions based on probabilities and patterns that traditional methods are not designed to pick up on. Another application involves using an autoencoder to design physical layer encoding strategies for communications systems [59, 60]. Leveraging machine learning in this process allows for the communication to adapt to environments as needed and not require a closed-form strategy, leading to numerous benefits. These are just a couple examples of ways that machine learning has revolutionized the way that RF communication applications can be performed. Others have been studied thoroughly as well and the field is expanding rapidly with great potential. The machine learning area of focus that forms the cornerstone of this work and will be discussed more in depth below is that of RF Spectrum Sensing. Applications in this field typically operate on raw communication data to better understand or monitor the spectral environment. Some examples of this include signal detection or separation [61, 62] and emitter identification [8]. The predominant application considered in this thesis is signal classification, otherwise known as Automatic Modulation Classification (AMC).

2.1.1 RF Signal Classification

In the application of RF signal classification, communication data, typically in the form of raw, in-phase quadrature (IQ) points are fed into a machine learning network with the

goal of correctly classifying the modulation scheme used to transmit the data. Gleaning the modulation scheme from a stream of transmitted data is an important and helpful step in allowing a receiver to demodulate and retrieve the underlying bits from the communication. A variety of different machine learning techniques have proven to be successful in performing modulation classification, such as RNNs [57, 58]; however the approach utilized in this work is that of CNNs. CNNs have been shown to be very successful in this area due to their ability to perform automated feature extraction on the input data [20, 21, 22, 23]. This parallels one of the original uses of CNNs in the research field of image recognition where they were used to pull out features from the image for classification [63]. Traditional methods required expertly defined feature mapping in order to attempt modulation classification. CNNs, however, remove this step as they autonomously determine features within the network during training, allowing for faster feedback and easy adaptation to different communication environments and requiring less *a priori* knowledge. Deep knowledge of environmental factors is not as imperative as long as they are reflected in the training data; however, one con in these learning-enabled applications is therefore the large amount of representative data required.

The configuration and classification process for CNNs used in modulation classification occurs in two stages. The first encompasses the convolutional operations, the stage that performs the feature extraction. The layers consists of kernels that convolve over the input, in this case RF signal, and during training these kernels are updated such that the resulting output of the convolutional layer is the distinguishing features. Once the feature extraction is accomplished in this stage, they are then fed into a fully connected neural network that leverages these learned features to make a decision on the classification. A fully connected neural network consists of layers of nodes where each node in a given layer is fed the output of each node from the previous layer. These inputs are weighted and summed together

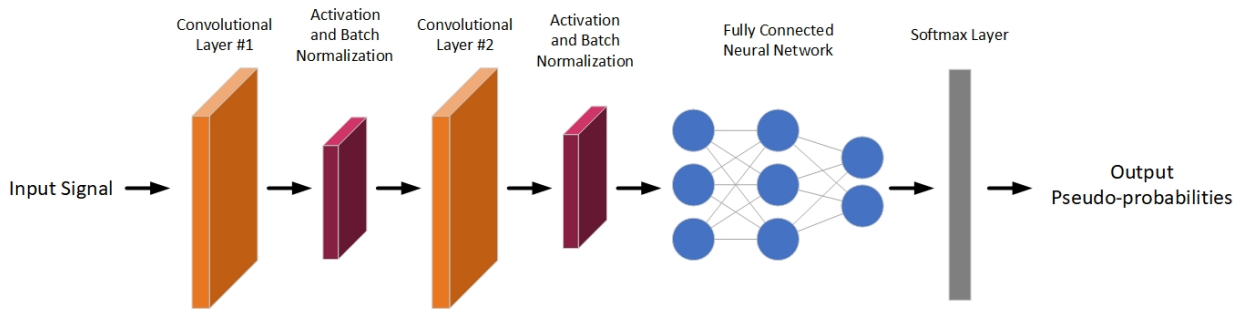


Figure 2.1: The architecture of the CNN used for RF signal classification in this work. This is comprised of 2 convolutional layers that extract the features and a fully connected neural network that uses these automatically learned features to determine psuedo-probabilites of each potential modulation class. While this is the architecture of the CNN primarily used in this work, the results can be generalized to other architectures as is studied in Section 3.3.3.

and run through an activation function to provide non-linearity to the process, creating the output of a given node. The output from the last layer is fed through a softmax function to create psuedo-probabilities for each potential modulation scheme, indicating the network's confidence in each. A generalized block diagram that illustrates the architecture of a CNN used for modulation classification is shown in Figure 2.1. A more specific description of the CNN configuration used in this work will be provided in Chapter 3 but this brief overview lays the foundation how the CNN is utilized throughout this work. Throughout the rest of this work, the usage of a CNN to accomplish modulation classification is also referred to as Automatic Modulation Classification (AMC).

2.2 Adversarial Machine Learning

Given the rapid increase in machine learning usage across a wide variety of research fields, it becomes imperative to understand the security of such systems. If machine learning becomes widely adapted in applications used in the real world, this could pose a very large security concern should these approaches be vulnerable to manipulation. Adversarial machine learning is the study of attacking machine learning networks using adversarial attack techniques.

The desired result of such an attack can vary from privacy attacks where the attacker learns information about the underlying network such as architectural choices [37, 38], to attacks where the network's ability to successfully operate is compromised [31, 32, 33]. There are numerous methods that can be used to accomplish these attacks but three that are elaborated on briefly in the proceeding sections are poisoning, Trojan, and evasion attacks due to their frequent use in the field of adversarial machine learning.

2.2.1 Poisoning Attacks

Poisoning attacks involve inserting malicious data into the training process so that the trained network behaves either in a way that the attacker desires (such as improperly classifying one specific class) or generally poorly for all scenarios [64, 65, 66]. One drawback to this form of attack is that it requires the attacker to have direct access to the data used to train the network in order to successfully inject poisoned data. While on the surface this may seem impractical, this has become an increasingly reasonable assumption in the current machine learning landscape. This is due to the fact that, in order to satisfy the need for large quantities of data for training and testing machine learning networks, many users have relied on open source datasets, using the machine learning community to help fulfil the data need [67, 68]. While generally beneficial for availability of data, ease of reproducibilty, and ability to leverage expert knowledge, attackers can use this open source concept to either insert malicious data into existing datasets or make their own poisoned datasets available for use. Additionally, many machine learning networks utilize an on-line training process that continues to train and update its weights using data fed into it while it is in use. This is used to help ensure the network is adapting to changes in data but opens the network up to poisoning attacks. Knowing this, an attacker could feed poisoned data to the system while it is on-line so that over time it becomes corrupted [64].

2.2.2 Trojan Attacks

Like poisoning attacks, Trojan attacks make use of malicious injections into the training process of a machine learning network to alter its functionality [69, 70]; however, unlike poisoning attacks that reduce a network's success for benign data, Trojan attacks operate much more stealthily. The goal of a Trojan attack is to change the network architecture in such a way that it operates as expected for normal data, but when presented with very specific malicious data, behaves in an orthogonal way to its intended use. The malicious data is similar to benign data except for a smaller, relatively unobservable added feature called a Trojan stamp. An example of a stamp, in the context of image recognition, could be a small logo that appears in the corner of an image. When this stamp is present, the network behaves in the way desired by the adversary. This is done so that the malicious entity can avoid detection or cause poor performance for a small subset of inputs without raising suspicion that the network is corrupt. This form of attack requires both access to the dataset used for training as well as the architecture and weights of the network being attacked in order to craft a successful stamp. This is because the network has to be fed the stamped data in order to perform the way the malicious actor desires when presented with it, and the stamp must be crafted in such a way that the network will key in on it when present. Generating a random stamp is not successful if the network does not extract it when performing feature extraction. Additionally, the stamp should be created such that it is not observable to humans, otherwise detection of the attack would be possible. Due to the complexity of the attack and the number of assumption that need to be made about the attack environment, this is a much less researched form of adversarial machine learning but one that has started to see increased focus in recent years.

2.2.3 Evasion Attacks

As has been highlighted, the previous two attack techniques discussed require access to the training data of a machine learning network; however, this may not be feasible in some scenarios. Evasion attacks provide a different attack vector. In this framework, rather than ingesting malicious data in the training process, adversarial input data is provided while the network is in use after training and testing [31, 33, 35]. The adversarial data is created by making small changes, referred to as perturbations in this work, that cause the network to react differently to the data. The perturbations are created with the goal in mind of minimally changing the underlying original data. This is important because providing a perturbed data sample that is essentially just noise has very limited use case. This form of attack requires no changes to the attacked network, something that is ideal for many scenarios, as instead the malicious actor is changing their own data in order to evade detection; however, many evasion attacks require access to, or knowledge of, the network architecture and weight values in order to craft a successful adversarial input. It has been shown that techniques such as transfer learning have been successful in enabling evasion attack on networks that the adversary does not have access to. Transfer learning involves carrying out an attack on a network that the adversary creates or has access to that is similar in architecture and configuration to the true attacked network [71]. The process used to attack the known network is then transferred to the desired network.

For an evasion attack, specifically as they apply to attacks against AMC networks, there are multiple different goals that the malicious actor can seek to achieve. One of the simplest goals is that of confidence reduction. In this scenario, the goal is for the attack to lower the confidence that the network has in its classification. For instance, if the network was previously 95% certain of the correct label and is now only 60%. The classification can still be successful, but this pushes the network closer to misclassification under more adverse

environments or may cause the network to be taken out of use due to the lower probability of correct classification. Alternatively, a much more sought after goal is that of true misclassification which occurs when the confidence in the true classification is lowered so much that a different and incorrect label now has a higher probability of being correct. This results in incorrect operation of the network.

In practice, misclassification can take two different approaches: untargeted and targeted [7]. In the simpler approach of the two, untargeted attacks attempt to cause the machine learning network to classify an input as anything other than the true label. As long as the output of the network is not the true classification, the result does not matter. On the other hand, in targeted misclassification attacks, the malicious actor attempts to force the network to classify an input as one specific incorrect label rather than any incorrect label. An example of these two techniques can be illustrated with a network used to classify handwritten numbers as a value between 0 and 9 [72]. If a given input was a 5, an untargeted attack would have the goal of making the network output anything other than 5, while a targeted attack would instead attempt to make the network classify it as something specific such as 8. Classifying the number as a 3 would then be successful for the untargeted scenario but not the targeted scenario. As a result, it can be seen that a targeted attack is much more difficult to carry out than an untargeted attack. For this reason, untargeted attacks are the predominant focus of this work as it is important to ensure that this scenario is successful before addressing the more complex targeted attacks.

The overarching and primary focus of all contributions in this work is to successfully execute evasion attacks. The machine learning network under attack throughout this work is the AMC described in previous sections. Using different methods and frameworks, raw IQ data will be perturbed in order to perform evasion attacks on AMC networks and avoid correct classification. Chapters 3 and 4 will elaborate on evasion attack methodologies and strategies

that leverage and/or address specific characteristics present in RF communications, namely forward error correction and spectral shape. The rest of this chapter is devoted to illustrating evasion attack techniques previously employed against AMC networks.

2.3 Modulation Obfuscation

As was stated, the primary focus of this work is to consider and implement evasion attacks against AMC networks in order to force modulation scheme misclassification; however, this is not the only method that can be used to hide the modulation scheme for an eavesdropper. Modulation obfuscation has been developed as a method for doing so using a technique that masks the true modulation scheme using a predefined strategy known by both the transmitter and any friendly receiver. In this method, the data is altered to appear like the highest-order modulation supported by the receiver. This has proved to be successful in thwarting AMC network's ability to correctly classify signals [46, 73]; however, this can require much more computational capacity at the receiver since it needs to both synchronize knowledge of the obfuscation strategy with the transmitter and have the processing capabilities to extract the true modulation and data. This is not always feasible in scenarios where the receiver is resource constrained or non-adaptable. Additionally, targeted attacks may not be feasible since modulation schemes are typically masked as whatever highest-order modulation scheme is allowed at the receiver. Therefore targeting a lower-order modulation scheme wouldn't be possible. For these reasons, adversarial machine learning approaches are employed in this work given their ability to perform targeted attacks as well as allow for simplistic receivers. Additionally, while this work focuses on attacking AMC networks, it can be generalized for other RFML spectrum sensing applications, something not true of modulation obfuscation.

2.4 Gradient-based Evasion Attacks

In the literature, especially for image-based application, one of the predominant ways to carry out an evasion attack is through the use of gradients [7, 31, 33]. The error between the desired attacked behavior (such as a misclassification) and the actual behavior of the network is calculated using a loss function. Then, the gradient of this error with respect to the signal is calculated and this is used to perturb the input. In image recognition applications, this means altering the pixels of the image. In spectrum sensing this instead means altering the raw IQ data (referred to as an adversarial signal throughout this work). This gradient process is carried out for every separate adversarial signal that needs to be perturbed. One of the primary methods utilized is the Fast Gradient Signed Method (FGSM). This is discussed in more detail below.

2.4.1 FGSM

During the training process for a DNN, a loss function is used to calculate and quantify the difference between the true output of the network and the target output. The gradient of this loss with respect to the weights of the DNN is calculated and back-propogated through the network, updating the weights until the network converges to the proper behavior. In gradient-based adversarial attacks, this same process is used instead to update the input to evade classification rather than updating the weights of the network to successfully classify inputs. FGSM is a gradient-based approach that was introduced as an algorithm for creating adversarial images in computer vision (CV) [31]. FGSM works to perturb the data by taking a single step with respect to the sign of the gradient of the loss function with a size determined by a given value, ϵ. The loss function is similar to what would be used in training a DNN and quantifies the error between how the adversarial input is classified and how the attack

wants this input to be classified. The function used to calculate the loss varies depending on the application but some examples are mean squared error, mean absolute error, or cross-entropy. The goal of an untargeted evasion attack using any method, but specifically FGSM in this case, from an evasion standpoint is to optimize

$$\underset{\boldsymbol{x}^*}{\operatorname{argmax}} \mathcal{L}(f(\boldsymbol{\theta}, \boldsymbol{x}^*), \boldsymbol{y}_s)) \tag{2.1}$$

where $\mathcal{L}$ is the loss function of choice, f is the network architecture, $\boldsymbol{x}^*$ is the perturbed signal, $\boldsymbol{\theta}$ is the network weight parameters, and $\boldsymbol{y}_s$ is the true modulation scheme employed. The network's classification prediction is denoted as $f(\boldsymbol{\theta}, \boldsymbol{x}^*)$. If the adversary instead engages in a targeted attack, the optimization goal changes to become

$$\underset{\boldsymbol{x}^*}{\operatorname{argmin}} \mathcal{L}(f(\boldsymbol{\theta}, \boldsymbol{x}^*), \boldsymbol{y}_t)) \tag{2.2}$$

where the perturbation looks to minimize the difference between the network's predicted classification based on the adversarial signal, $f(\boldsymbol{\theta}, \boldsymbol{x}^*)$ and the targeted modulation class, $\boldsymbol{y}_t$.

In both the targeted and untargeted scenario, the gradient of the loss function is calculated and used to create the adversarial signal. FGSM creates adversarial samples by performing

$$x^* = x + \epsilon \times sign(\nabla_x \mathcal{L}(\cdot)) \tag{2.3}$$

where x^* is the new, perturbed signal, x is the original signal, and where y_s is the true label of x. $\mathcal{L}(\cdot)$ represents the loss function, either the maximization problem (2.1) for untargeted attacks or the minimization problem (2.2) for targeted attacks. The only difference is that the original signal, x, is used to calculate the loss, rather than the adversarial, x^*. The sign of the calculated gradient of the loss with respect to the original signal is what determines

the direction (positive or negative) of the update while the constant, ϵ, determines the magnitude. As seen, this method only needs to process the signal on one pass, meaning it is computationally efficient, since it only takes one step. In the case of FGSM evasion attacks on RF signals, the limiting constant ϵ is adjusted to meet the desired level of E_s/E_j or signal-to-jamming ratio (SJR). E_s/E_j is the ratio of the power of the original signal to the power of the jamming signal, or perturbation in this case. A larger ϵ allows for a larger perturbation and therefore a smaller SJR value. If trying to minimize the corruption to the original signal, the ϵ should therefore be decreased.

FGSM is a computationally efficient algorithm due to its single step at the cost of providing a less fine-tuned perturbation. Updates to the FGSM process have been studied that, while more complex, can provide a better, more optimized adversarial signal. One such method is the Momentum, Iterative FGSM approach (MI-FGSM) described in the following subsection.

2.4.2 MI-FGSM

MI-FGSM was introduced as an improvement over FGSM that creates better optimized adversarial inputs [34]. Instead of just taking one step in the direction of the gradient, the MI-FGSM algorithm takes several smaller steps that are ϵ/T in size, where T is the number of iterations, allowing for more fine tuning of the perturbation. This has been shown to be an improvement over FGSM and older techniques.

MI-FGSM is similar to FGSM, however it introduces a momentum factor in order to help the attack converge and an iterative approach to create a more fine tuned perturbation rather than the more granular perturbation created by just one step in FGSM. Before finding the perturbation, the gradient of the current iteration, where t is the current iteration, of the

perturbed signal is given by (2.4),

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x \mathcal{L}(f(\theta, x_t^*), y_s)}{||\nabla_x \mathcal{L}(f(\theta, x_t^*), y_s)||_1} \tag{2.4}$$

where g is the gradient and μ is some momentum value. (2.4) is for an untargeted attack due to the use of y_s. In a targeted attack, the gradient of the current iteration is found by instead replacing (2.1) in (2.4) with the targeted equation, (2.2). Once the current gradient is found, it is possible to find the perturbed signal of the current iteration by (2.5),

$$x_{t+1}^* = x_t^* + (\epsilon/T) \cdot \text{sign}(g_{t+1}) \tag{2.5}$$

where T is the total number of iterations. After iterating T times, x_{t+1}^* is the final perturbed signal that is used.

Work presented in [48] performed analysis on the success of MI-FGSM attacks for both untargeted and targeted scenarios against AMC networks. For each approach, the attack environment was structured as a direct access attack, something introduced formally in [7]. A direct access attack is defined as the scenario in which the attacker is able to directly perturb signals at the input to the attacked network. This differs from other attack environments such as self-protect attacks when the attacker must transmit the perturbed signal over-the-air to the network, or cover attacks when the attacker transmits a perturbation over-the-air with the goal of combining it with a benign signal from a separate transmitter that will then propagate to the network. The latter two attack environments are more practical to real world communications scenarios, however direct access was used in [48] to provide an initial examination into the usability of MI-FGSM on communication signals. Given that direct access attacks do not need to consider and adapt to channel effects, since they assume the adversarial signals are fed directly into the network rather than transmitted over the air,

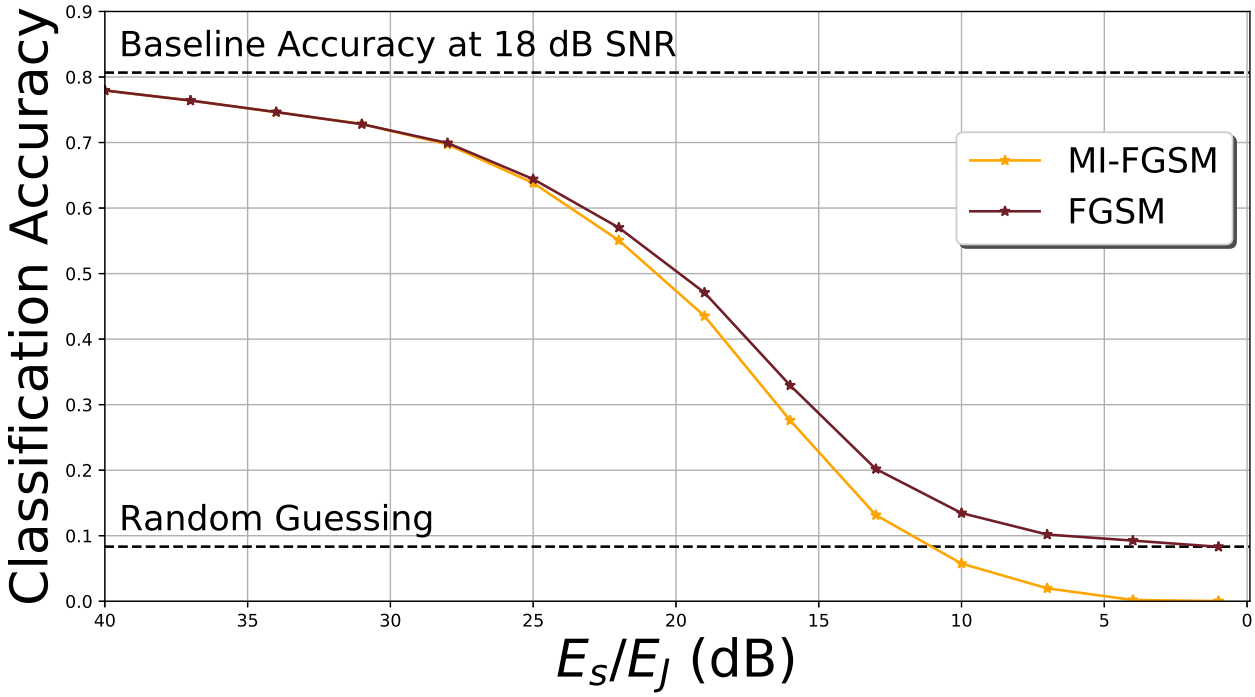


Figure 2.2: Direct access untargeted attacks using MI-FGSM and FGSM, showing the classification accuracy over E_s/E_j.

this scenario provides the ideal test for suitability and initial success. If it doesn't work for direct access, then it most likely will not work when translated to more complex attacks.

Results from [48] indicate that MI-FGSM performs better than FGSM when considering an untargeted attack. Figure 2.2 showcases this advantage. The figure is a plot of the model's classification accuracy versus E_s/E_j (SJR). The further to the left a data point is, the less powerful the perturbation. The maroon (right) line on the plot shows the classification accuracies for adversarial examples generated using an FGSM while the orange (left) lines shows classification accuracies for adversarial examples generated by the MI-FGSM. As evident in Figure 2.2, MI-FGSM is able to create adversarial examples that are able to cause the model to misclassify to a higher degree than FGSM for a given E_s/E_j. MI-FGSM achieved this results due to the algorithm's ability to fine tune the perturbation over multiple iterations. Because MI-FGSM is able to accomplish this, at a similar classification accuracy level, it is

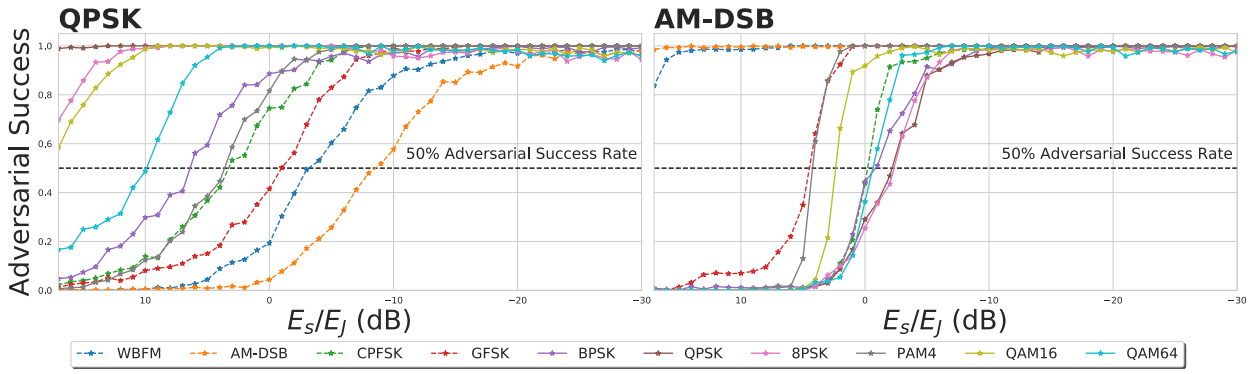


Figure 2.3: (left) QPSK and (right) AM-DSB source targeted classification accuracy over E_s/E_j.

able to produce similar results to FGSM at about a 10 dB advantage in the 15 to 0 SJR range, meaning that less of the original signal is perturbed. This power advantage decreases rapidly when E_s/E_j increases past 20 dB; however, in practice, an attack of this limited intensity would not be used because, on average, neither algorithm achieves untargeted success since the classification accuracy is above 50% indicating proper classification.

However, when moving to targeted attacks, some of the limitations of the gradient-based approaches, in this case MI-FGSM come to light. Figure 2.3 provides results on the ability for QPSK and AM-DSM signals to undergo targeted attacks for a total of 10 modulation classes when using MI-FGSM. In this figure, adversarial success is defined as the probability the AMC network has that the provided adversarial input is the targeted class. While both source modulations are eventually successful in misclassifying to the desired modulation, most don't reach even 50% success until around 0 dB SJR. 0 dB SJR occurs when the perturbation is just as powerful as the original signal. While this is allowable when considering the lone goal of evasion, this causes issues when considering a signal's primary goal of providing valid information. If an intended receiver of the signal cannot retrieve the underlying bits from the original signal, then the accomplishment of evasion is rather moot. Given the SJR value of approximately 0 dB required to perform targeted attacks, it is highly unlikely that these signals would offer reliable communication and would likely result in high bit error rate

(BER). Additionally, an attack with such a powerful perturbation would be easily noticeable to a human observer which is undesired since the attack should go undetected.

These results show that even though MI-FGSM is more efficient and less detrimental to the original signal than FGSM, it still struggles when carrying out targeted attacks. For this reason, it is unlikely that either FGSM or MI-FGSM attacks would be well adapted for the the more applicable communication scenario when an adversarial signal has to provide both effective communications and evasion success since the original signal is essentially destroyed by the perturbation in order to achieve targeted success. This is especially true given the results and analysis offered above are for direct access attacks, the most ideal and simplistic scenario. This calls into question the feasibility of using these methods in the given communications scenario. Due to these considerations, a framework of attack called a communication aware attack was introduced.

2.5 Communication-aware Evasion Attacks

When developing evasion attacks, there is a natural trade-off that arises between attack success and the intended application of the original input. For example, while the goal of an evasion attack against an AMC machine learning algorithm is to cause a misclassification and/or reduce user confidence, it is important that the perturbed signal still accomplish its intended use of still being successfully received by its intended target. In the field of image recognition, this manifests in the idea that an image perturbed by an adversarial attack should still be easily discernible, and even viewed as untouched, by a human observing the image [31]; however, this trade-off is much more difficult and complex when given in the context of RF communications. In this scenario, the adversarial signal isn't being interpreted by a human as with image recognition, but rather with a communications receiver. While

it is unlikely that small pixel changes will be perceptible to humans when considering image recognition, this same assumption cannot necessarily be made for receivers interpreting perturbed IQ data. This is the underlying difficulty with evasion attacks in the field of RF communication.

Without proper care, evasion attacks used to fool an AMC machine learning algorithm generally have a drastic negative impact on the communication link between the transmitter and intended receiver. This was seen in Figure 2.3 by the high SJR required. Unfortunately, gradient-based approaches do not provide a method for improving communications other than by decreasing the perturbation power. Recently, research has examined how evasion attacks can be improved in order to provide a better balance between the two conflicting goals of evasion and communication. Hameed et. al. [74] accomplished this by introducing a gradient descent training method to craft signal perturbations that utilize a combined target function that considers both evasion performance and BER. While BER is non-differentiable, and thus not suited to gradient based learning approaches, a gradient is estimated using simultaneous perturbation stochastic approximation (SPSA). This approaches offers improvement over previous methods where the perturbation was simply power limited, utilizing ϵ in the hope that this would lead to decreased BER. Flowers et al. improved upon these prior works through the development of a so-called "communications aware" attack [12] which forms the basis for the contributions of this work.

For the communications aware attack, an adversarial residual network (ARN) is leveraged in order to learn to make intelligent signal perturbations that balance the two opposing goals of evasion and communication. The ARN is itself a CNN that is fed the original signal and outputs the malicious perturbation that is then added to the signal to create the adversarial signal. The ARN, just like any CNN, is trained with the use of a loss function. In this approach, tailored loss functions can be used to enable certain behavior in

the resulting perturbation. This approach utilizes three separate loss functions to accomplish this: adversarial loss, communication loss, and power loss. These are defined as:

$$\mathcal{L}_{\text{adv}} = p_s^2 \tag{2.6}$$

$$\mathcal{L}_{\text{comm}} = \text{EVM}(S_{tx}, S_{tx+p})^2 + I_s \times \text{EVM}(S_{tx}, S_{tx+p}) \tag{2.7}$$

$$\mathcal{L}_{\text{pwr}} = \max(0, L - \frac{E_s}{E_p})^2 \tag{2.8}$$

where

- *Adversarial Loss* ($\mathcal{L}_{\text{adv}}$): prioritizes the ARN's ability to successfully learn to avoid classification by the eavesdropper. It is calculated using the confidence of the eavesdropper in the true source modulation, p_s, determined using the output of the final softmax layer in the eavesdropper's AMC.
- *Communication Loss* ($\mathcal{L}_{\text{comm}}$): prioritizes the AMN's ability to successfully learn to maintain the communication link between the transmitter and friendly receiver. It does this by using the error vector magnitude (EVM) between the clean symbols and the perturbed symbols, defined as $|S_{tx} - S_{tx+p}|$ as well as an indicator, I_s, that serves as a proxy for the symbol error rate (SER) and is 0 if the hard decision on the clean and perturbed symbols results in the same value and 1 otherwise.
- *Power Loss* ($\mathcal{L}_{\text{pwr}}$): Limits the power of the perturbation to be below some SJR, L.

These three losses are each weighted and summed together to guide the ARNs learning process:

$$\mathcal{L}_{\text{total}} = (1-\alpha)\mathcal{L}_{\text{adv}} + \alpha[(\beta)\mathcal{L}_{\text{comm}} + (1-\beta)\mathcal{L}_{\text{pwr}}] \tag{2.9}$$

Due to the existence of the adversarial loss and the communication loss, this attack framework is able to generate adversarial signals that balance the needs of evasion and communication needs more-so than the previous gradient-based approaches discussed. The weighting of each of these losses can be altered to meet the needs of the system. For instance, the adversarial loss can be prioritized more highly if evasion needs to be stressed or the communication loss can be more prioritized if reliable communication is needed. Through this training process, the ARN explicitly learns to generate signals that still provide strong communications whereas FGSM and MI-FGSM did this only indirectly through use of the limiting ϵ value.

However, this work did not take into account FEC, something present in most communication applications to help handle errors, which could help provide an additional attack vector for the ARN given the structured redundancy. Additionally, the resulting adversarial signals, as would be intuitively expected, took advantage of the assumed 8 times oversampled signal but in doing so, created adversarial signals with significant out-of-band frequency content. This limitation could cause the attack to be detected given its irregular nature, open the attack up to defensive mechanisms such as filtering at the eavesdropper to remove out-of-band perturbations, or make the transmission not comply with a spectral mask required in commercial use.

Chapter 3

Improving the Communications Aware Attack through FEC Coding

As discussed in the previous chapter, traditional evasion attacks typically assume that the intelligently crafted perturbations used in the attack are created and applied directly at the input to the Deep Neural Network (DNN). However, for most real-world communications applications, this assumption of direct access to the DNN is impractical. A more realistic scenario is that a transmitter must craft the adversarial signal before it is sent over the air. Therefore, for RFML applications, there are two key considerations that must be considered. First, the evasion attack must be resilient to the propagation channel between the signal transmitter and the DNN. Secondly, and perhaps most importantly, is the fact that the perturbations must minimize their impact on the intended receiver. While the previous works have considered the first goal in detail, there traditionally has been little work to date on consideration of the second goal. To remedy this, [12] developed a communications aware framework briefly touched on in the previous chapter that provides a mechanism for balancing the conflicting goals of successful communication and DNN evasion.

The work presented in this chapter extends the communications aware evasion attack framework through intelligent leveraging of forward error correction (FEC) coding. Due to the usage of FEC coding in the vast majority of real-world communication systems to correct errors that arise due to hardware impairments, channel propagation effects, etc., incorpo-

rating FEC in this framework is a natural extension of this prior work and is shown to improve performance in more adverse environments. More specifically, this work shows that the ARN inherently learns to utilize the nature of the coding to limit the negative impacts of altering the original signal while having a negligible impact on the attack's performance on the target DNN. In other words, this work shows that this intelligent perturbation can be learned without leveraging explicit knowledge of the FEC in the perturbation creation process. The goal of implicit learning of the code is beneficial because it allows the framework to be used on a variety of coding schemes without needing to adjust the configurations of the attack. However, while initially focusing on implicit learning, this work also shifts to address an adapted framework that provides explicit information of the coding scheme. While the communications aware framework is discussed herein in the context of AMC, the results can be generalized for other RFML applications of interest.

This chapter is broken down as follows. Section 3.1 provides a description of the overall system model and environment assumed in this attack. Section 3.2 lays out the framework for the communications aware attack that utilizes FEC. The results of this work and an analysis of the transmitter's ability to implicitly craft smarter perturbations with FEC is shown in Section 3.3. A shift is made in Section 3.4 to address providing explicit information of the FEC during training. Finally, this work concludes with a discussion of the key findings of this research as well as directions for future work.

3.1 System Model

Figure 3.1 depicts the wireless communications scenario for this work which consists of three main components: a transmitter, an intended receiver, and an eavesdropper.

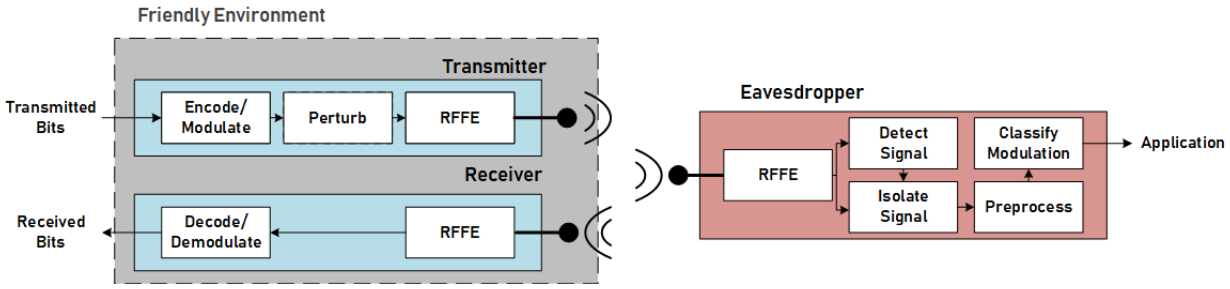


Figure 3.1: The wireless communications scenario considered within this work in which an intended communications link is being eavesdropped. The "perturb" block of the transmitter utilizes the developed communications aware attack framework to perturb the transmitted signal to evade the eavesdropper while minimizing the impact on the intended receiver.

3.1.1 Transmitter

The transmitter has two competing goals within this scenario: to successfully communicate with a naive receiver, and to evade modulation classification by an AMC-based eavesdropper. The receiver is considered naive because it has no knowledge of the attack, a situation applicable to when a transmitter needs to communicate with a legacy system. Here, the two metrics used to evaluate the success of these goals are, respectively, the bit error rate (BER) at the receiver and the modulation classification accuracy of the eavesdropper. Within the communications aware framework, the transmitter balances between these goals by crafting specially trained adversarial signals. In this chapter, the adversarial perturbations are produced by a specially trained network called an adversarial mutation network (AMN). This is represented by the "Perturb" block in Figure 3.1 and is described in Section 3.2.

Here, without loss of generality, the transmitted data is assumed to be modulated using a linear digital-amplitude phase modulation scheme (ASK, PSK, QAM, etc.) and is pulse shaped using a root-raised-cosine (RRC) filter. Unlike the previous work, the data is assumed to be encoded using an FEC code in order to add redundancy for correcting errors induced by the propagation channel. In particular, block codes are utilized within this work. Block codes are assumed in this work as they allow for a more recognizable pattern of redundancy

during encoding, given that they encode blocks of bits independently, which the AMN should be well suited to learn. This is because the AMN is developed using a CNN which operates over blocks of the input signal, so the structure of the AMN aligns well with block coding. There are a variety of block codes used in practice, such as Hamming, Golay, and SEC-DED, however this work examines Hamming codes and more specifically Hamming (7,4) as a more simplistic first step. This is a common code that takes 4 bits and encodes it into 7 bits for added redundancy. Once this attack framework is shown to be successful on Hamming codes, the methodology is also tested on convolutional codes in Section 3.3.4 to ensure transferability to other common codes.

As previously mentioned in this chapter, there are two possible approaches in this form of the attack: implicit learning of the FEC, and explicit. Implicit learning means that there is no indication that FEC is being used during training of the AMN. In other words, the AMN is not configured such that it is optimized to learn any particular coding structure and there is no change in its architecture across different FEC schemes. Explicit learning, however, means that architectural choices in the training process represents the existence and structure of the FEC and is done to allow the AMN to more efficiently learn and utilize the specific code in use. As an initial step, learning without explicit information of the coding is desirable so that the communications aware attack framework can easily be executed on different coding types without needing to change the architecture of the AMN, or attack framework, which is useful in modern automatic modulation and coding approaches.

In block codes, such as those examined, there exists a point of intersection between the theoretical BER curve for normal signals and that of the FEC-enabled signals. For signal-to-noise ratio (SNR) less than the SNR at the point of intersection, the FEC-enabled signals experience worse BER results during transmission over the air but then start to perform better after the intersection. These theoretical curves can be seen in the plots of the results

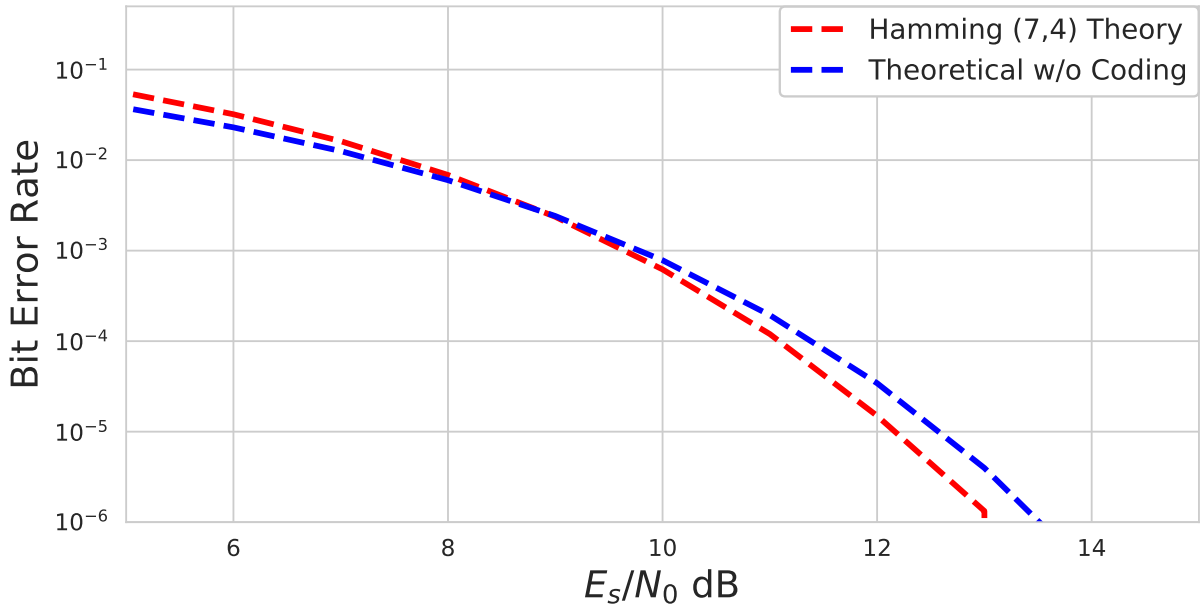


Figure 3.2: The theoretical BER of a QPSK signal for both Hamming (7,4) and un-coded. The intersection between the two occurs at 9 dB, showing that the region of operation for a QPSK signal with Hamming (7,4) encoding occurs after 9 dB SNR

section. For Hamming (7,4), most modulation schemes experience this intersection around 10 dB so for this reason, the range of 5-15 dB SNR was used in this work as it encompasses this region of trade-off. Figure 3.2 shows this trade-off and region of operation for a QPSK signal, as an example.

In order to guarantee that the eavesdropper's performance is not impacted by the FEC code itself, the transmitter also uses data whitening after FEC coding to whiten the bits and create a more uniform distribution of bits during transmission. This is common in real world signals such as Bluetooth. More specifically, this is done in order to guarantee that the eavesdropper is only impacted by the ability of the communications aware framework to create intelligent perturbations signals rather than the inherent difference between FEC-enabled and non FEC-enabled signal structures. The added redundancy in FEC-enabled signals could potentially adversely impact the eavesdropper's classification ability. Whitening occurs when a pre-

determined sequence of bits is XOR-ed with the original bit sequence (encoded bit sequence in this case) to provide the uniformity. This work uses the IBM version of a whitener which operates on a block of 8 bits at a time, using a 9-bit shift register for XOR operations [75].

3.1.2 Intended Receiver

Simply put, the role of the intended receiver is to successfully receive and interpret (by retrieving, demodulating, and decoding the bits) the transmitted data. In this work, the receiver is assumed to be static and unaware of the perturbations being applied at the transmitter. This models a scenario in which the transmitter is adaptable to changes in the environment (such as the presence of an eavesdropper) but the receiver is an already deployed legacy system that cannot be easily adapted on the fly. Additionally, for ease of analysis and preliminary performance analysis, it is assumed that the receiver and transmitter are synchronized (e.g. through the use of a header and/or control channel); however, there exists a noisy channel between the transmitter and receiver modeled using additive white Gaussian noise (AWGN).

3.1.3 Eavesdropper

The eavesdropper utilizes a state-of-the-art RFML approach to perform modulation classification using the raw IQ data sent by the transmitter. The eavesdropper is assumed to have limited knowledge of the transmitted signal and therefore must detect, isolate, and pre-process the raw data prior to modulation classification. It is not assumed that the eavesdropper is synchronized with the transmitter and there if therefore a time offset. This work focuses on disrupting the classification stage of this processing chain. More specifically, in this work it is assumed that the eavesdropper uses an AMC, like that presented in [22] and

discussed in Chapter 2, to perform modulation classification. The architecture of the CNN consists of 2 convolutional layers, each followed by batch normalization and a ReLU activation function, and a fully connected neural network. The fully connected network utilizes 2 dense layers of nodes also separated by batch normalization and ReLU activation. Finally, a softmax layer is used to translate the outputs into psuedo-probability values that represent the likelihood of each modulation class. The eavesdropper's CNN was trained using synthetic signal data created using LiquidDSP [76] with SNRs ranging from 0-20 dB for five modulation schemes (described in the following). While this is the architecture attacked in this work, this attack implementation is not specific to just this example and can instead be generalized to other architectures and DNN applications. Additionally, while the eavesdropper is employing an AMC network for modulation classification, this work can be generalized to other RFML applications. Like the receiver, the eavesdropper is assumed to have no knowledge of the communications aware framework and therefore does not react to the attack.

3.1.4 Data and Environmental Assumptions

In this work, without loss of generality, the modulation schemes considered are BPSK, QPSK, 8-PSK, 16-QAM, and 64-QAM. For this set of modulation schemes, the communications aware attack framework is trained and its performance is evaluated. The propagation channel between the transmitter and both the receiver and eavesdropper is assumed to be modeled by uncorrelated AWGN channels. For both training and testing the framework, the SNRs at the receiver and eavesdropper are assumed to be uniformly distributed between 5 and 15 dB. The SNRs and noise realizations between the transmitter and receiver and the transmitter and eavesdropper are assumed to be independent from one another and are each determined by two different random number generators that vary the SNRs during training. Additionally,

an integer sample time offset is introduced as a channel effect for the eavesdropper in order to assume asynchronous operation with the transmitter. This time offset is assumed to be uniformly distributed between 0 and 8 (the assumed number of samples per symbol). Prior work has shown that time offsets larger than the samples per symbol have little effect on the success of the adversarial attack [7]. Carrier frequency and phase offsets should also be considered in future work.

3.2 Communications Aware Framework

This section presents the methodology used to carry out a communications aware attack that utilizes FEC to deceive an eavesdropper while maintaining effective communication between the transmitter and receiver. A description of the nature of the AMN that is used to create the perturbed signal is provided first in this section. Then the custom loss functions used in this work to balance the goals of communication and evasion are discussed. These are updated and different than the losses used in [12]. Finally, an overall explanation of the training and testing procedure is presented.

3.2.1 Adversarial Mutation Network

In previous work, various gradient-based approaches have been used to generate perturbations of the original transmitted signal to achieve the goal of classification evasion [31, 34], as discussed in Chapter 2. While effective, these techniques typically focus on the success of the evasion with little regard for the intended communication link between transmitter and receiver. This process involves perturbing a signal using a loss gradient that is back-propagated through a surrogate classifier network to create an adversarial signal. This may

have to be done multiple times per signal block and must be done separately for every signal block transmitted, creating a very computationally-exhaustive process in a wireless transmitter that it typically resource constrained. Alternatively, a network known as a adversarial transformation network (ATN) makes use of a separate neural network to generate the perturbation automatically [35]. These networks can utilize custom loss functions in order to balance the adverse effect that perturbations have on the intended communication. Additionally, while the training process of a network requires a large number of computations, once trained it only requires one forward pass of a signal through the network in order to create a perturbed signal to be transmitted, which is much more computationally efficient than solutions relying on gradient-based optimization.

The work presented in [35] provides two examples of such a network: an adversarial autoencoder (AAE) and a perturbation - adversarial transformation network (P-ATN). While based on similar concepts, the output of these two networks are different. An AAE is provided an input signal and outputs the adversarial signal to be sent over the air. This signal is a learned combination of the original signal plus a perturbation and is represented by the equation

$$x^* = g(\theta, x) \tag{3.1}$$

where $g(\cdot)$ denotes the AAE, θ represents the parameters of the network learned during training, and x is the original signal. A P-ATN is given the same input signal as the AAE but instead outputs a perturbation that is then added to the original signal for transmission. The following equation shows this process.

$$x^* = x + g(\theta, x) \tag{3.2}$$

In summary, the AAE crafts the complete signal while the P-ATN creates a perturbation to

be added to a signal.

Previous work in communications aware attacks used a P-ATN to accomplish the goals of the transmitter [12] but this work shifts to the use of an AAE. While P-ATN implementations have simpler convergences (a P-ATN would only need to output 0 in order to transmit a signal optimal for communication while an AAE would need to learn to pass the original signal through the network unchanged), these introduce a different problem. The scaling of the perturbation with respect to the original signal must be done outside of the network. The P-ATN doesn't explicitly learn this scaling process. An AAE inherently learns to balance the power of the perturbation and original signal directly since it outputs the combined signal and therefore simplifies the process. Scaling is important because if the perturbation was significantly powerful, the transmitter would have an unfair advantage over the eavesdropper because the adversarial signal could essentially become random noise. Additionally, most communications are limited in the power that they can transmit. For this reason, the iteration of this attack developed in this thesis imposes a power limit on the AMN. This is different than the power limit of [12] because in this work, the power limit constrains the entire transmitted signal whereas the previous power limit was only imposed on the perturbation. The current work will utilize an AAE and refer to this as the AMN throughout the paper. The name is changed from the ARN of previous work because the network is now mutating the entire signal rather than generating a perturbation based on the signal. The architecture of this network consists of a three convolutional layers with *tanh* activation functions in between these layers. *Tanh* activation functions are used as opposed to ReLU used in previous work so that the resulting output signal can consist of both positive and negative values, as is necessary in RF communication. The AMN takes in a single-channel complex input of size [1, 1, 2, N] for a signal with N samples and 2 channels for IQ and outputs a signal of the same size and dimensions.

3.2.2 Loss Functions

The AMN utilizes custom loss functions during training in order to achieve its goal of deceiving the classification abilities of the eavesdropper while simultaneously limiting the BER at the receiver. This communications aware attack improves the ability of previous evasion attacks that only focused on misclassification. Due to the dueling nature of the transmitter's focuses of communication and evasion, multiple loss functions must be used that are then balanced. The loss functions used in this work are all updates to those used in prior work. All loss functions used are summed to create a total loss function to be back-propagated to update the AMN during training. The new total loss function is defined as.

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{adv}} + \beta\mathcal{L}_{\text{comm}} + \gamma\mathcal{L}_{\text{pwr}} \tag{3.3}$$

The current work uses three loss sub-functions: adversarial loss, communications loss, and power loss denoted as $\mathcal{L}_{\text{adv}}$, $\mathcal{L}_{\text{comm}}$, and $\mathcal{L}_{\text{pwr}}$ respectively. The adversarial loss seeks to minimize the eavesdropper's ability to successfully classify the signal, the communications loss looks to minimize the BER impact at the receiver, and the power loss seeks to minimize the power of the perturbation, thus keeping the adversarial signal similar to the scale of the original signal. These losses are summed together to provide an overall loss metric for the training process. In order to tune each of these loss terms, balancing constants are set for each of the losses. This helps set the desired trade-off between communication success and classification evasion for a given application. The three constants, α for $\mathcal{L}_{\text{adv}}$, β for $\mathcal{L}_{\text{comm}}$, and γ for $\mathcal{L}_{\text{pwr}}$, are assumed to sum to 1 and have a range of [0,1]. As α grows, the transmitter becomes more focused on evasion and the adversarial signal tends towards noise. As β increases, communication improves at the detriment of the evasion ability. As γ grows, the perturbation becomes more limited in power. Testing was done to ensure that each

loss function was relatively the same scale compared to each other without the weighting constants. This was done to ensure that one of the losses wasn't orders of magnitude larger (and would therefore drive the training process) or smaller (and therefore be irrelevant) when compared to the others.

Each of the separate loss functions are constructed so that they converge to 0 when achieving their desired effect, as is typical of other loss functions used to train DNNs. The optimization technique Adam, which utilizes gradients of the loss, is used during training which therefore requires the loss functions be differentiable [77].

Adversarial Loss

Adversarial loss looks to maximize the ability of the AMN to evade classification by the eavesdropper. In this sense, the intent of the adversarial loss metric mirrors that of a loss that would be used in more traditional evasion techniques such as FGSM. The metric used is the confidence the eavesdropper network has that the received signal is the original source modulation, determined by a softmax output. A decrease in this confidence can lead to a successful untargeted attack when the true class is no longer the one determined as most probable by the classifier.

In typical classification machine learning problems, cross-entropy is used as the loss function but this has the inverse behavior that is desired as it will approach 0 only as the confidence grows. Instead, the loss function used is rooted in log-likelihood and approaches 0 as the confidence decreases but tends toward ∞ as the confidence increases. The adversarial loss is defined as

$$\mathcal{L}_{\text{adv}} = -log(1 - p_s) \tag{3.4}$$

where p_s represents the confidence of the original source modulation scheme and is obtained

as the result of a softmax activation layer at the output of the eavesdropper's AMC network. This work only evaluates untargeted attacks but could easily be used to implement a targeted attack using $-log(p_t)$ where p_t is the confidence in the targeted modulation scheme. The loss presented here is updated from Eq. (2.6) to follow a more log-likelihood-based approach consistent with other areas in machine learning. This helps the classification accuracy stay minimal because this new loss will approach ∞ as the confidence in the true source modulation gets closer to 100%, forcing the AMN to learn to drive the classification accuracy down.

Communication Loss

The intent of the attack presented in this work is to carry out an attack that evades classification by a malicious eavesdropper while simultaneously allowing for effective communication between a transmitter and receiver. While the adversarial loss metric described above helps accomplish the former, it has a negative impact on the latter. As the AMN has increased success at fooling the eavesdropper, this typically means that the transmitted signal is very different from the original and therefore the communication reliability will degrade. The communication loss is therefore included to guide the AMN to find a better balance between adversarial success and communication success.

One way to quantify the reliability of the intended communication link is with BER. In the system model used in this work, the bits are retrieved by making a hard decision on the received symbols. Unfortunately, this process is not differentiable which means a gradient can't be calculated for the training process. The reason for that it is not differentiable is due to the hard decision process used in this work to determine received symbols. This is done by determining the symbol closest to the received symbol out of all possible symbols using an *argmin* determination. This *argmin* makes the hard decision process non-differentiable. If

the hard decision process was differentiable, this loss could simply be a mean squared error (MSE) between the original and received bits or could be defined by a gradient formula as shown in [74]. In order to circumvent this issue, the communication loss utilizes two components, BER and error vector magnitude (EVM) of the symbols and is defined as

$$\mathcal{L}_{\text{comm}} = b_r \times EVM(S_{tx}, S_{tx+p}) \tag{3.5}$$

where b_r is the BER calculated using the signal received over the noisy channel after the bits have been decoded with FEC. Since the encoding used in this work is done at the bit level instead of encoding the symbols, the error calculations are also performed on the bits rather than just on the symbols (the latter of which was done in the prior work). The EVM shown in the loss function represents the distance between the original symbols and the noiseless symbols after the perturbation is added. The EVM is calculated as $|S_{tx} - S_{tx+p}|$. The noiseless version of the symbols are used in the calculation rather than those that have undergone the AWGN channel so that the added randomness of the noise is not present in the EVM. The BER calculation, however, is made based on the noisy symbols since this is what is seen at the receiver. EVM is used in the loss because it is differentiable; thus the BER serves as a magnitude of the adjustment while the EVM serves as a direction for the weights to update. This is similar in design to how in FGSM, the sign of the gradient specifies the direction while the limiting factor, ϵ, determines the magnitude or severity of the update; however, in this framework, the magnitude changes as the BER changes whereas ϵ is held static regardless of communication success.

Power Loss

The third and final loss component used in this work is the power loss. The power loss aims to reduce the power of the perturbation compared to that of the original signal. The loss is calculated using an altered version of the signal-to-perturbation ratio (SPR), given as

$$\mathcal{L}_{\text{pwr}} = \frac{1}{E_s/E_p} = \frac{E_p}{E_s} \tag{3.6}$$

where E_p is the energy of the perturbation and E_s is the energy of the signal. The ratio is switched so that it decreases as the perturbation energy decreases, mirroring the behavior needed for the loss function. Additionally, the above equation is done on a linear scale rather than logarithmic so that the values are between 0 and ∞ rather than $-\infty$ and ∞. This allows the loss to converge to 0, providing numerical stability during training. This is an update from Eq. (2.8) and allows for more variability in power loss calculation than the previous maximum-based loss that instead attempted to impose a hard limit on the perturbation.

3.2.3 Training and Testing Process

The training process follows that depicted in Figure 3.3. The encoding and whitening of the randomly generated bits occurs before the modulating, sampling, and shaping processes and the final transmission process is the adversarial signal generation via the AMN. Before transmission, the adversarial signal is normalized so that the average symbol power is 1, enforcing a power budget. PyTorch is used to implement the AMN and training process. The training and testing is performed in a simulated environment, rather than OTA which is how the losses are able to be passed back to the AMN.

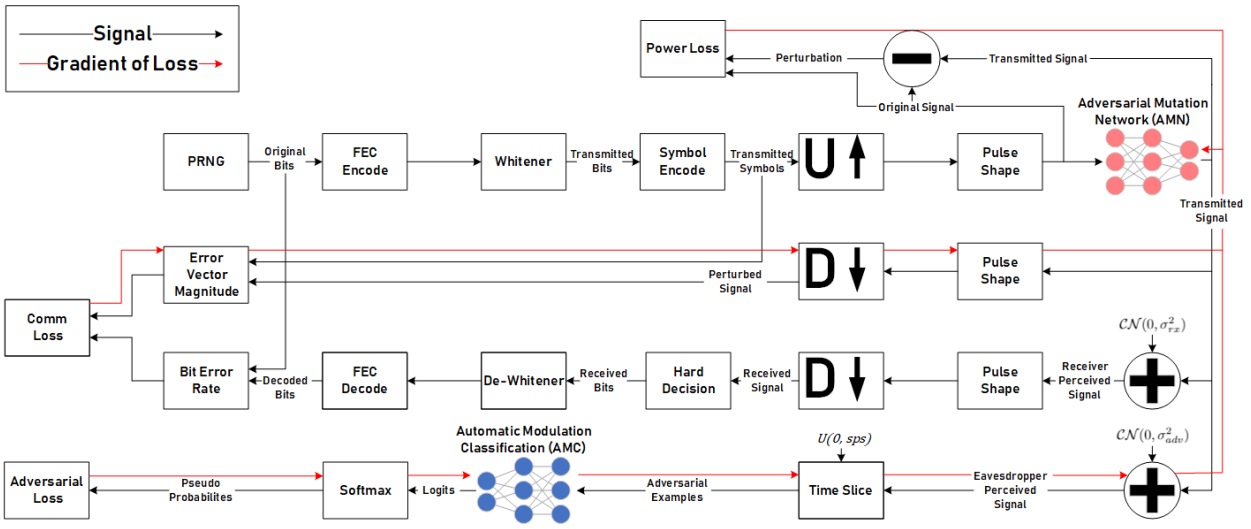


Figure 3.3: The communications aware attack framework training process. Three loss functions (power loss, communications loss, and adversarial loss) are utilized by the AMN during the training process to intelligently craft the signal perturbations for the given spectral environment. This training is performed in a simulated training environment.

3.3 Results using Implicit Approach

The communications aware attack framework just described was used to train a variety of AMNs. There are four main focuses for the results:

- to determine the difference in communication and evasion capabilities between AMNs trained with and without FEC in order to prove implicit learning of the FEC
- to study the effects of the adaptations made on the communication aware framework when compared to prior work
- to understand the impact of changing γ (power loss constant) and thus varying the power, and therefore impact, of the perturbation to understand the trade-offs involved in limiting the perturbation power
- to examine the success of the attack across different coding schemes, source modulations, and AMC networks to ensure generality of the implemented framework

In these results, the value of the power constant, γ was varied in order to study different perturbation powers. For ease of analysis, the α (adversarial loss constant) and β (communication loss constant) values were fixed at 70% and 30% of the remaining $1-\gamma$ respectively (since the three loss constants are set to sum to 1). These values were set such that evasion was prioritized adequately regardless of the value of γ. Otherwise, the transmitter would not successfully evade signal classification for larger γ as the value of α would be too insignificant. These exact values were determined based on trial and error.

3.3.1 Intelligent Perturbations with FEC

The addition of FEC inherently improves the intended communication link during an evasion attack given its inherent ability to correct bit errors; however, this work aims to show that the developed improvements to the communications aware attack improve the performance beyond the FEC's capabilities acting alone. In other words, there will certainly be improvement in the communication success due to the nature of FEC coding but this work will show that there is additional improvement based on the AMN's ability to learn the code improve the communication link as well. To demonstrate this, Figure 3.4 shows the results of both the framework considering FEC during training and the framework when FEC is taken out of the training process for a γ set to 0.1 (shown with the solid lines). The modulation scheme and FEC coding are QPSK and Hamming (7,4), respectively. As can be seen, there is improved intended communication performance given that the SNR differences between the BER curves for Hamming (7,4) and non-coded communication links are further apart than their respective theory curves in the code's operating region. For example, the SNR required to achieve a BER of 10^{-3} using an AMN trained with FEC has an improvement of roughly 1.5 - 2 dB over an AMN trained without.

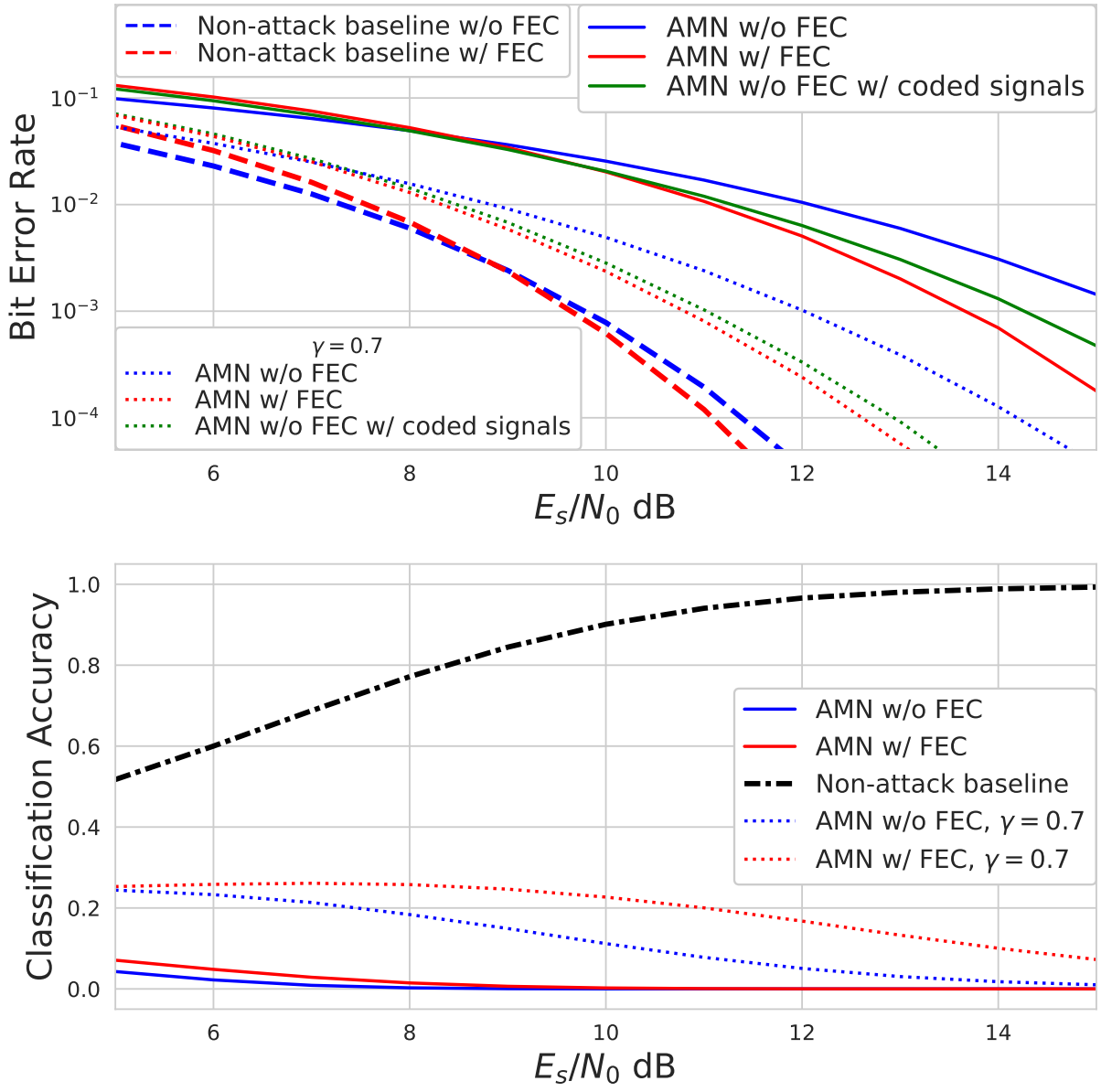


Figure 3.4: The performance shown is for Hamming (7,4) and γ values of 0.1 (solid lines) and 0.7 (dotted lines). This is the intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a QPSK modulated signal. The baseline theoretical curves of BER are shown given both FEC coding and non-coding as well as the baseline classification accuracy of the eavesdropper with no perturbation applied. In each case the addition of FEC in the training process improved communication with respect to BER with little to no degradation in the reduction of eavesdropper performance.

To further emphasize the improved framework's ability to inherently leverage the FEC code, this figure also shows the BER curve when the transmitted signal utilizes a Hamming (7,4) code but is perturbed using an AMN that was trained on signals without FEC. This approach is important because it allows for better understanding of the improvement that can be seen due to the inherent nature of FEC without the AMN having examples of it during training. If there is still improvement between this scenario and that of when the AMN is trained with FEC, then it can be seen that the AMN inherently learned to utilize the coding. In this case described, there is still a roughly 25% improvement in communication performance. These results show that *the AMN has inherently learned how to more intelligently craft the perturbation when FEC is present in the training process such that it limits the hit on communications performance, allowing for a better balance between evasion and communication success.*

While Figure 3.4 shows that there is a noticeable improvement in communications performance, for this case there is also an increase in the eavesdropper's classification success. However, this is a very small improvement compared to the communication improvement and both still result in misclassification. Additionally, the accuracy of the transmitters trained with and without coding is the same for the region of operation. This operation region exists in the SNR region after the intersection point of the coding and non-coding BER curves, after approximately 9 dB SNR. For SNR values less than this, the communication ability of the FEC-enabled signals is actually worse so it would not be practical to use in this region. Therefore an increase in eavesdropper accuracy for the FEC-enabled AMN is less important outside of the intended operating range than it is for higher SNR within the operating range where the adversarial success is more equal. However, in studying this effect further, Figure 3.5 demonstrates an observed case where the impact to communications between the AMN trained with FEC and that without is more equal, as indicated by the green and red lines

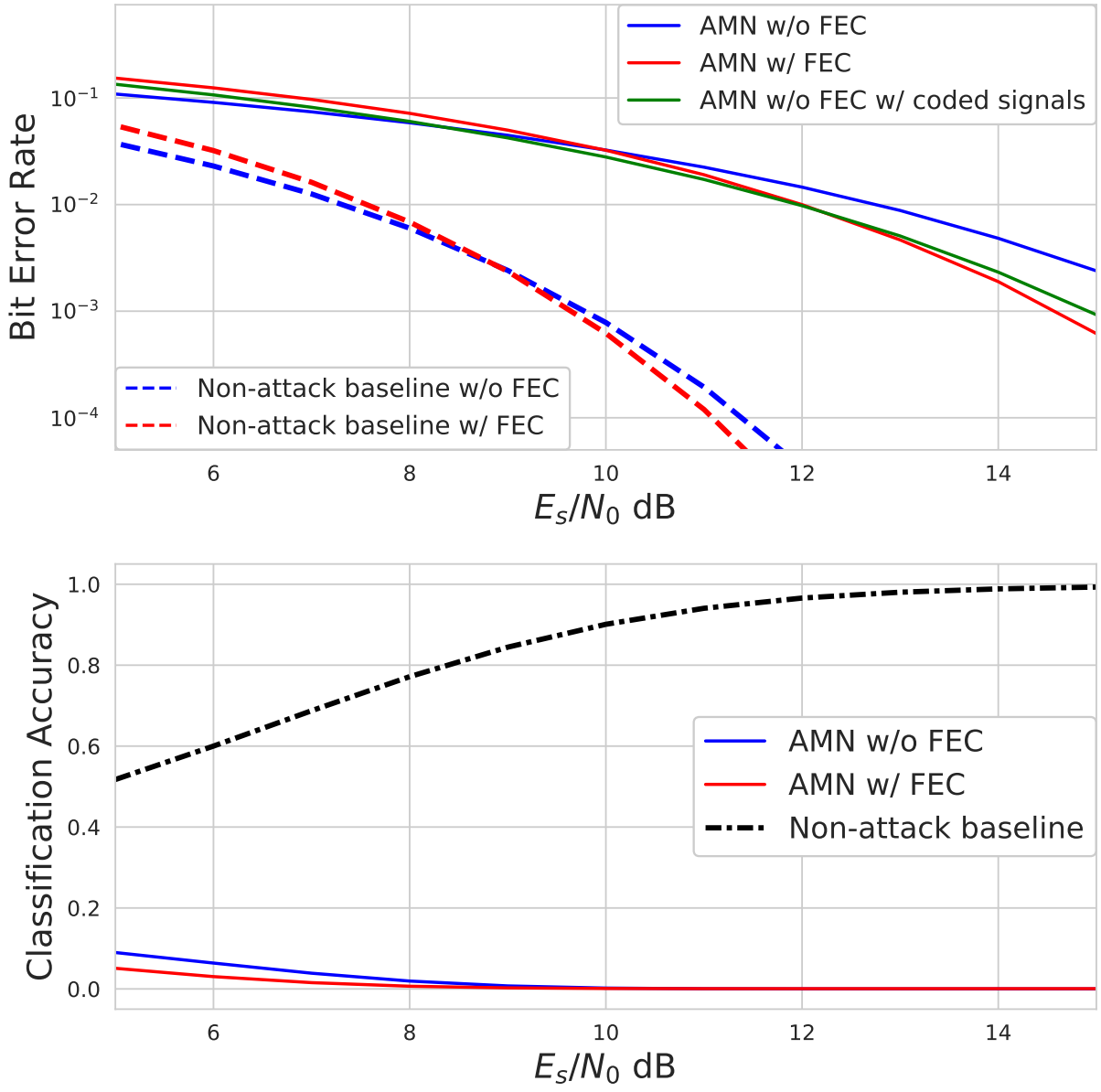


Figure 3.5: The performance shown is for Hamming (7,4) and a γ value of 0.1 when the communication impact between coding and non-coding is more equal. This is the intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a QPSK modulated signal. The baseline theoretical curves of BER are shown given both FEC coding and non-coding as well as the baseline classification accuracy of the eavesdropper with no perturbation applied. The evasion success is shown to be better when using coding given similar communication reliability.

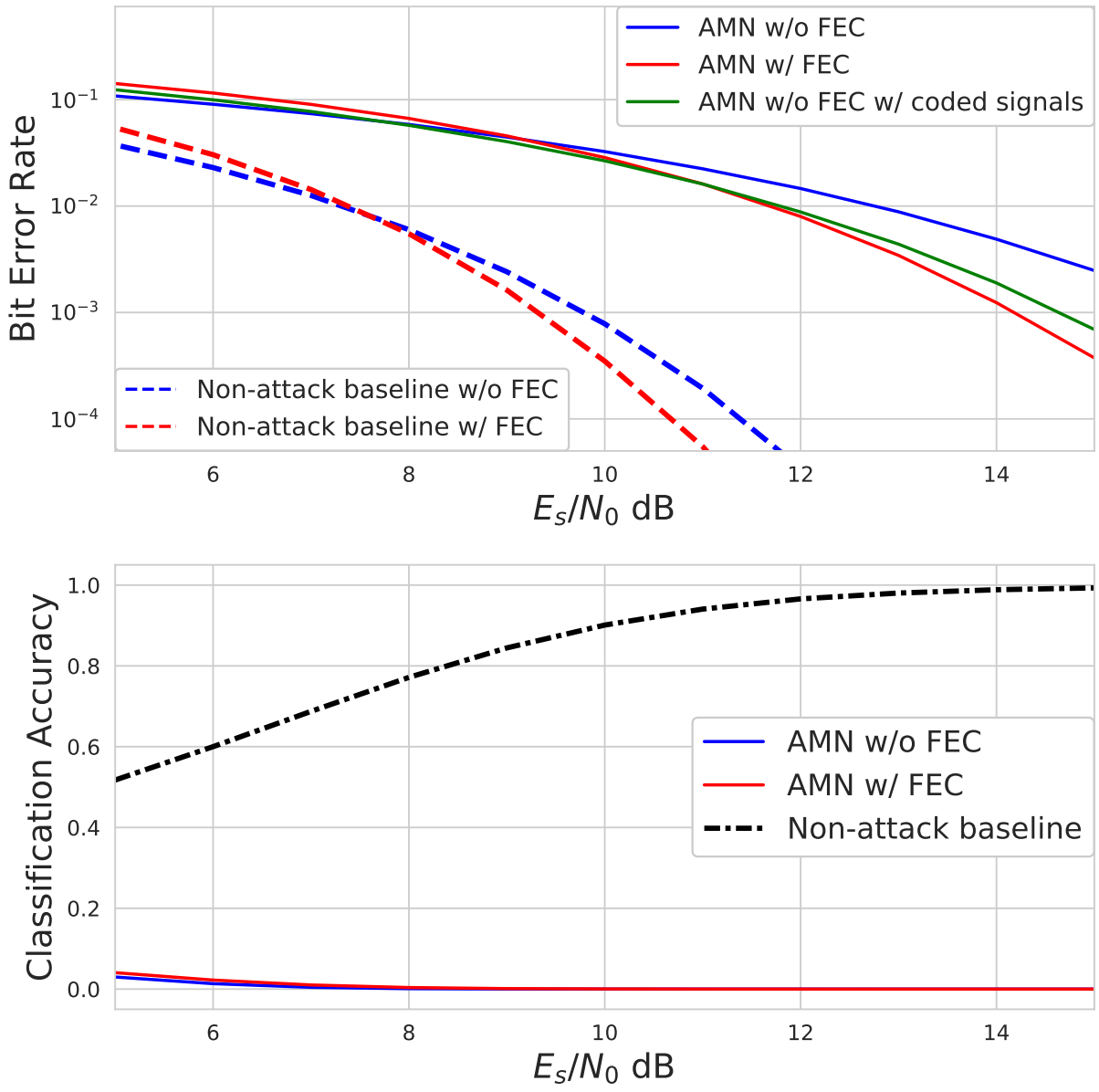


Figure 3.6: The performance shown is for Hamming (12,8) and a γ value of 0.1. This is the intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a QPSK modulated signal. The baseline theoretical curves of BER are shown given both FEC coding and non-coding as well as the baseline classification accuracy of the eavesdropper with no perturbation applied. This shows that the results generalize to other coding schemes even without changing the AMN architecture or training process.

being closer together. In other words, transmitting encoded signals using the AMN trained with FEC would offer the same BER as transmitting encoded signals using the AMN trained without FEC. When this occurs, the accuracy of the eavesdropper decreases when the AMN is trained with FEC. Therefore, if the desire is to improve the evasion performance, *the transmitter trained with FEC can decrease the success of the eavesdropper's AMC if maintaining the same level of communication reliability as a transmitter trained without FEC instead of improving the reliability.*

Figure 3.6 shows similar trends for a Hamming (12,8) applied to a QPSK signal. No changes were made to the training process, including architecture configuration, loss constants, and signal length, other than changing the scheme used to encode the signals. As can be seen by the plots, the same behavior and trends observed for Hamming (7,4) continues for the Hamming (12,8) scheme. The Hamming (12,8) performs better than Hamming (7,4) as the improvement in communication is still distinguishable but the eavesdropper's classification accuracy is nearly identical. Given these results, it is shown that *the AMN is able to learn to use the coding without any architecture changes* since it learns the FEC implicitly and doesn't rely on knowledge of the specific coding, assuming hard-decision block coding is used.

As discussed in Section 3.2, the γ value represents the weighting of the loss function that controls the power of the perturbation relative to the signal. To show the impact of γ, Figure 3.4 also shows the results when γ is increased from 0.1 to 0.7 (shown with dotted lines). This higher γ value means the training process prioritizes the goal of minimizing the perturbation power over that of evasion. This effect can be seen in the figure as improved BER performance at the cost of increased classification accuracy as expected. As seen in Figure 3.4, while there is still a communications improvement between the non-coded and FEC-enabled AMN implementations, the difference in eavesdropper success is much more pronounced. This shows the result that the FEC-enabled AMN offers much more clear

improvements for lower values of γ. In other words, the improvement when training using FEC is most significant when the evasion is prioritized (α is larger) since the FEC-enabled AMN is more efficient with the limited communication and power losses. This can also be understood as the FEC-enabled AMN offers more benefits when the perturbation is more powerful which makes intuitive sense due to the added error correction present in training and testing.

It was seen that the attack could be generalized across different FEC schemes but it is also important to ensure that this is the case across different modulations without updates to the framework. Figure 3.7 shows the attack carried out for 8-PSK with a γ value of 0.1 as this was the predominant γ used for QPSK testing. As the plots show, the attack provides an extremely drastic improvement in communication when training with FEC both over the non-coded attack and the attack that utilizes coded signals after training without FEC. When not training with FEC, the communication hit renders the attack borderline impractical but when utilizing FEC to train the AMN, there is about a 6 dB improvement that makes the BER much more reasonable. While the eavesdropper accuracy is higher for the attack that doesn't utilize FEC, the difference only occurs for the range of 5-9 dB and the evasion success of the attack with FEC is still very low, never getting above 10%. Figure 3.8 then shows the attack employed with 16-QAM. In this attack, there is communication improvement shown by the gap in BER being larger for the attack curves than for the theoretical curves. Additionally, the intersection point between the coding and non-coding BER curves is shifted about 2 dB lower, enlarging the operating range.

Figure 3.9 shows the BER of the intended communications link and the classification accuracy of the eavesdropper for a variety of γ values assuming a 16-QAM signal held at a constant SNR of 12 dB. 12 dB is used because it is in the middle of the range of SNRs where FEC offers improvement in BER. As can be observed, as γ increases, the perturbation power

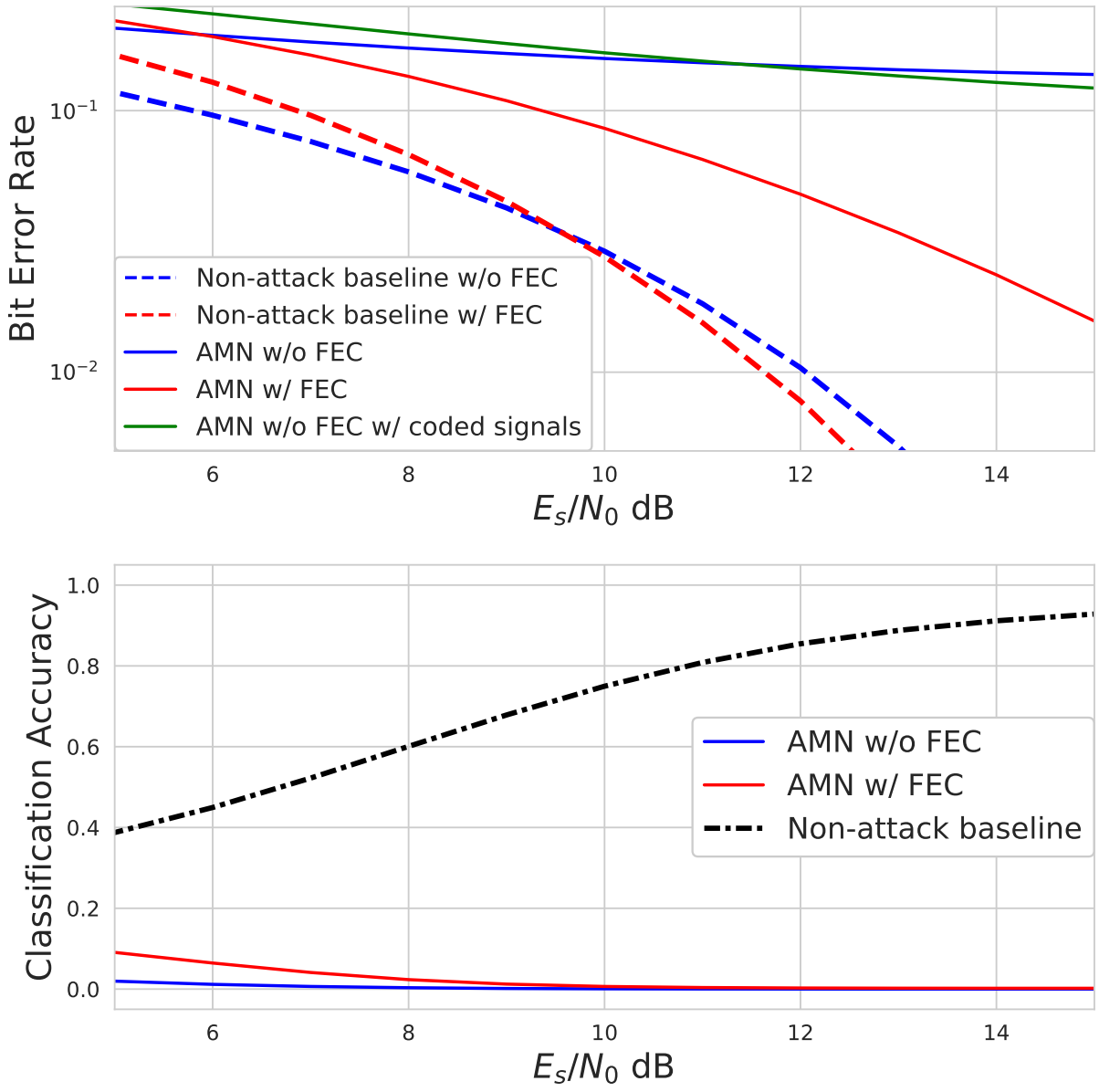


Figure 3.7: Intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a 8-PSK modulated signal. The baseline theoretical curves of BER are shown given both FEC coding and non-coding as well as the baseline classification accuracy of the eavesdropper with no perturbation applied. The performance shown is for Hamming (7,4) and a γ value of 0.1. This shows that the results generalize to other modulation schemes without changing the AMN architecture or training process. The communication success is substantively better for the coded approach.

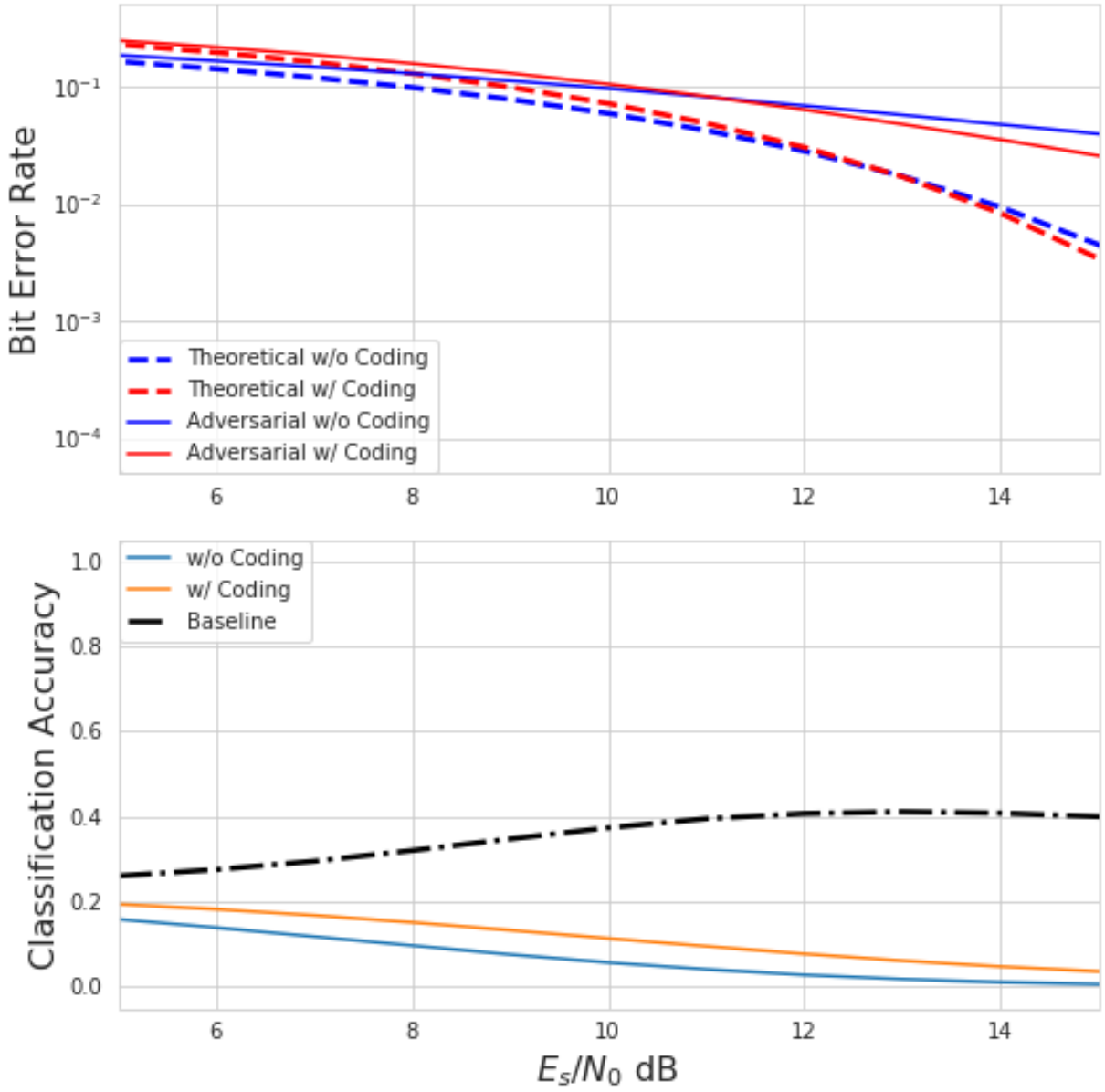


Figure 3.8: Intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a 16-QAM modulated signal. The baseline theoretical curves of BER are shown given both FEC coding and non-coding as well as the baseline classification accuracy of the eavesdropper with no perturbation applied. The performance shown is for Hamming (7,4) and a γ value of 0.7. This shows that the results generalize to other coding schemes even without changing the AMN architecture or training process.

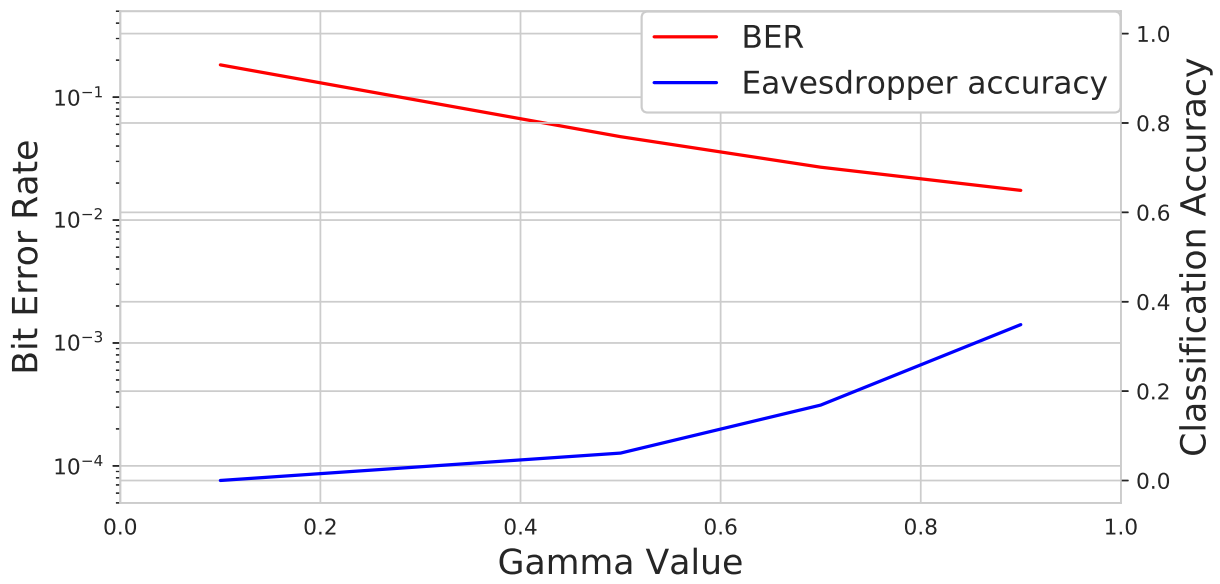


Figure 3.9: Intended communications link BER and eavesdropper classification accuracy given a transmitted 16-QAM signal with SNR=12dB for different weightings of the power loss function during the communications aware attack framework's training process (represented by γ).

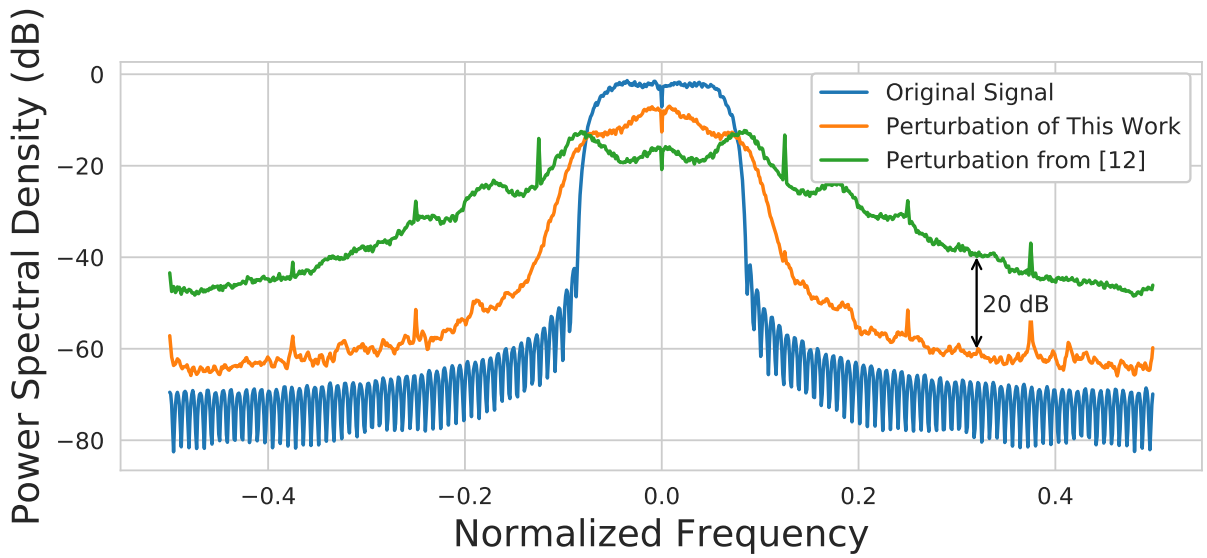
loss is more prioritized which leads to the perturbed signal being closer to the original signal. As this occurs, the BER decreases but the accuracy increases as would be expected, and as is seen across all modulation schemes and FEC codes tested, further ensuring generality across configurations. This shows that the behavior of the attack follows what would be expected as the loss constants vary, i.e. there is a trade-off between evasion and communication success. Further, in looking at the slope of the eavesdropper accuracy plot in Figure 3.9, it can be seen that the accuracy starts to drastically increase as γ goes up. This would indicate that smaller γ are more efficient for this attack, as was also determined based on the results from Figure 3.4.

The results of this work show that the FEC-enabled AMN is able to improve the trade-off between evasion and communication success. For a given level of communications success, the evasion can be improved with an AMN trained with FEC over one trained without it.

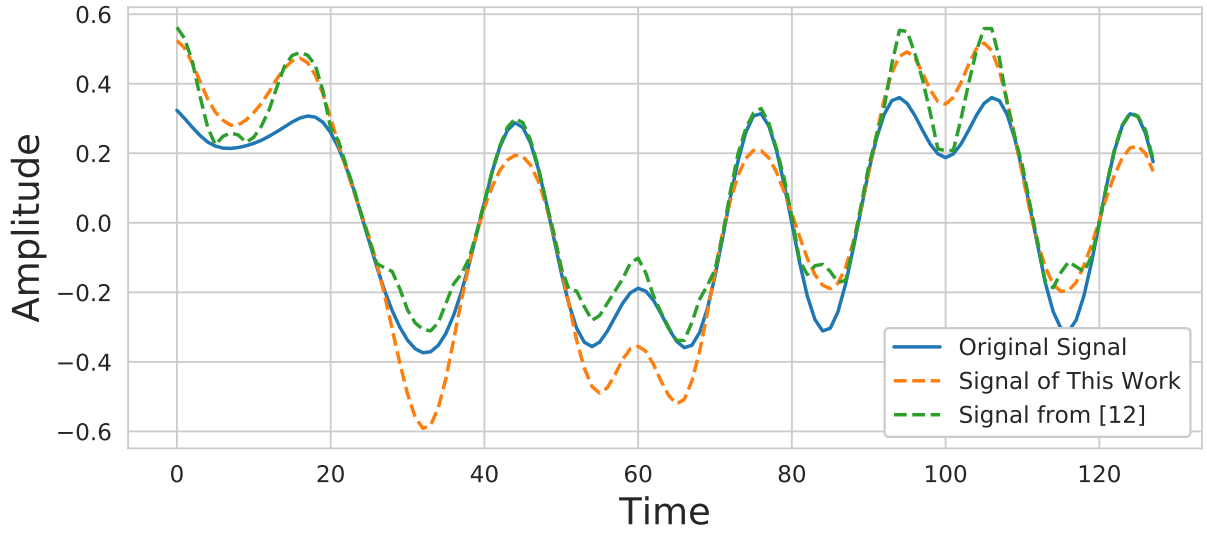
Additionally, the attack can be generalized to other modulation and FEC schemes without needing to change or update the architecture of the AMN.

3.3.2 Spectral Improvement

The communications aware attack framework developed in the previous work inherently tended to spread the perturbation out-of-band of the transmitted signal to reduce impact on the communications link, given the assumption of oversampling at the eavesdropper (an intuitive result indicating correct functionality) [12]. Figure 3.10a shows the spectrum usage of the signal and perturbation for both the developed and previous framework in [12]. As can be seen, this work improves on the spectral efficiency of the perturbation by moving more of the perturbations in-band and reducing out-of-band emissions. While there is still perturbation outside the main band, it is less significant in power (by about 20 dB) and more closely follows the frequency structure of the original signal. This is important because the eavesdropper could attempt to remove the perturbation using a preprocessing such as low-pass filter. The signal crafted using this work would be more robust to this defense compared to that of previous work since it is more powerful in-band. Additionally, this is important because it allows the signal to better stay within an allotted frequency band. These improvements are due to the AMN and loss constant (predominantly the power loss) updates of this work since the AMN now has to learn to craft a legitimate signal and is penalized by the power loss if it doesn't. In the previous work, only a perturbation was crafted and there was no incentive to have the combined perturbation and original signal appear benign. Figure 3.10b shows a plot of the time domain for the original signal, the adversarial signal generated using [12], and the adversarial signal generated with this work. The signal created with the framework developed in this work more resembles the original signal in structure. The adversarial signal in prior work is less smooth due to its



(a) Power Spectral Density



(b) Time-Domain Representation

Figure 3.10: The (a) spectral shape and (b) time-domain representation of a transmitted QPSK signal with and without perturbation. The improved communications aware framework developed in this work reduces the out-of-band effects caused by the perturbation over the prior work.

high frequency content, especially in the minima. Additionally, the signal power given this approach is equal to the original signal but the prior work generates a signal with an 8-10% increase in power. Therefore, *this work presents an attack that is both more spectral efficient and power efficient than the prior work.*

3.3.3 Transfer Learning

The results up until this point have assumed that the AMN has knowledge of the eavesdropper's specific architecture and trained weights. This is used in training to determine the adversarial loss using the output of the AMC network and the weights are used to allow for backpropogation to the AMN. While this assumption of complete knowledge is tolerable for understanding if the communications aware attack works in a best case scenario, this is not practical in real world scenarios as this knowledge of a malicious entity is unlikely. Therefore, this attack is tried against 3 additional forms of the eavesdropper, each with different assumptions made about the similarities between the AMC used in training and that seen in the actual execution of the communications aware attack. The first assumes an AMC with the same architecture and training data but trained independently from the training AMC. The second assumes the same architecture but different training datasets. However, while the dataset is different, it still assumes similar data, i.e. the same modulation schemes, SNR range, etc. The third and final AMC is one that has a different dataset and a different architecture. Specifically, a third convolutional layer is added to the AMC. This method of carrying out the adversarial machine learning attack against eavesdroppers that the attack wasn't trained on is also referred to as transfer learning and is important in understanding the applicability of the attack to real world use cases.

This transfer learning study is carried out for two different attacks to understand how well

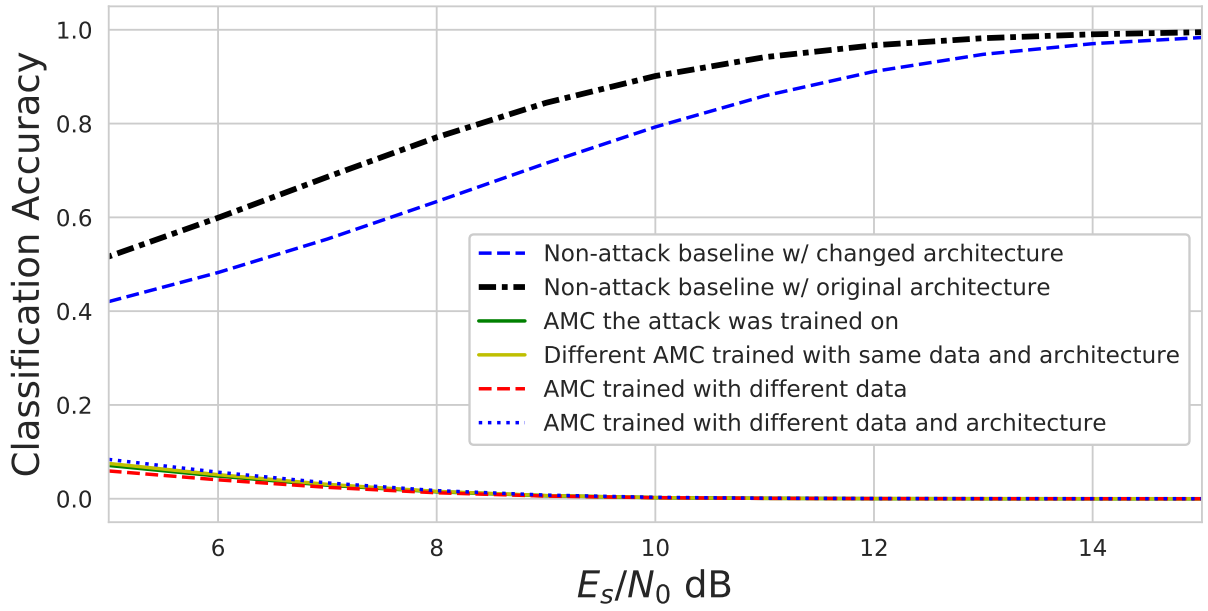


Figure 3.11: The eavesdropper's classification accuracy for an FEC-enabled attack with a γ value of 0.1. Shown is the evasion success for the original training eavesdropper, one with the same dataset and architecture, one with the same architecture but different dataset, and one with both a different architecture and dataset. Additionally the un-attacked classification accuracy is shown. The attack is still successful for all eavesdroppers even though they were not the ones used in training, as the resulting classification accuracy is within about 1% of the original.

the attack can generalize to other AMC networks. Specifically, the attacks correspond to the two QPSK, FEC-enabled attacks shown in Figure 3.4. One involves a γ value of 0.1 and the other a γ value of 0.7. This is done to see how well the attack transfers when the perturbation is more powerful ($\gamma = 0.1$) and when it is less powerful ($\gamma = 0.7$). The evasion success for all three cases are compared to the original evasion success when the attack assumes perfect knowledge of the eavesdropper.

Figure 3.11 shows the results of the transfer learning attack for an attack trained with a γ of 0.1, meaning the perturbation power less restricted. Two baseline accuracy curves are shown, showing the classification accuracy when the AMC is fed normal, unperturbed signals. As expected, all of the eavesdroppers that had the same architecture had essentially

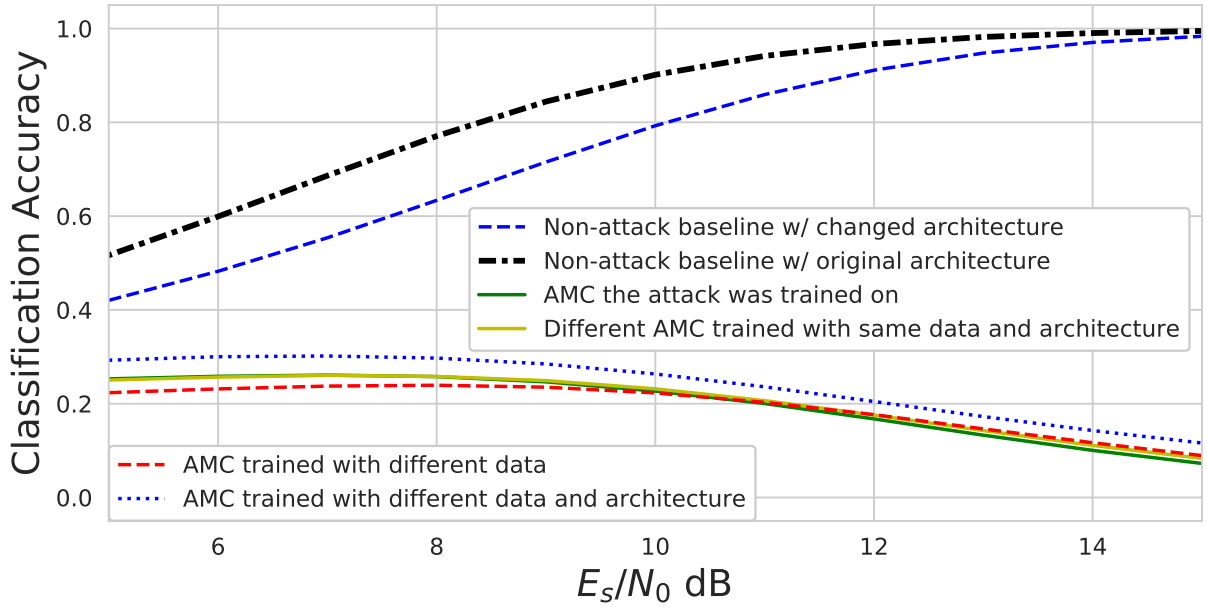


Figure 3.12: The eavesdropper's classification accuracy for an FEC-enabled attack with a γ value of 0.7. Shown is the evasion success for the original training eavesdropper, one with the same dataset and architecture, one with the same architecture but different dataset, and one with both a different architecture and dataset. Additionally the un-attacked classification accuracy is shown. The attack is still successful for all eavesdroppers even though they were not the ones used in training. The eavesdropper that uses a different architecture had slightly higher classification accuracy when attacked but was still successful in significantly lowering the accuracy of classification.

the same baseline accuracy even if trained with a different dataset. The eavesdropper that incorporated an additional convolutional layer had different accuracy that was actually worse for the considered scenario. Figure 3.11 also shows evasion success for all four eavesdroppers that the attack was tested on. The classification accuracy is the same for all four, plus or minus about 1%.

Figure 3.12 shows the resulting eavesdropper classification accuracy for the four eavesdroppers when tested with an attack trained with a γ of 0.7, where the perturbation power is more restricted. One again, the attack still results in low classification accuracy for all four eavesdroppers, even though the attack was only trained on one of them. For this case, it is seen that the eavesdropper that was trained with different data and employed a different architecture classified the signals about 3-4% better than the other three eavesdroppers, even though its baseline accuracy was worse. This is due to the fact that this eavesdropper is the most different from the eavesdropper used in training. However, even given this slight improvement, the attack is still successful in significantly decreasing the classification accuracy.

The results of Figure 3.11 and 3.12 show that *the attack can be successfully transferred to other eavesdroppers the attack was not trained on* and that the AMN only needs to be supplied with a generally similar eavesdropper. While there may be some reduction to the evasion success when the attack is carried out on a different eavesdropper, as shown in Figure 3.12 for the eavesdropper employing a different architecture, it is not a large reduction and the attack is still results in significantly lowered classification accuracy. This is an important takeaway for ensuring the communications aware attack can still be carried out in scenarios where the AMN doesn't have complete access to the eavesdropper during training, something true in many real world applications. Intuitively, as the architecture of an eavesdropper diverges more from that used in training, the performance of the attack decreases.

3.3.4 Convolutional Coding

The results shown in this chapter have considered block codes, specifically Hamming, as the category of FEC employed by the communication system. While this offers insight into the impact that FEC has on the training and success of communications aware attacks, other coding schemes such as convolutional codes are much more prevalent in modern communications systems. Block codes were used as a baseline for quantifying the benefits of utilizing FEC in the attack and how the AMN is able to learn to make use of the coding. These results and trends should transfer to other coding schemes, such as convolutional codes, just as it did between different Hamming codes as shown in Figure 3.6. This subsection addresses this concept more directly.

For these results, the same architecture, data, and training process as the previous results of this chapter were employed. The only difference is that the FEC used is a convolutional code with a rate of 1/2 and a constraint length of 9. This means there are 2 encoded bits for every 1 decoded bit and the shift register used to determine the output is 9 bits long. Two different configurations were tried, one with a γ of 0.1, and one with a γ of 0.7, the same as the configuration utilized in Figure 3.4.

Figures 3.13 and 3.14 show the results for γ values of 0.1 and 0.7, respectively. As can be seen in both figures, the BER of the coded signals is much better than the signals that don't utilize FEC given that convolutional codes have much stronger correction capabilities than the Hamming codes previously studied in this chapter. Similar to the behavior seen when studying the effects of training with Hamming codes in Figure 3.4, the BER is lower for the adversarial encoded signals crafted with an AMN trained with FEC than for encoded signals perturbed with an AMN that was not exposed to encoded signals during training. This is seen for both a γ of 0.1 and a γ of 0.7 and indicates that the AMN is able to learn to improve

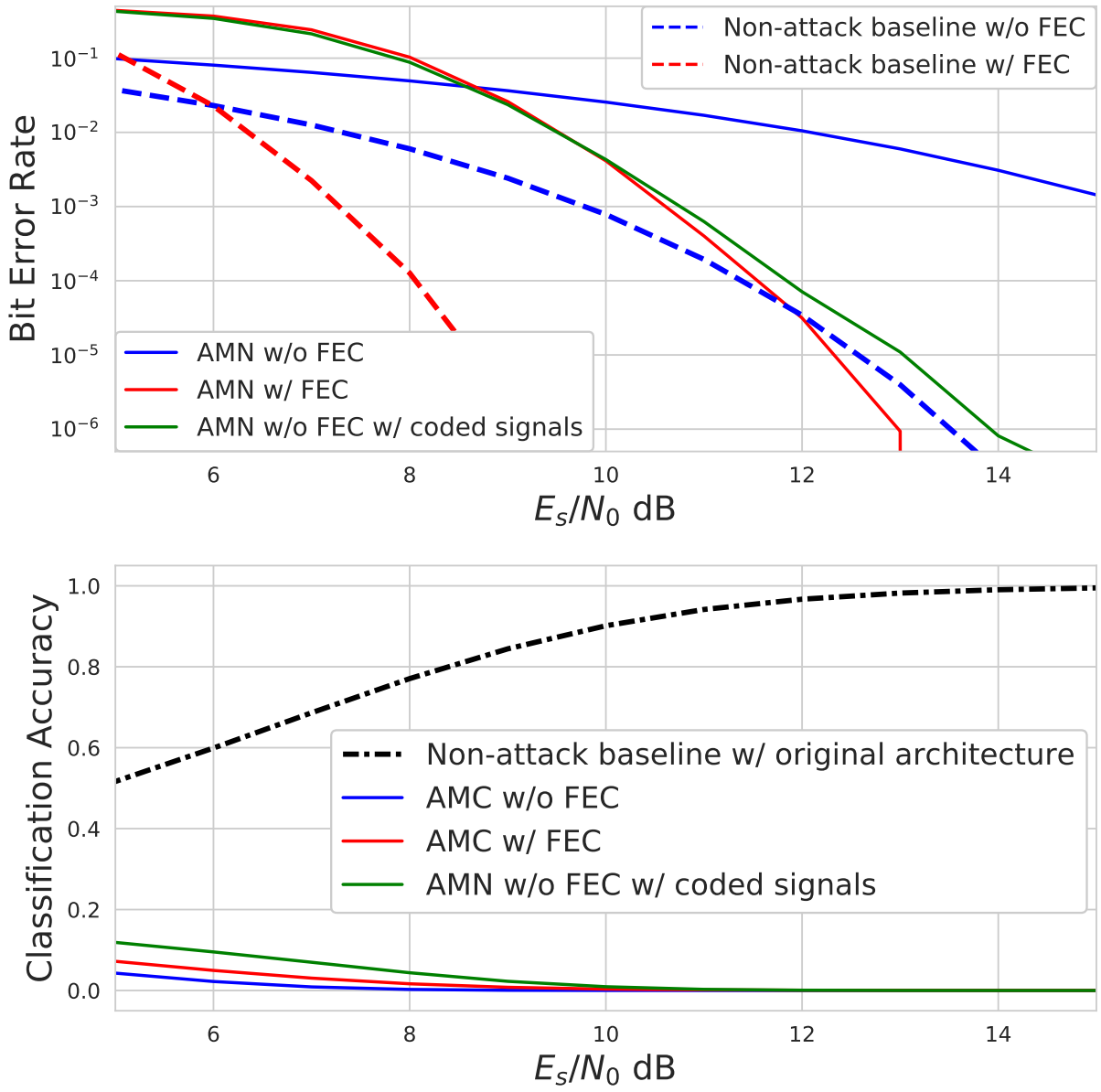


Figure 3.13: The performance shown is for convolutional coding with a rate of 1/2 and constraint length of 9. The attack configuration uses a γ of 0.1. This is the intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a QPSK modulated signal. The baseline theoretical curves of BER are shown given both FEC coding and non-coding as well as the baseline classification accuracy of the eavesdropper with no perturbation applied. The addition of FEC in the training process improved communication with respect to BER and improved the evasion success with respect to the classification accuracy.

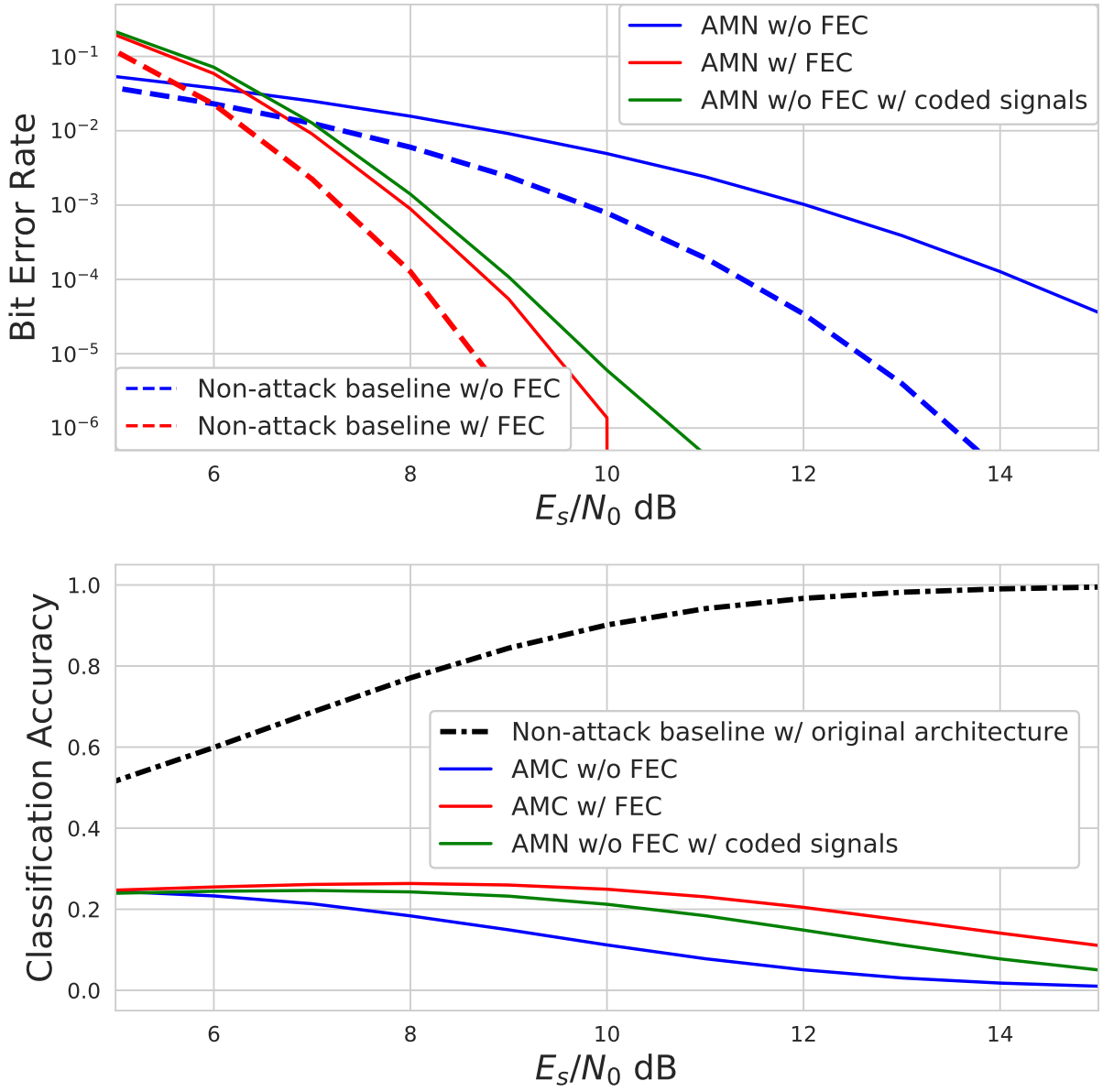


Figure 3.14: The performance shown is for convolutional coding with a rate of 1/2 and constraint length of 9. The attack configuration uses a γ of 0.7. This is the intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a QPSK modulated signal. The baseline theoretical curves of BER are shown given both FEC coding and non-coding as well as the baseline classification accuracy of the eavesdropper with no perturbation applied. The addition of FEC in the training process improved communication with respect to BER with little degradation in the reduction of eavesdropper performance.

communication, through lowering BER, when exposed to FEC during training. In addition, when the RF signals utilize FEC but are perturbed with an AMN that wasn't trained with FEC, the classification accuracy becomes higher than the FEC signals perturbed with an AMN trained with FEC, when γ is 0.1. This means that when employing convolutional coding in a communications system, both the evasion and communication success can be improved when trained with coding. In general, *when an AMN is trained with convolutional coding, it is able to better limit the trade-off between the communication and evasion success.*

These results are for a more implicit approach, as was done with the other results in this chapter so far, where no characteristics of the coding scheme employed is used to dictate the architecture or training process of the AMN. Future work could consider a more explicit approach that specifies the AMN architecture based off the constraint length.

3.4 Explicit Knowledge

Up until this point, the results and framework have made the assumption that no explicit information about either the existence of coding or specific information about the coding scheme in use was provided to the AMN or used during architecture design. The intended result was to show that the AMN would be able to utilize the coding implicitly based on solely the generic architecture and learning process, as was then shown to be true in Section 3.3. While this is beneficial because it allows for easy changes in coding scheme and modulation without needing to reconfigure the architecture, something that may be needed for real world applications that are resource constrained, it may limit the ability for the AMN to utilize the FEC coding to the fullest potential. Intuitively, it would make sense that if provided with knowledge of the coding strategy in use, and configured properly to account for this, the evasion attack described in this work should become even more successful.

3.4.1 Background

There are a few different ways that the training process could supply the AMN with explicit knowledge of the coding scheme. One would be to do so utilizing the loss functions. The loss functions could be re-implemented such that they reward behavior in the perturbation that utilizes the code such as placing a high perturbation on one specific symbol within a coding block. However, this may prove to be too complicated of an approach since the loss functions would still need to be differentiable. Being differential and providing explicit FEC coding information are two difficult aspects that would need to be combined and simultaneously satisfied. This leads to the second method, the one studied in this work, that looks to change the architecture of the CNN used by the AMN as a way to explicitly leverage the FEC code. One way to do this is with the striding and kernel size of the convolutional layers.

As was mentioned previously, this work predominantly utilizes block codes as the form of FEC. This means that blocks in the signal are coded independently and have no inter-block relation. Given this structure, if the kernel size (number of samples included in a convolution) and stride length (how many samples the kernel moves between convolutions) are set to be equal to the coding block size, then the convolutions would follow along the natural block structure of the coded signal. In the previous implementation, the configuration of the convolutional layers, namely the stride and kernel size, was not adjusted to match the FEC code blocks so the coded signal and convolution were not synchronized and therefore the AMN was not optimized to learn to utilize the coding. This implicit approach is illustrated in Figure 3.15 where the stride length is 1 and the kernel size is not equal to that of the FEC block size. In contrast, the proposed explicit approach is illustrated in Figure 3.16 where the stride length and kernel size are equal to the FEC block size. The stride length and kernel size illustrated in these figures are meant to serve as an example of possible values and the general approach, they do not reflect the actual values used.

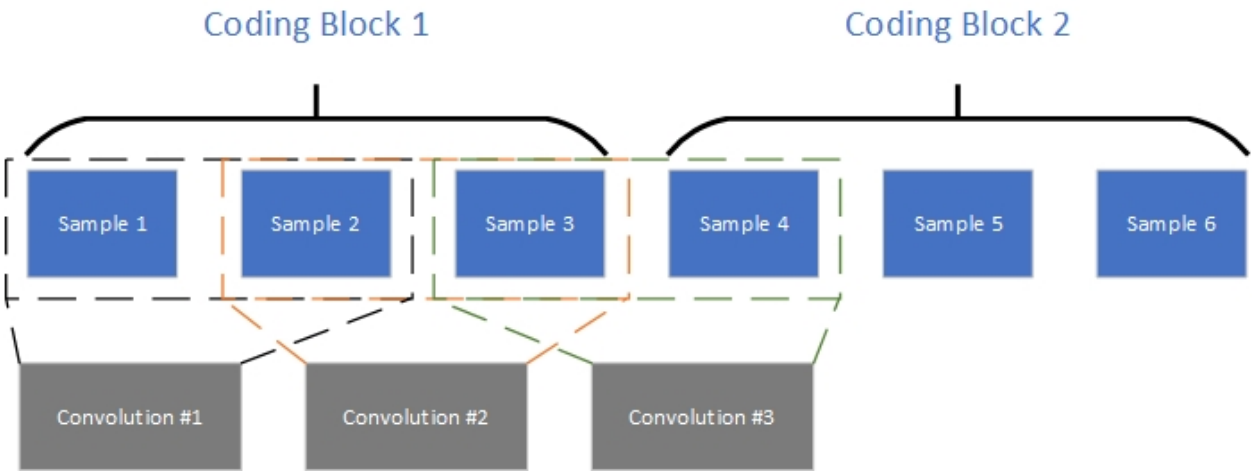


Figure 3.15: The implicit approach used in Section 3.3 and prior work. In this approach the stride length is 1 and the kernel size is unrelated to the FEC block size, meaning no information of the coding scheme is explicitly provided to the AMN. The kernel size is 2 and the block size is 3. These values serve as examples so that the process can be seen visually, the true values were a stride of 1 and a kernel size of 7.

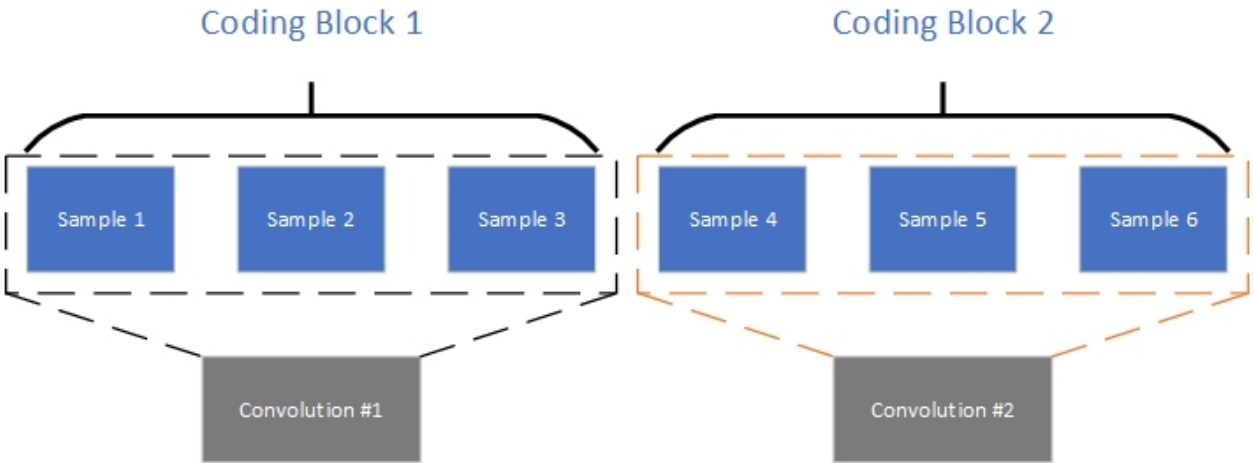


Figure 3.16: The explicit approach proposed in this section. In this approach the stride length and kernel size are equal to the FEC block code size, 3 in this scenario. This means the AMN should be better suited to learn characteristics of the actual FEC scheme is use. These values serve as examples so that the process can be seen visually, the true stride and kernel size were larger.

Given this new implementation, with specifically tailored striding and kernel sizes, the configuration of the CNN employed by the AMN will need to be changed not only for different FEC schemes but also for each different modulation due to differences in the number of bits per symbol for different modulations because this changes the effective block size. The FEC block size is indicated in bits but the convolutional layer operates over samples so the number of bits need to be converted to the number of corresponding samples. When changing the modulation scheme, the number of bits per symbol (and therefore sample) changes, creating a different effective block size and requiring a change in the stride and kernel size of the convolutional layer.

3.4.2 Implementation

This work on utilizing explicit knowledge is intended as a preliminary observation into the potential implementations that can be leveraged as well as the success seen with such approaches. Due to this, the approach of updating the kernel size and stride length to match the coding block size was implemented for just one modulation scheme and FEC block code in order to see initial results. A more general study is left as future work. The configuration studied is QPSK and Hamming (12,8) for the modulation and FEC code respectively. For this given implementation, there is a limitation on the types of coding schemes that can be used. The coding block size, in bits, must be divisible by the bits per symbol (BPS) of the modulation scheme. Otherwise some of the resulting symbols and therefore samples during the modulation process would be made of bits from two different FEC coding blocks. Such a scenario would make the approach more like the implicit approach discussed earlier in the chapter. For example, Hamming (7,4) would only work for BPSK (BPS of 1). Hamming (12,8), on the other hand, is feasible for BPSK, QPSK (BPS of 2), 8-PSK (BPS of 3), 16-QAM (BPS of 4), and 64-QAM (BPS of 6) since 12 is divisible by all BPS.

Additionally, the transmitter processing stage needs to be updated for this approach. Previously, the bits were mapped to symbols, upsampled, shaped by an RRC filter, and then perturbed with the AMN. However, due to the filter transients added by the RRC, this process is not transferable to this explicit information approach. This is because the filter transients add an odd number of samples to the signal and therefore the striding no longer lines up with the coded blocks of the signal. Instead, the perturbation will be added between the upsampling and the RRC filter. The signal input to the AMN is reshaped such that the coding blocks get separated by being in different dimensions. The signal shape becomes *Batch* x *Channel* x *IQ* x *Block size* x *Time*. This is different than the previous shape which was *Batch* x *Channel* x *IQ* x *Time*. An extra convolutional layer is added to the CNN (as the first layer) in order to help learn the coding scheme. This layer is implemented as a 3D convolution where the dimensions convolved over are the block size, the IQ components, and the time domain, and is added because of the increase in signal dimensions. In the previous setup there were three convolutional layers, one that convolved over the IQ and time dimensions and reduced the IQ dimension to a size of 1 and two additional layers that convolved over the resulting, essentially one dimensional, signal (besides the added size in the channel dimension). In the new setup an additional layer was added so this same approach could be used: one layer that reduces the IQ dimension to a size of 1, one that reduces the block size dimension to a size of 1, and an additional two layers to operate on the remaining one dimensional signal. The extra layer (added as the first layer) is the sole layer that will utilize the updated kernel and stride because the concept of FEC block size is most apparent before the dimensions of the signal are rearranged by the convolutions. The value for each of these for the given QPSK and Hamming (12,8) is 48 due to the samples per symbol of 8 and QPSK mapping 12 bits to 6 symbols.

3.4.3 Results

The success of this explicit knowledge approach was tested on an architecture optimized for QPSK signals and Hamming (12,8). A γ value of 0.5 was used along with α and β equal to 70% and 30% of the remaining $1 - \gamma$ respectively, in continuation of what was done for the implicit implementation. This results in a α value of 0.35 and a β value of 0.15. A γ of 0.5 was used as this was a good median value of configurations previously tested. These constant values were also used because the success of the FEC-enabled attack for higher γ values for the implicit approach was seen to result in a worse trade-off between evasion success and communication ability. Therefore, seeing improvements in higher values of γ, such as at 0.5, would be beneficial to the practicality of the approach. In order to show improvement over an implicit approach, the results from this configuration are compared to the same architecture trained with Hamming (7,4) QPSK signals. Since the architecture in this case is still optimized for Hamming (12,8), training it with Hamming (7,4) is essentially an implicit approach where the architecture configuration doesn't reflect the coding scheme and is not synchronized with it. If the AMN trained with Hamming (12,8) performs better than that trained with Hamming (7,4), then it shows that the AMN learned to utilize the coding structure explicitly. Additionally, the results of the two FEC coding schemes using this updated architecture are compared to the same coding schemes trained with the previous implicit-based architecture. This is done to ensure that improvements are due to the AMN's ability to explicitly learn an FEC code (Hamming (12,8) in this case) and not any underlying benefits that one code may have over the other.

Figure 3.17 first shows the comparison between Hamming (7,4) and Hamming (12,8) when the implicit approach from the previous sections in used (the AMN has striding and kernel sizes unrelated to the coding block size). The difference in BER between the two attacks is roughly equal to the gap between their theoretical curves, indicating the AMN learned

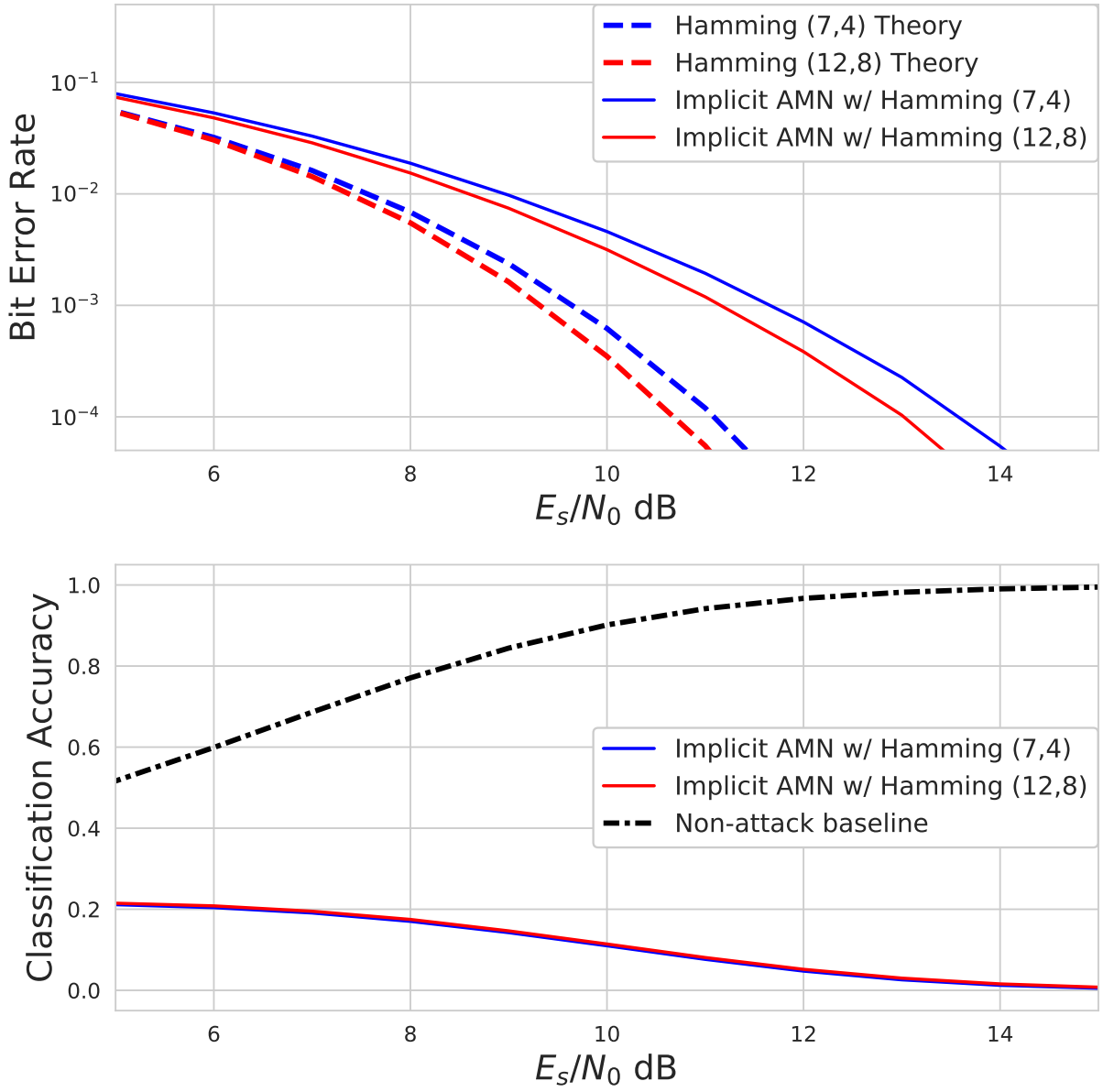


Figure 3.17: Intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a QPSK modulated signal. A γ of 0.5, α of 0.35, and β of 0.15 are used. A comparison between an implementation with Hamming (12,8) and one with Hamming (7,4) is shown using the implicit approach discussed earlier in the chapter. The resulting BER for both attacks remain essentially constant when compared to the theoretical curves and the classification accuracy is the same. This means that the AMN learns to utilize both code schemes equally well when employing the previous implicit approach.

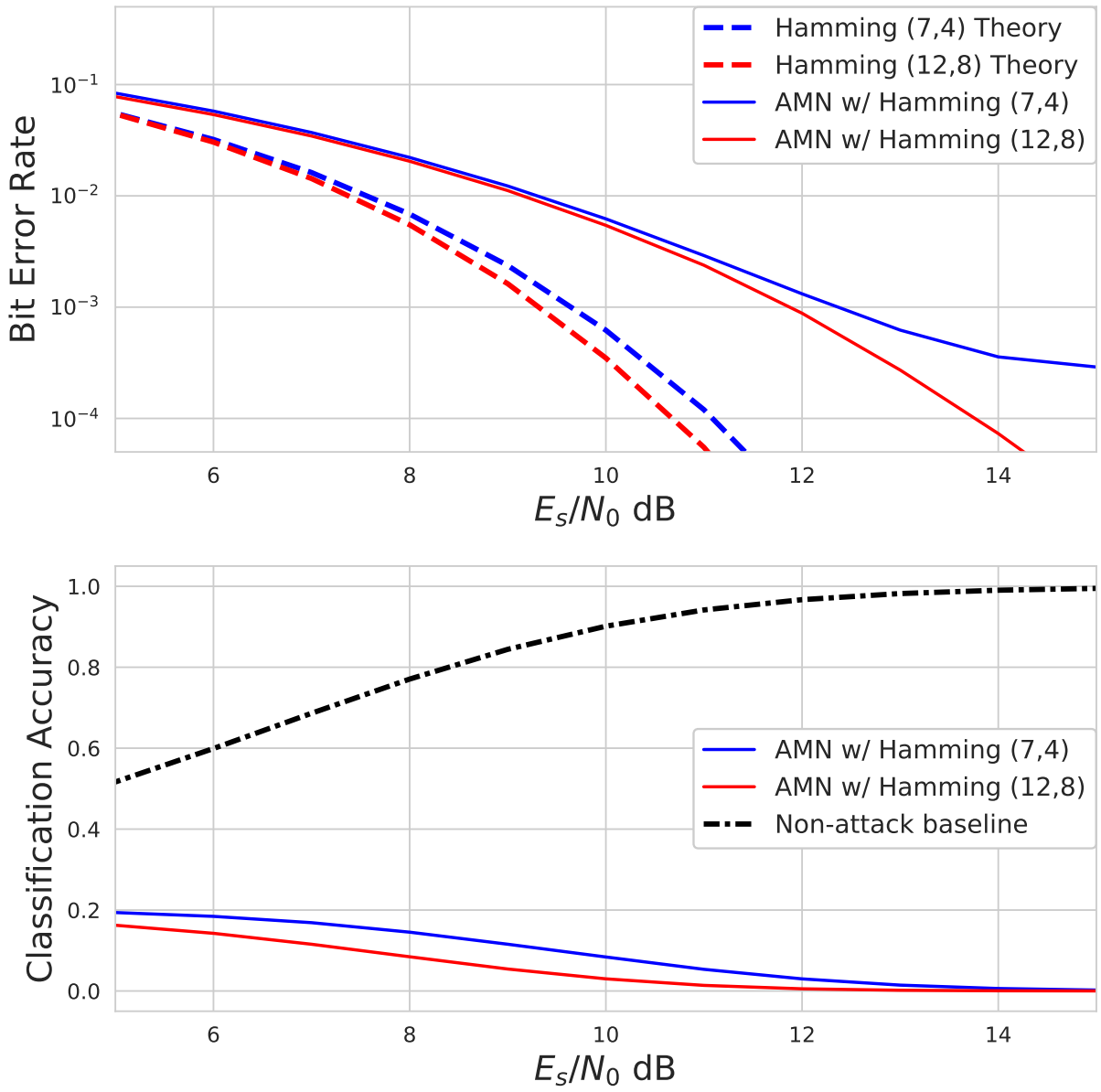


Figure 3.18: Intended communications link BER and eavesdropper classification accuracy for different values of SNR (E_s/N_0) given a QPSK modulated signal. A γ of 0.5, α of 0.35, and β of 0.15 are used. A comparison between an implementation with Hamming (12,8) and one with Hamming (7,4) is shown where both utilize AMNs optimized for Hamming (12,8). The attack utilizing Hamming (12,8) encoded signals observes better success, especially in evasion ability.

to communicate equally effectively between the two schemes. Additionally, the resulting classification accuracy for the two is the same. This would be expected given the implicit approach should allow the AMN to learn to utilize any coding scheme and do so equally.

Figure 3.18 shows the results for both Hamming (12,8) and Hamming (7,4) when using an architecture that has been optimized to explicitly learn Hamming (12,8) coding by updating the stride and kernel size. As can be seen from the theoretical curves, there is a slight inherent BER improvement for Hamming (12,8) over Hamming (7,4). However, the BER gain between the attack implementing Hamming (12,8) over that with Hamming (7,4) is equal to or greater than the gain in the theoretical curves as was seen in the implicit approach of Figure 3.17. Even given the slight improvement, the most noticeable and telling result is that the evasion success is improved for Hamming (12,8). There is a 5-10% improvement in evasion success for the SNR range tested and the success hits essentially 0% roughly 2 dB before the Hamming (7,4) attack. When comparing the two coding schemes while employing an AMN where the stride and kernel size don't match the block code (implicit), the success is equal. However when comparing the two schemes while employing an AMN configured for Hamming (12,8) (explicit), the success of Hamming (12,8) is better, especially for evasion. These results show *the AMN is well suited to learn the coding scheme when provided explicit information of the structure through the architectural configuration, leading to improved attack success.*

While there is promise in these results, this is still just a preliminary study. As can be seen from Figures 3.17 and 3.18, the communication became worse with the updated architecture. This may be due to a decreased number of channels in the added convolutional layer which results in decreased dimensionality for the AMN to adequately learn to communicate as well as with the previous implicit approach. A study into the best architectural configuration should be done to help improve this and determine the most suitable number of channels,

layers, and placement of the layer with adjusted stride and kernel size (such as putting this layer as the last layer rather than the first). However, this does not change the observed result that the AMN can utilize specific striding and kernel size to better learn the employed coding scheme in order to make more intelligent perturbations.

3.5 Conclusion

This work has shown that the communications aware attack framework, trained with signals utilizing FEC, can inherently learn to leverage the added data redundancy to generate more intelligent perturbations that have less impact on the intended communication link while not impacting evasion performance against an eavesdropper. To achieve this, modifications to the framework were developed that allow for improved feedback through the training loss functions that more directly represent the impact of FEC on the intended communications link. Performance analysis shows that for the operating region of the FEC code, the improved framework developed in this work was able to better evade the eavesdropper for a given intended communications link bit-error rate over a system not utilizing FEC. The results of this work demonstrate that the improved performance is not just due to the inherent benefits of using FEC on the communications link, but also due to the framework intelligently learning to manipulate the transmitted signal based upon the capabilities of the given FEC code. Further, the improvements seen when utilizing FEC carry across other FEC block codes and modulation schemes without changing the attack framework implementation. Additionally, this framework was shown to be successful for convolutional coding, indicating it does not apply solely to block codes. It was also proved that the success of the attack could be transferred to eavesdroppers other than the one used in training the AMN including eavesdroppers employing AMC networks with different architecture. This

means the attack can still be successful even when the AMN doesn't have access to the true eavesdropper.

In addition to the enhanced attack performance, the improved framework developed in this work provides for perturbed signals that better hold their original spectral shape than what was seen in the prior work [12]. A limitation of this and prior work was the assumption that the eavesdropper oversampled the received signal naturally allowing for out-of-band perturbations effects. This improved framework therefore allows for both relaxed eavesdropper assumptions and more efficient bandwidth utilization of the perturbed transmitted signal.

Furthering this investigation into FEC adaptation of communications aware attacks, it was found that the kernel size and stride length could be configured in a way that helps provide explicit information of the coding during training. When doing this, improvements were seen in the evasion success of the attack when compared to that of the implicit approach.

Ultimately, the results presented in this chapter show that DNNs can learn to enhance adversarial attacks using aspects of the signal processing chain already inherent in most RF communication systems. Using this, previous work in the RF machine learning field could be improved without adding any overhead to what would normally be found in a communication system.

More work should be done to optimize the explicit approach. It is possible that other configurations of the convolutions may work even better than what was done in this work. The results presented in Section 3.4 were preliminary and could be improved especially with respect to the communication success. For instance more channels could be added to the convolutional layers to allow for increased dimensionality, giving the attack network more flexibility to learn. Further, targeted attacks should be examined to determine their success within this framework. Results from the MI-FGSM approach discussed in Chapter 2

showed extremely large perturbations were required to successfully execute targeted attacks. The results and improvements offered in this chapter should make targeting attacks more reasonable.

Additionally, while the work presented in this chapter provided improvements to the spectral characteristics of the attack more work could be done to further improve the spectral characteristics of the perturbed signal through the incorporation of loss metrics that aim to keep the transmitted signal within its predefined spectral mask. This would have the added benefit of improved performance against eavesdroppers with intelligent filtering processes aimed at removing out-of-band perturbations. This is the focus of the next chapter. Another target for future work includes incorporation of knowledge of the channel propagation effects between the intended receiver and/or eavesdropper, as was done in [41] for more complex operating environments.

Chapter 4

Spectral Integrity

In communications aware attacks thus far, the idea of successful perception of the signal at the receiver is driven using metrics such as bit error rate (BER) that indicate the success of the communication. This chapter presents a new novel form of perception to be considered alongside BER, that of spectral integrity. The previous works in this area have shown that adversarial perturbations naturally tend to manifest out of the main lobe of the signal and lead to adversarial signals that do not hold well the same spectral shape as the original signal [12]. This change in the spectral shape of the signal poses a problem to the success of the attack, as the eavesdropper could leverage preprocessing stages to reduce the impact of the perturbation, such as with a filter, and the shape could potentially lead to increased likelihood of detection that an attack is taking place since the spectral shape does not appear benign. Additionally, typical communication links are assigned channels within a specific spectral mask. The out-of-band spectral content could pose an issue with this constraint, making the transmission non-compliant.

This chapter introduces both a new loss metric for training communications aware machine learning-based adversarial evasion attacks and an altered attack method that both separately help maintain spectral integrity of the adversarially perturbed signal while still successfully achieving evasion and intended communications. Section 4.1 of this paper first provides a background on the necessity of this study. Section 4.2 offers a solution by laying out a process that perturbs symbols rather than samples. Section 4.3 then introduces candidate

spectral integrity loss metrics and provides relevant performance analysis for attacks performed instead on the samples. Finally, Section 4.5 concludes this work and briefly discusses future work based on these findings.

4.1 Background

As has been shown in previous work [12], Chapter 3, and more specifically in Figure 3.10a, the perturbation creation process in a communications aware attack tends to create a combined adversarial signal that exhibits significant out-of-band frequency content. This can pose a problem to the usability of the attack for a variety of reasons. First, an eavesdropper could utilize various RF preprocessing strategies, like a matched filter, to remove this out-of-band content, leaving only the perturbation that lies in the main lobe. Given how much of the perturbation lies outside the main lobe, this could potentially render the attack ineffective. Given the ease at which an eavesdropper could employ such a preprocessing strategy, it is something that needs to be addressed and the communications attack updated to minimize or eliminate this limitation.

An additional problem that could arise from the irregular spectral shape and behavior of the adversarial signals seen in previous work is that this could be easily detectable by either a human operator or machine learning system deployed to flag when attacks are occurring. It would be fairly trivial to notice a difference between the perturbations and the original signal shown in Figure 3.10a. Depending on the scenario in which the communications aware attack is deployed, being detected might be just as detrimental as the decrease in evasion success. For this reason, it is desirable that the final adversarial signal appear, both in shape and structure, to be benign.

Finally, given the large number of devices and protocols operating in the frequency spectrum,

especially with the recent release of 5G, spectrum allocation is becoming extremely vital. Often, communication links are allotted specific frequency bands to stay within. By utilizing out-of-band perturbations, the resulting adversarial signals may violate these spectral masks. In a commercial environment where these assignments must be followed, this may force the communication link to have to go offline. Additionally, it could cause interference with communications in neighboring channels.

Given that the goals of going undetected and complying with the spectral mask should in theory go hand-in-hand with the desire for the attack to be robust against filtering (more benign frequency content would mean more of the perturbation in the main lobe and less outside), these are addressed in tandem in this chapter. The focus of this chapter is therefore to increase the spectral integrity of the attack to achieve these goals and decrease the potential limitations.

Throughout this chapter, the methodology, attack environment, eavesdropper architecture, training process, etc. are the same as that introduced in Chapter 3 with the exception that forward error correction is not implemented. This is done for the sake of simplicity so that the results are solely reliant on the updates made for increased spectral integrity and not on the benefits introduced with FEC. Additionally, the loss constants (α, β, and γ) are set in this chapter to better test and address the trade-offs between the losses. The values for these constants used in the results of this chapter represent only some of those tested and are presented to best represent the general trends and behavior.

4.2 Perturbing the Symbols

One approach that can be leveraged in order to eliminate out-of-band content is to perturb the modulated symbols of the communication stream. In the previous implementations of

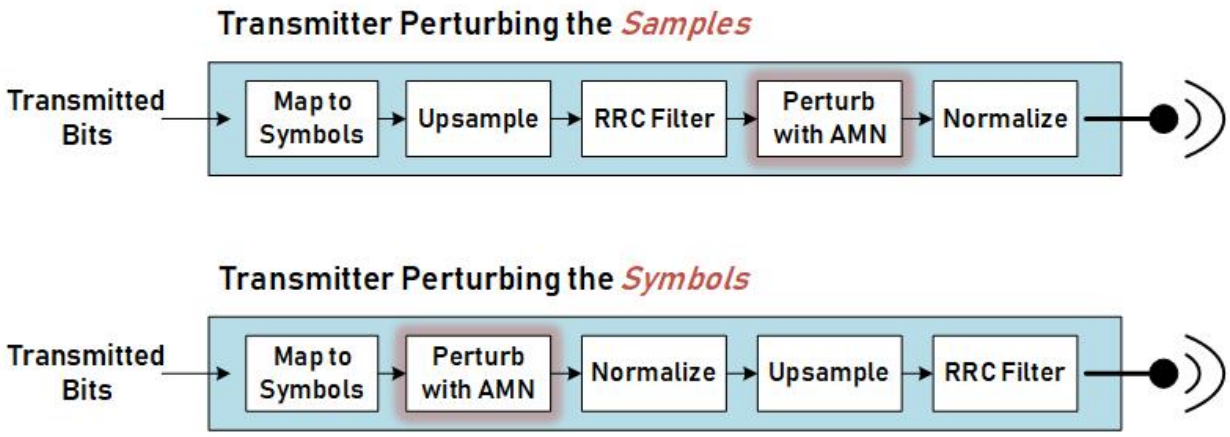


Figure 4.1: The process for crafting the adversarial signal at the transmitter. The upper diagram shows the transmitter used in previous work such as Chapter 3 where the perturbation is done on the final samples. The lower diagram shows the transmitter considered in Section 4.2 where the perturbation is performed on the symbols before interpolation.

the attack the perturbation was instead computed and added to the samples. By doing so on the symbols instead, the AMN in not able to utilize the extended bandwidth created by oversampling the symbols and thus the perturbation will be restricted to the main lobe. This approach is only valid for linear modulation schemes, other modulations would need to utilize the approach developed in Section 4.3.

In order to compute and add the perturbation on the symbols, the AMN must be inserted earlier in the transmission process. Rather than using this as the final step, as was done previously, the perturbation is determined and added to the signal after the bits are mapped to symbols for the given source modulation. After the perturbation is added, the adversarial symbols are upsampled and filtered with the RRC filter to create the final adversarial signal. The transmitter utilized in for this approach, as well as the one used previously, is depicted in Figure 4.1. This is a novel approach that hasn't been implemented in previous iterations of the communications aware attack, or other RFML evasion attacks (such as those in Chapter 2) in general. The remaining aspects of the training process and AMN, such as the loss functions, remain unchanged.

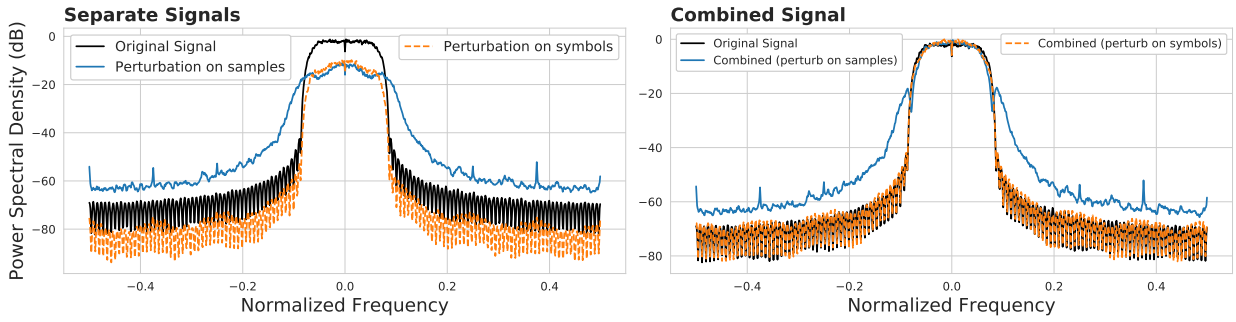


Figure 4.2: The signals shown were created with AMNs of $\alpha = 0.3$, $\beta = 0.4$, and $\gamma = 0.3$ using QPSK modulation. This is the power spectral density (PSD) of the signals generated when either perturbing the samples or the symbols. The method that perturbs the symbols results in a perturbation and adversarial signal that minimally impacts the spectrum.

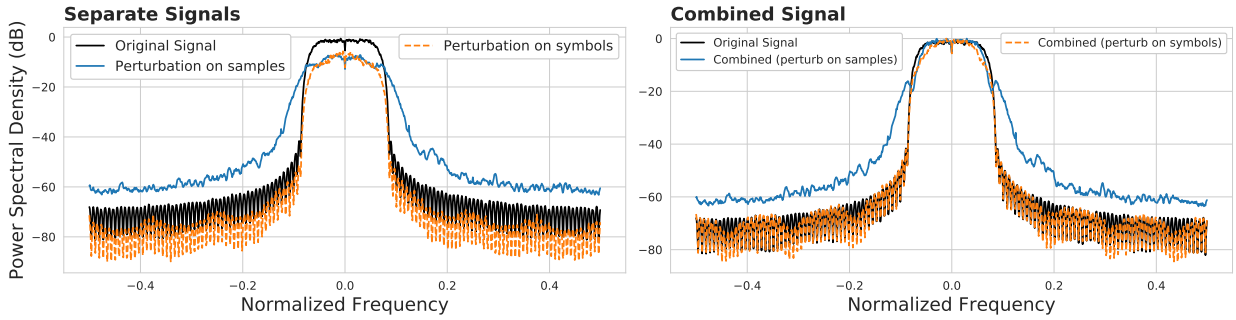


Figure 4.3: The signals shown were created with AMNs of $\alpha = 0.6$, $\beta = 0.3$, and $\gamma = 0.1$ using QPSK modulation. This is the power spectral density (PSD) of the signals generated when either perturbing the samples or the symbols. The method that perturbs the symbols results in a perturbation and adversarial signal that minimally impacts the spectrum.

In order to determine the success of this symbols-based implementation, it is compared to the original samples-based implementation using the same configuration across both. The only difference being where the AMN is placed in the transmission process. Two different loss constant configurations are considered, one that prioritizes communication and another that considers evasion more heavily. Figure 4.2 shows the spectral characteristics of each attack method for the first configuration of $\alpha = 0.3$, $\beta = 0.4$, and $\gamma = 0.3$. As would be expected intuitively, the resulting power spectral density (PSD) plot shows that when perturbing the symbols, the perturbation and resulting adversarial signal have minimal impacts on the spectrum. This is in contrast to the method where the samples are perturbed in which there is side content. This continues to hold true for the second loss constant configuration of $\alpha = 0.6$, $\beta = 0.3$, and $\gamma = 0.1$, shown in Figure 4.3.

Next it is important to consider the effects that this spectral improvement has on the BER and evasion success. Figures 4.4 and 4.5 show these results for the configuration prioritizing communication and that prioritizing evasion, respectively. For both scenarios, while there is a slight performance loss for both BER and evasion success when perturbing the symbols rather than the samples, it is not very substantial. Because of this, it can be seen that *implementing the communications aware attack that perturbs symbols rather than samples, is successful given that there is strong improvement in spectral integrity with very minimal detriment to BER and evasion.*

While these results are extremely promising, it comes with a large assumption about the attack environment. In order to place the perturbation on the symbols, it must be assumed that the AMN has access to the intermediate stages of the transmission process in order to place the perturbation process between the symbol mapping and upsampling. While this is practical in many applications, this may be infeasible in scenarios where the RF processing stage is a legacy system in which nothing can be changed within and all that is seen is the

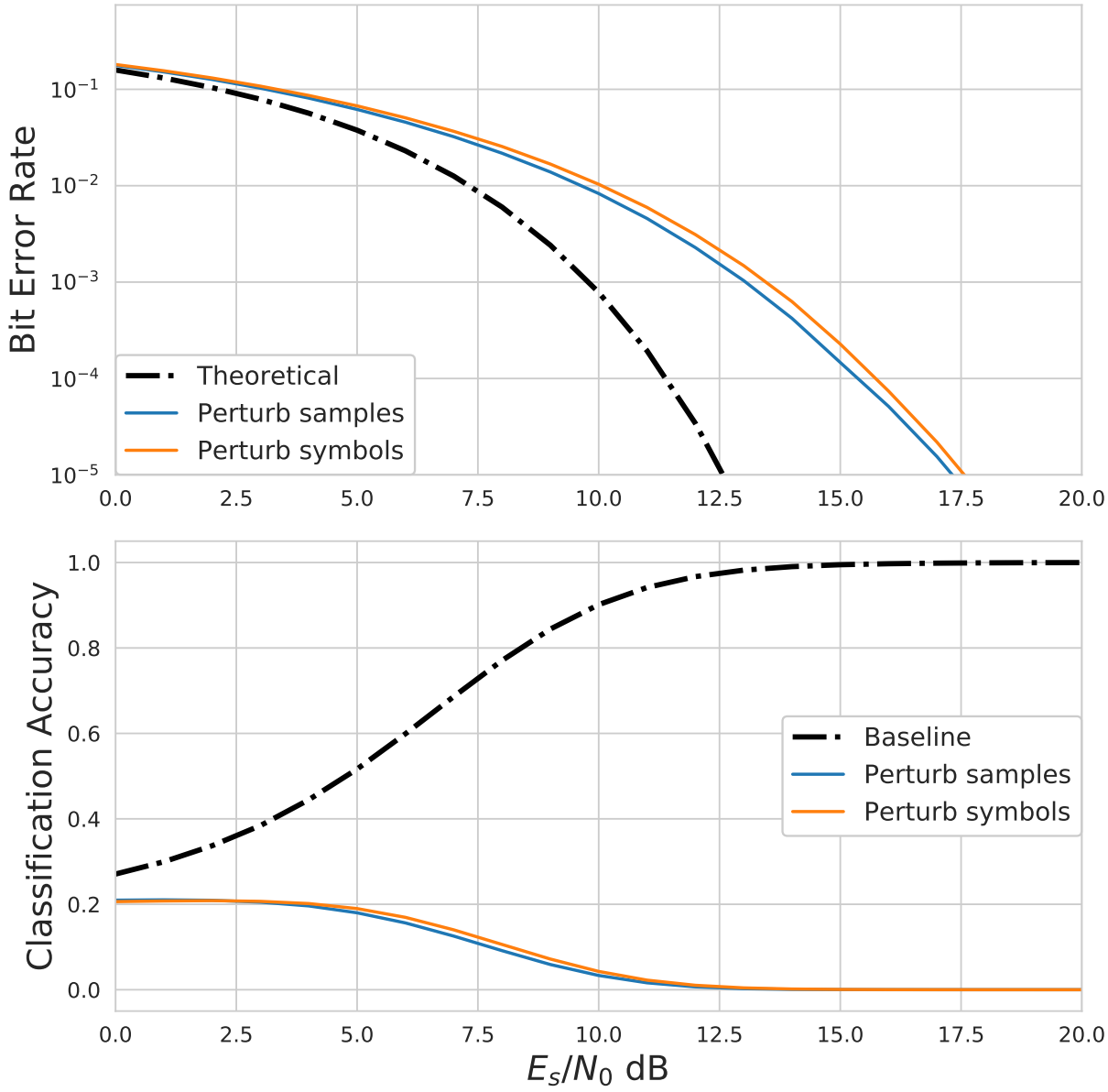


Figure 4.4: The signals shown were created with AMNs of $\alpha = 0.3$, $\beta = 0.4$, and $\gamma = 0.3$ using QPSK modulation. This is the BER and eavesdropper classification accuracy for both an attack that perturbs the symbols and one that perturbs the samples. While there is slightly worse performance in both metrics for the attack carried out on the symbols, it is not by much. Therefore even the improved spectral integrity does not come at the expense of significant decreases in BER or evasion success.

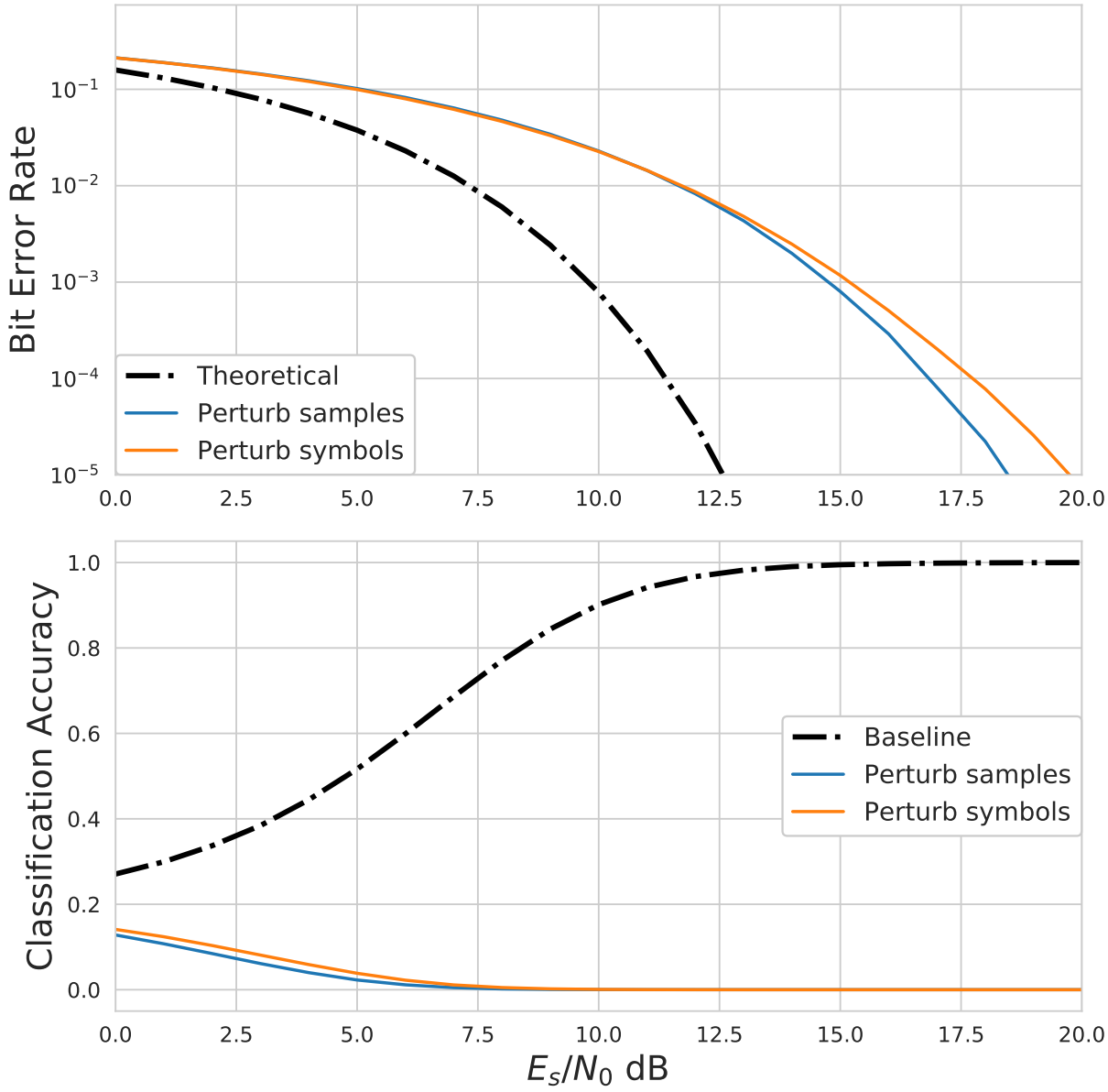


Figure 4.5: The signals shown were created with AMNs of $\alpha = 0.6$, $\beta = 0.3$, and $\gamma = 0.1$ using QPSK modulation. This is the BER and eavesdropper classification accuracy for both an attack that perturbs the symbols and one that perturbs the samples. While there is slightly worse performance in both metrics for the attack carried out on the symbols, it is not by much. Therefore even the improved spectral integrity does not come at the expense of significant decreases in BER or evasion success.

inputted bits and the outputted signal. The attack implementation that operates on the samples can be plugged in at the end of the transmission whereas the attack operating on the symbols has to be inserted within the transmission process, potentially infeasible if it's a legacy system. Additionally, in a cover attack scenario, described in [7], where the AMN is a separate entity, not co-located with the transmitter, the AMN needs to craft a perturbation that can be combined with the clean, fully processed signal consisting of samples, not the symbols. This is a realistic example of treating the RF transmission process like a block box where the AMN would not have access to the symbols. In such a scenario, the symbols-based method detailed in this section would not be feasible, however the implementation introduced in Chapter 3 would be, given that it operates on the samples and acts as the last step in the transmission chain. In order to provide stronger spectral integrity, the focus of this chapter, updates need to be made to the implementation of Chapter 3 that still allow the AMN to be the final step. In order to do so, a new loss function is introduced called the spectral deception loss that replaces the previous power loss and looks to push the adversarial signal to exhibit spectral behavior more like the original signal. This approach would be realistic for both the black box transmission environment as well as one where the AMN has more access. Additionally, this would be a more feasible approach when considering modulation schemes that are not linear modulations.

4.3 Spectral Deception Loss

In this section, a variety of candidate spectrum deception loss metrics are presented, and their different impacts on the adversarial signal's spectral content are analyzed, along with their performance on eavesdropper evasion and intended communication capabilities. As previously discussed, it is desirable for the adversarial signal to have a similar spectral shape

as the original signal so that it avoids detection and defensive capabilities. In this work, we determine this similarity through the power spectral density (PSD) and associated phase plot of the original signal, perturbation, and combined adversarial signal.

4.3.1 Examining the Necessity of Deception Loss

With the previous work, there was uncertainty over whether the power loss metric was sufficiently useful at providing the desired intent of maintaining the original shape of the signal. This was due partially to the fact that the two main performance metrics, BER and evasion classification success, were driven directly by the communication and adversarial loss, respectively, and not by the power loss. Additionally, the power loss and communication loss were shown to push the network to converge in the exact same way for these metrics, as is shown in Figure 4.6, leading to unnecessary redundancy among these two losses.

It makes sense that these two losses would provide similar results for the chosen performance metrics; however, observation of the PSD of the resulting adversarial signal when prioritizing each loss highlights the true differences between them. An example of this difference is shown in Figure 4.7. Prioritizing the power loss results in a PSD shape for the perturbation that is similar to the original signal. On the other hand, prioritizing the communication loss results in a PSD that is more jagged in the center lobe and has significant side lobe content. From this result, it can be observed that the power loss metric steers the training of the AMN to keep the spectral shape of the original signal while the communication loss metric disregards the original signal shape as long as the intended receiver is minimally impacted.

While the power loss appears to provide the exact behavior desired to maintain spectral integrity, this is only true under an ideal scenario. In the power loss result shown in Figure 4.7, the power loss is the only loss prioritized; however, when being balanced with the

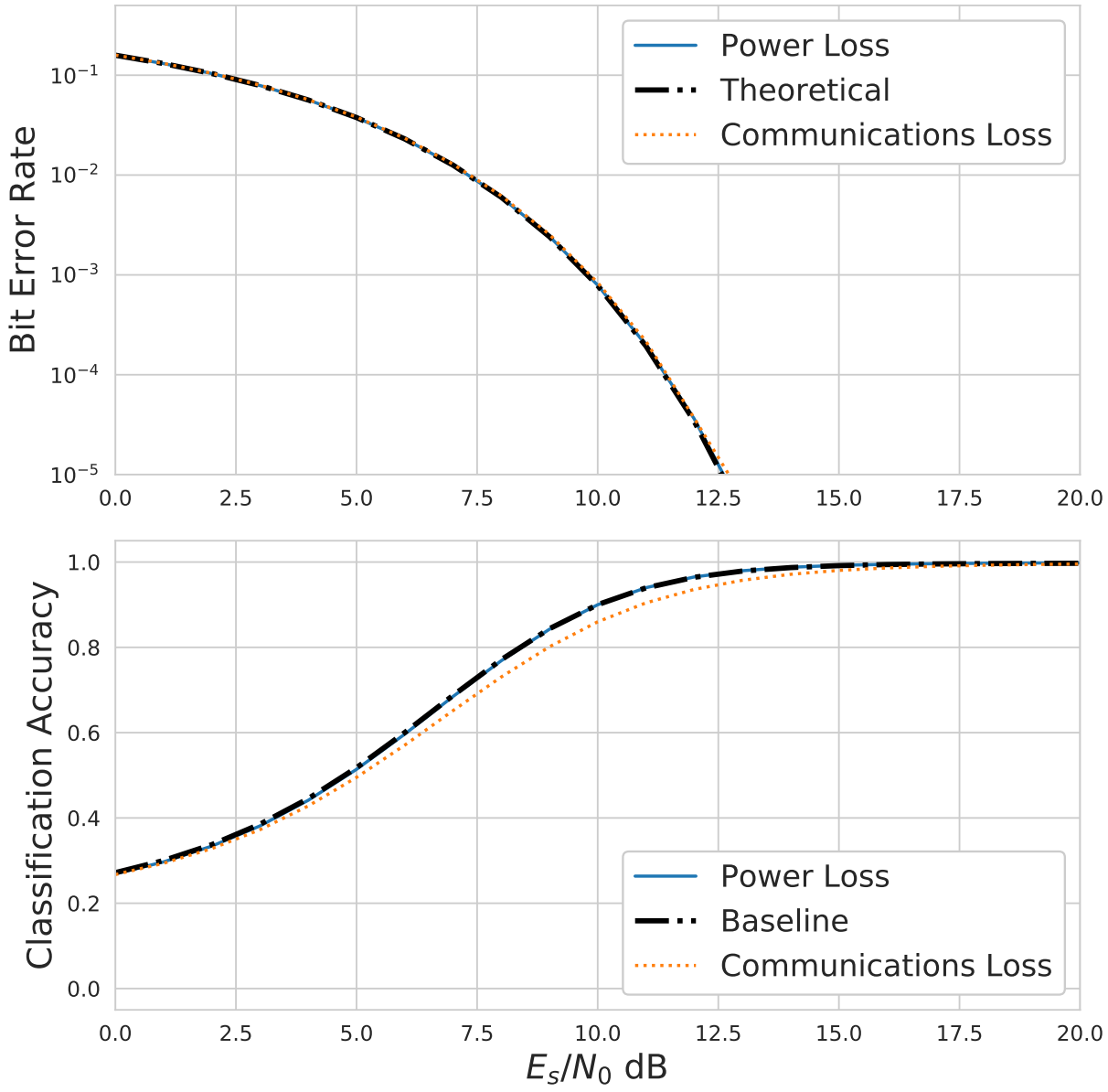


Figure 4.6: The BER and eavesdropper classification accuracy for QPSK adversarial signals when trained with either only communications loss ($\beta = 1$) or only power loss ($\gamma = 1$). The values are plotted over a range of 0-20 dB SNR. The theoretical values for the BER and classification accuracy of QPSK are shown.

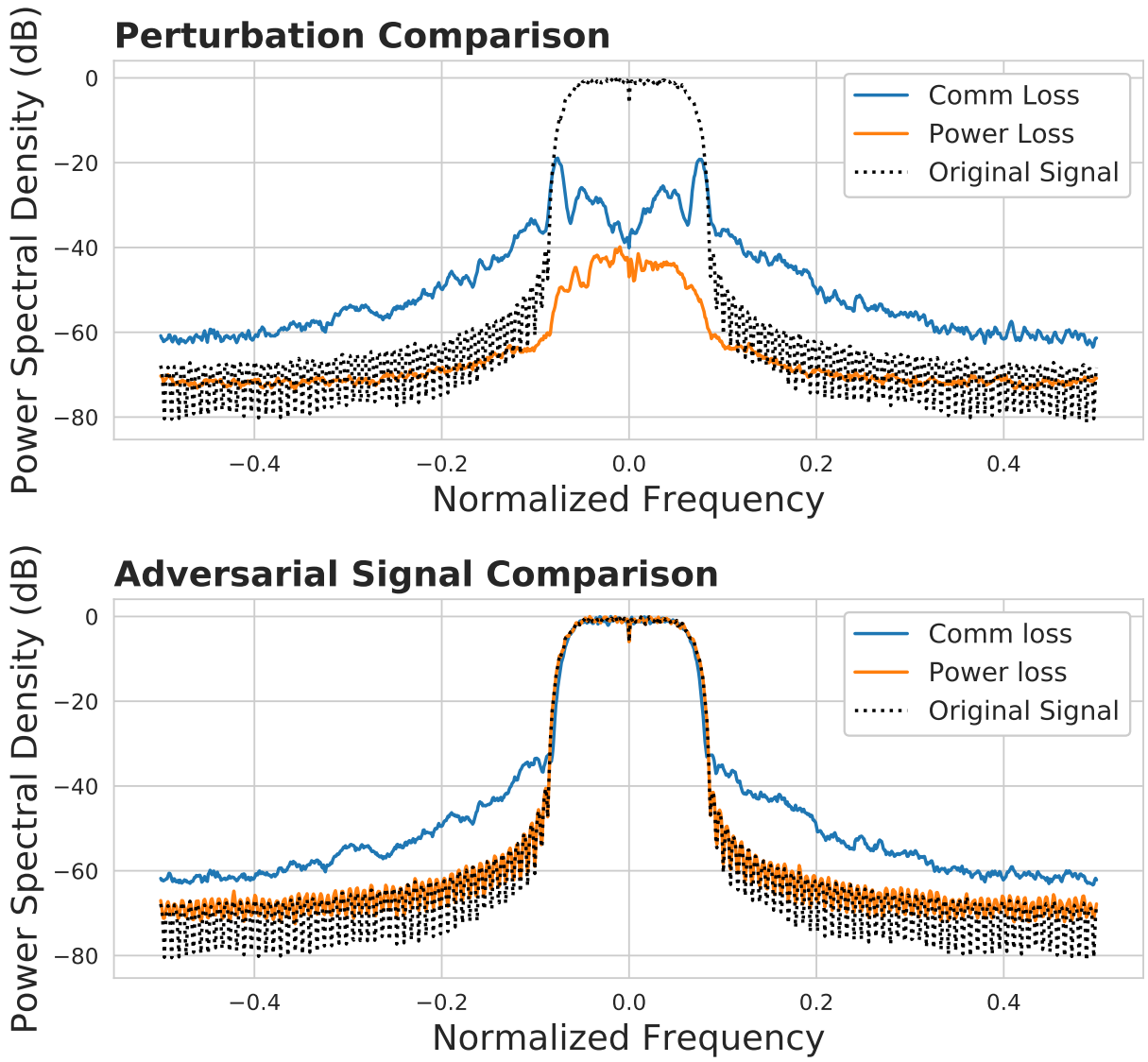


Figure 4.7: The PSD plots for the perturbations and combined adversarial signals created using either only communication loss or only power loss compared to the PSD of the original signal. The power loss configuration appears benign while the communication loss configuration has significant side-content.

communication and evasion losses, the shape, while still an improvement on previous work, no longer resembles a clean signal and has some side lobe content as shown by the sample-based perturbation in Figure 4.3. Spectral deception loss is introduced as a solution to this problem so that the spectral integrity can be preserved even when successfully evading and communicating, and is meant as a replacement for the power loss. The deception loss operates in the frequency domain and thus allows for the AMN to better control the frequency content of the signal as opposed to the prior power loss metric that controls the time content of the signal. This should allow for better success in shaping the signal.

4.3.2 Deception Loss

The deception loss method developed within this work is based upon the frequency domain characteristics of the signal. This is done so that the perturbation lies more in-band and thus the adversarial signal will exhibit less side-lobe content and appear more benign. There are two approaches considered in this work. The first is to force the perturbation to be similar in shape (based off the resulting PSD) to a clean signal and the other is to make the total adversarial signal (perturbation added to the original signal) similar in shape to a clean signal. While the ultimate goal is to make the combined adversarial signal similar to a benign signal (the latter approach), it was seen in previous results that when the perturbation was in-band, the combined signal was as well (such as in Figure 4.3). It could prove to be easier to shape the perturbation rather than the combined signal so both methods are considered. Further, two representations of the signals are used, the fast Fourier transform (FFT) and the PSD. Thus, there are 4 approaches considered. Minimizing the difference between:

- The original signal and perturbation's FFT.
- The original signal and the perturbation's PSD.

- The original signal and the combined signal's FFT.
- The original signal and the combined signal's PSD.

A function must be used in order to quantify the difference between the two signals. Three functions, mean squared error (MSE), mean absolute error (MAE), and Huber, are examined in this paper for their potential use in the deception loss and described further next.

4.3.3 Mean Square Error

Mean Square Error (MSE) is a regression loss function that determines the difference between expected and actual values. In this paper, MSE is used as the average squared difference between the original signal and either the perturbation or adversarial signal output by the AMN. MSE is defined as:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.1}$$

where y is the value of the original signal and $\hat{y}$ is the actual output. After calculation, the MSE is normalized to the range $0 \leq \text{MSE} \leq 1$ (essentially a max normalization or form of Normalized MSE (NMSE)) to better align with the scale of the communication and adversarial loss values.

4.3.4 Mean Absolute Error

While MSE is suitable for most scenarios, it does not handle outliers very well, as these drastically bias the loss (the loss gets squared so large losses become even bigger by comparison).

Mean Absolute Error (MAE) is a potential solution to this issue. The equation is defined as:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

where the inputs to the function and normalization process are the same as above for MSE. While MAE handles outliers better than MSE, it has a more difficult time converging due to the slope of the loss not changing as the loss gets closer to zero (whereas the slope of MSE loss decreases around this value), and since the loss has no gradient when $\hat{y}_i = y_i$.

4.3.5 Huber

Huber is a hybrid loss function that offers a balance between MSE and MAE. It handles outliers better than MSE, like MAE, but allows for better convergence at low loss values like MSE. Huber loss mitigates the affect of outliers through an adjustable value, δ. If the absolute difference between the expected and actual value is less than δ, then Huber loss calculates their difference using an equation similar to MSE. Otherwise, the affect of the outlier is adjusted using MAE. The Huber loss function is shown below.

$$\begin{cases} \frac{1}{2}(y-\hat{y})^2 & |y-\hat{y}| \leq \delta, \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{4.3}$$

where y is the value of the original signal and $\hat{y}$ is the value of the perturbed signal or perturbation itself depending on the method used as is described in the next section. The equation above specifies the function used to calculate the difference between the value of δ used in this work is 1 and the loss is normalized similar to the other metrics.

The deception loss metrics to be discussed within this work are based upon the frequency domain characteristics of the signal. The deception loss is intended as a replacement for the power loss that sought to minimize the perturbation power in order to maintain similarity with the original signal. For each of the potential approaches detailed above, some or all of the loss functions (MSE, MAE, and Huber) are tested. Huber and MSE were the predominant functions considered given MSE's high usage in machine learning techniques and Huber's robustness and tendency to exhibit the positive aspects of both MSE and MAE. In the following subsection, each of these methods will be analyzed within the considered AMN evasion attack framework.

4.3.6 Results

The primary qualitative metric used when examining the success of the various spectral deception loss methods at maintaining the spectral shape was visual inspection of the PSD. Phase plots were additionally examined for the PSD-based deception loss approaches to see if they still maintained phase information. Quantitative metrics used to validate the success of the considered metrics include the BER of the intended communications link and the achieved reduction in classification accuracy of the eavesdropper. The results presented in this section are predominantly examined with AMNs trained for BPSK and QPSK modulated signals. However, other modulation schemes were also tested and exhibited the same characteristics but are partially omitted for brevity. The eavesdropper's AMC used in this work was trained on BPSK, QPSK, 8-PSK, 16-QAM, and 64-QAM. The lost constants used are set based on multiple trials and those shown are used to illustrate the general trends.

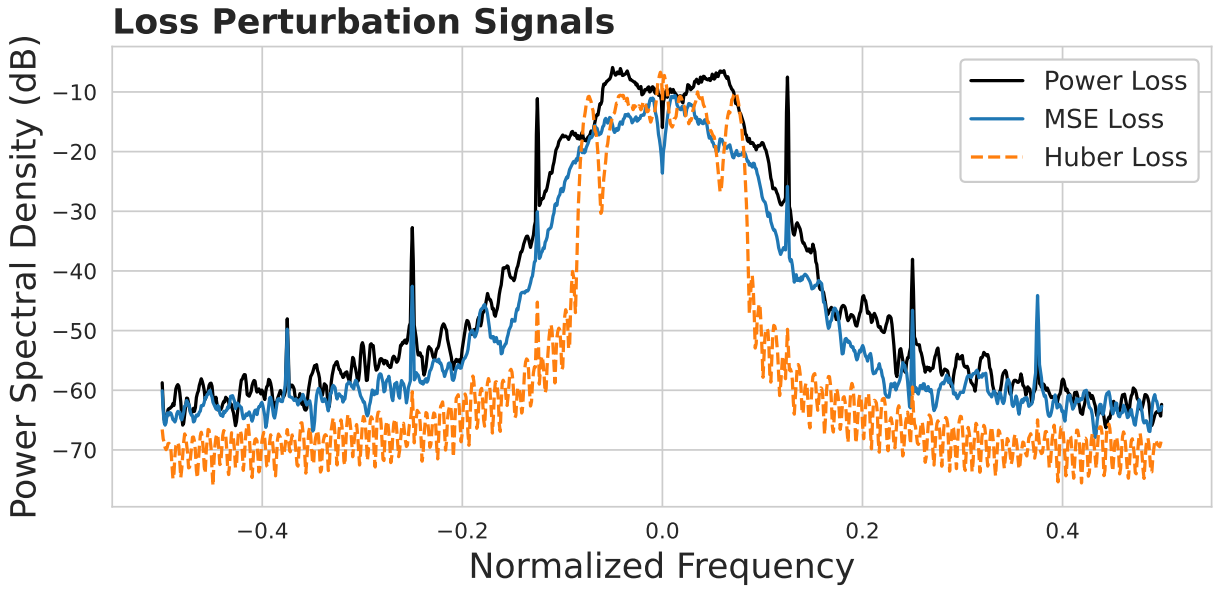


Figure 4.8: The PSD for the perturbations created by the original power loss and both the MSE and Huber loss methods on the FFT of the perturbation for BPSK signals.

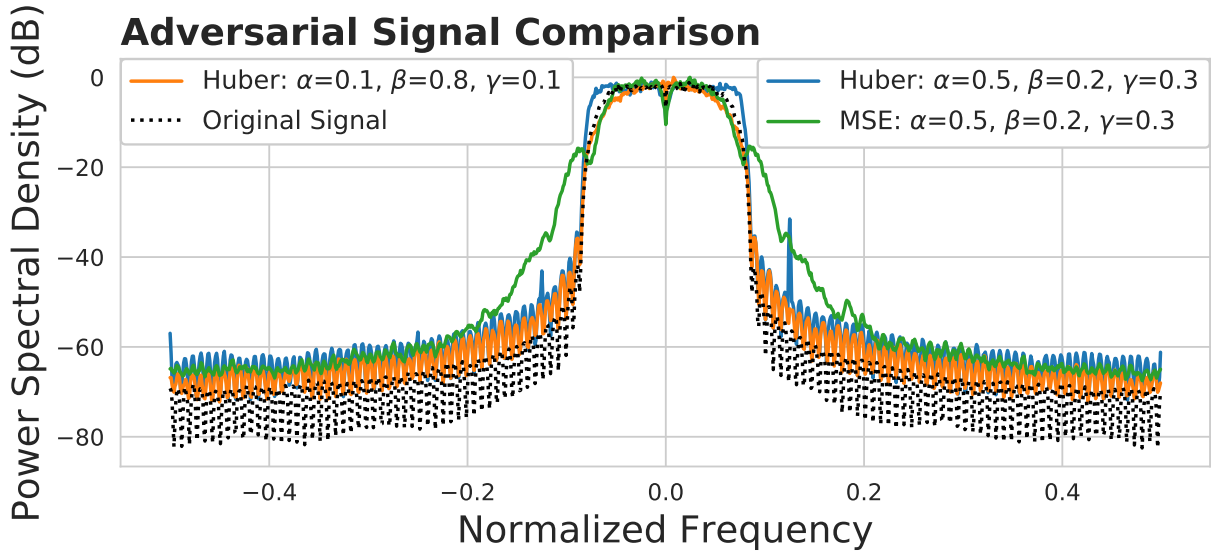


Figure 4.9: The PSD for the adversarial signals created by the MSE and Huber loss methods on the FFT of the perturbation for QPSK signals.

FFT Approach with Perturbations

As mentioned previously, this FFT-based approach was tested using both the MSE loss function and the Huber loss function. Figure 4.8 shows the resulting PSDs of just the perturbation for the MSE loss, Huber loss, and the original power loss from Chapter 3 for BPSK signals. This figure illustrates that there is slight improvement with the MSE method over the power loss metric, but very minimal. On the other hand, the Huber loss method exhibits much better behavior over the power loss metric of Chapter 3 given that the shape of the perturbation is much more in-band to the original signal. This difference is due to the fact that the Huber loss is able to better handle situations of extreme error, which can occur during the training process especially at the start of training. Figure 4.9 shows the PSDs of the resulting combined adversarial signals for QPSK signals tested with this method. As can be seen from this figure, the Huber loss method is much more successful than the MSE loss method in minimizing side-band content of the combined signal while both offer similar main-lobe corruption power.

As expected, this trade-off in spectral shape performance comes at the detriment of intended communication performance. Figure 4.10 shows the BER and eavesdropper classification accuracy over the SNR range of 0-20 dB for the two methods, along with the theoretical QPSK bit error rate with no perturbation added. When using loss constants of $\alpha = 0.5$, $\beta = 0.2$, and $\gamma = 0.3$, the BER rate for the Huber method is much worse than that of the MSE method. Additionally, when the MSE deception loss is the only loss considered during training (i.e. α and β are set to 0), the BER converges to the theoretical curve, which does not occur for the Huber loss. However, by adjusting the loss constants, the communication performance can improve as is shown by the Huber result with $\alpha = 0.1$, $\beta = 0.8$, and $\gamma = 0.1$. Naturally, this does lead to worse evasion performance. Interestingly, in Figure 4.9 it can be observed that the resulting spectral shape of the adversarial signal does not seem to

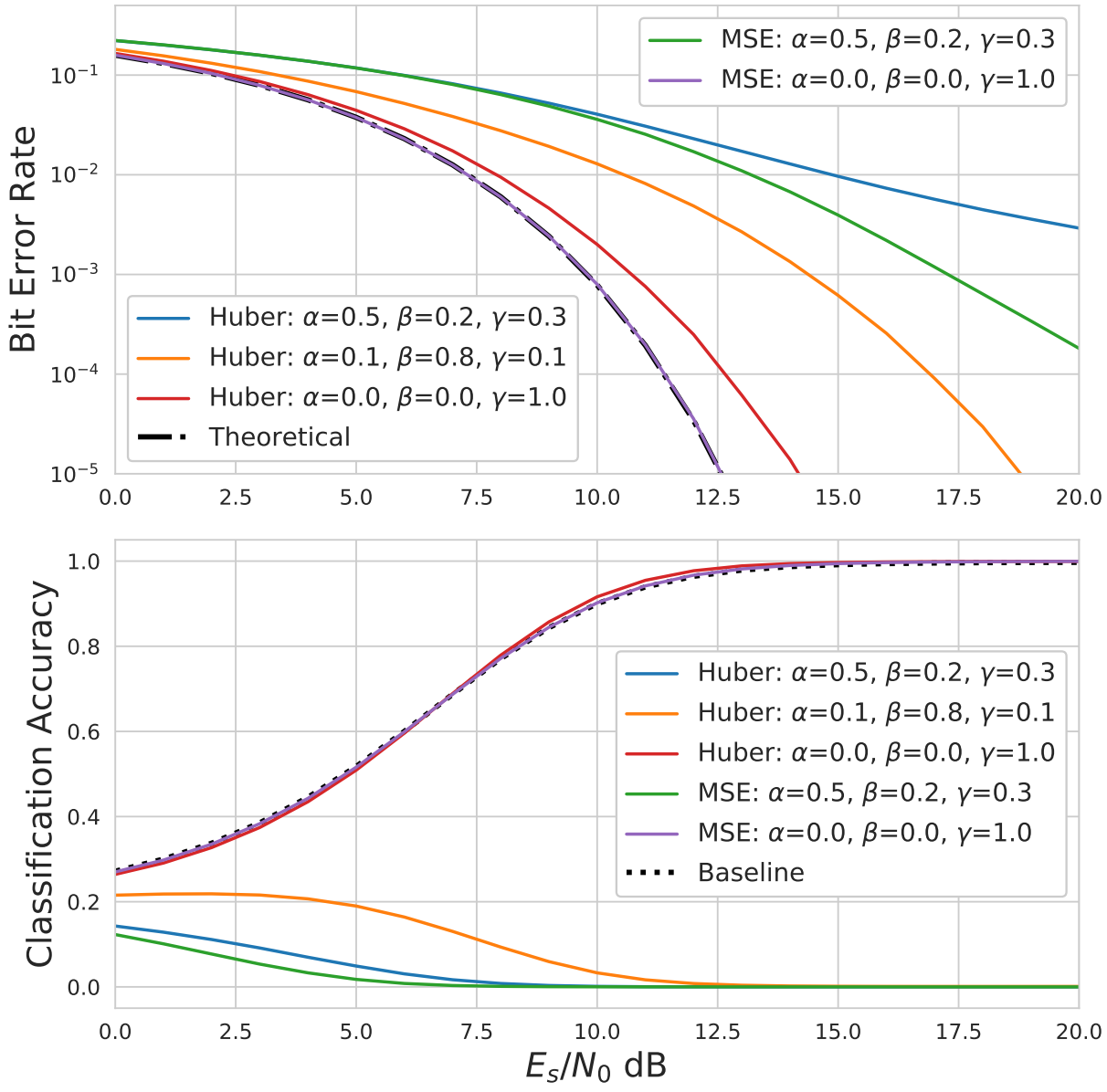


Figure 4.10: The BER and eavesdropper classification accuracy for QPSK adversarial signals when with trained the deception loss attempting to minimize the difference between the FFTs of the perturbation and original signal using both Huber and MSE loss. The signals correspond to those shown in Figure 4.9

drastically change for this second Huber trial even though the deception loss is less prioritized. This shows that the constants can be adjusted to meet the needs of the attack and that when using Huber loss in the deception loss, the spectral shape can be maintained even when less prioritized (so more priority can be spent on evasion or communication improvement). This shows a more reasonable trade-off between spectral integrity and communication success.

FFT Approach with Combined Signals

The next approach examined is that of trying to minimize the difference between the FFT of the combined adversarial signal and that of the clean, original signal. Like for the FFT-based approach that operated on the perturbation, attacks with $\alpha = 0.5$, $\beta = 0.2$, and $\gamma = 0.3$ were first tried but also showed poor communication success like that seen in Figure 4.10. For this reason, attacks with $\alpha = 0.1$, $\beta = 0.8$, and $\gamma = 0.1$ were predominantly examined. In addition to Huber and MSE, MAE loss was also tried to see if it offered any better results. The results shown are for QPSK signals.

Figure 4.11 first shows the resulting PSD plots of all three methods of attack. As was seen previously, MSE still exhibits some out of-band content. The Huber and MAE methods, on the other hand, result in adversarial signals that appear to match the clean signal perfectly. The combined adversarial signal's PSD is more similar to the original signal's PSD for this approach than in the first deception loss method examined in the previous sub-section because the side content shown in Figure 4.8 was slightly more powerful for the adversarial signal than for the original. This is not the case in this second method, shown in Figure 4.11. This is due to the fact that the previous method attempted to shape the perturbation to cause the combined signal to appear like the benign, clean signal whereas this method does it directly. Interestingly, the MAE perturbation appears to be less powerful outside the main lobe than even the original signal, something not seen before. Also, the PSD of the

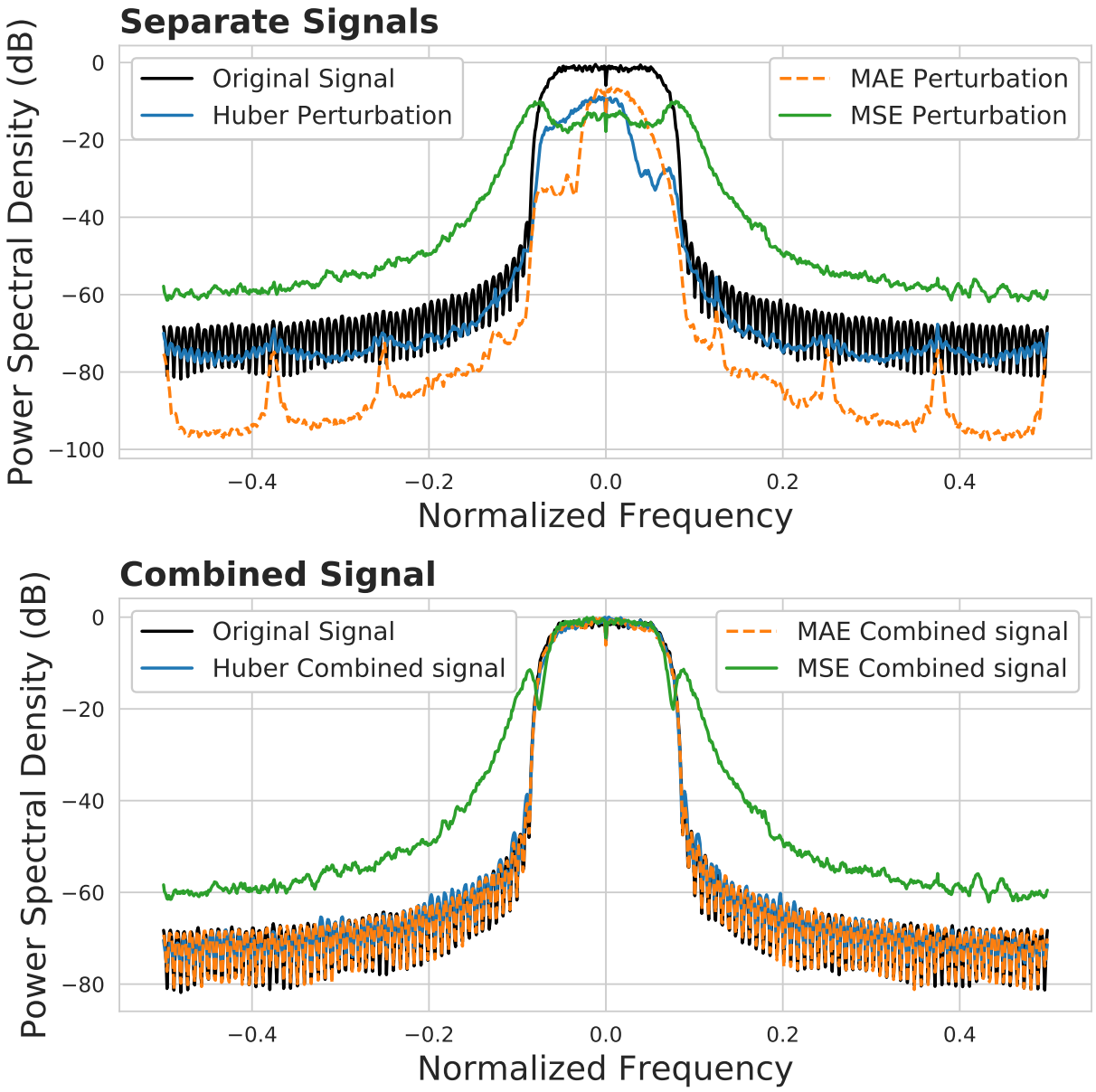


Figure 4.11: The resulting PSD plots created by the MSE, MAE, and Huber loss methods on the FFT of the combined signal for QPSK signals. MAE and Huber methods both qualitatively match and mimic the original signal while the MSE method exhibits out-of-band content.

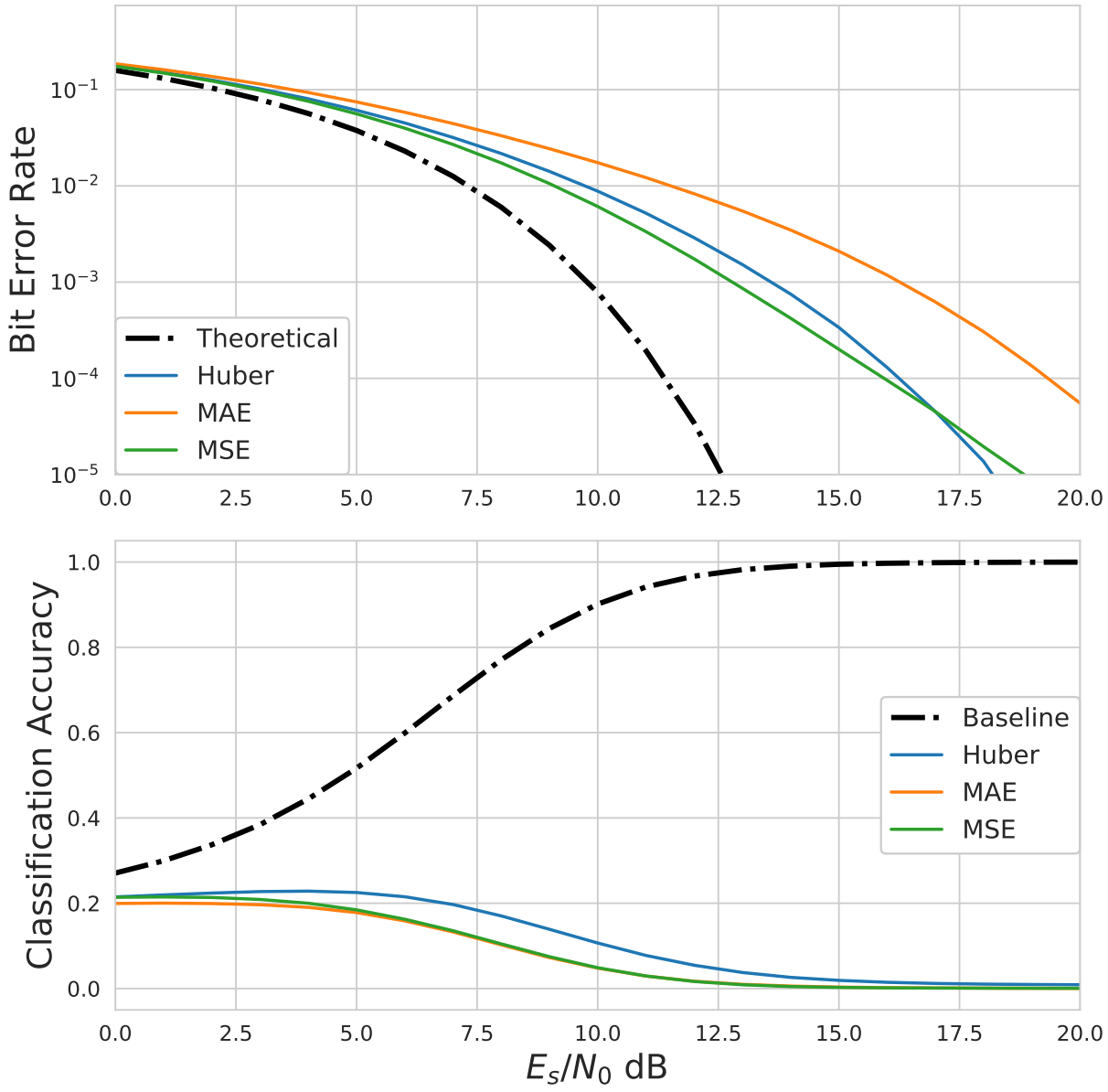


Figure 4.12: The BER and eavesdropper classification accuracy for QPSK adversarial signals when with trained the deception loss attempting to minimize the difference between the FFTs of the combined adversarial signal and original signal using both Huber and MSE loss. The signals correspond to those shown in Figure 4.11.

perturbations for MAE and Huber do not follow the same rectangular shape seen for the perturbations in Figure 4.8 but are more irregular. This is because the goal is just to shape the combined signal's spectral shape for this method and not the perturbation.

When examining the BER and evasion success in Figure 4.12, the communication reliability is good and the attack on the eavesdropper is still successful even with the small value of α. The BER is actually better for the Huber method at higher SNRs than it is for MSE even though Huber is much better for the spectral integrity. MAE is noticeably less successful in it's communication ability than Huber even though they offer the same result in the PSD. Therefore MAE is not considered to be as viable of a solution as Huber.

The results shown in Figure 4.11 and 4.12 as well as those shown for the FFT-based perturbation approach indicate that these attack methods are valid solutions to the problem of spectral integrity, based on qualitative observations of the PSDs. *They are able to mimic the PSD of the original signal without much degradation, if any, to the communication and evasion success. This form of attack accomplishes the goal of lessening the likelihood of detection and maintaining compliance with the spectral mask.*

PSD-based Methods

As was done for the FFT-based methods, two PSD-based methods were examined: one that tried to minimize the difference between the PSDs of the perturbation and the original signal and one that did so for the distance between the combined adversarial signal and the original. This PSD method is attempted because the goal is to match the PSD of the original signal so it would make sense to use this representation directly for the loss. For both PSD methods, all three losses (MSE, Huber, and MAE) were tried. Unfortunately, neither method proved fruitful.

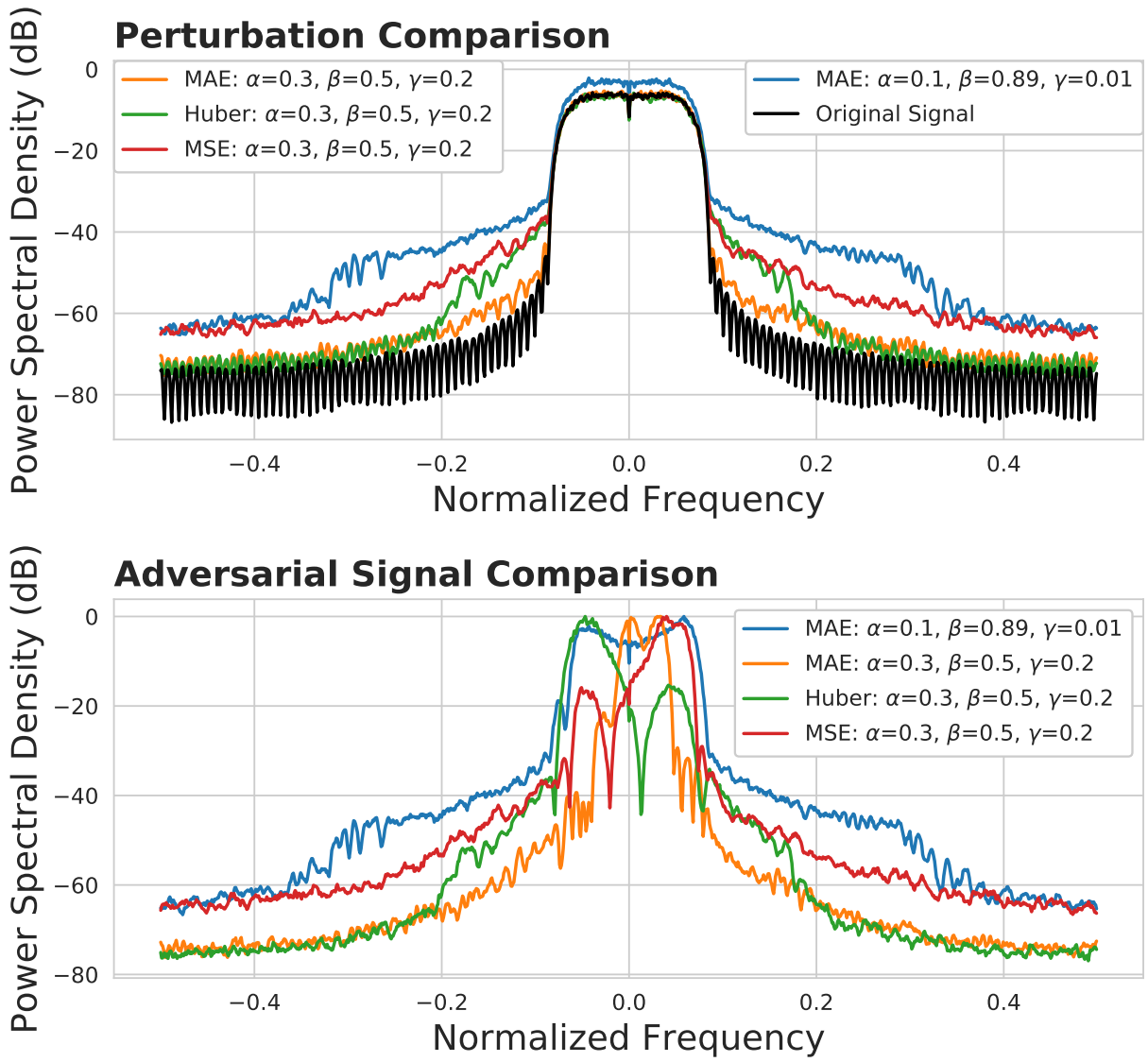


Figure 4.13: The PSDs for the perturbations, original signal, and combined adversarial signals created by the MSE, Huber and MAE loss methods for the perturbation PSD-based approach. Loss constants of $\alpha = 0.3$, $\beta = 0.5$, and $\gamma = 0.2$ are used for all as well as another test of MAE loss with less priority on the deception loss.

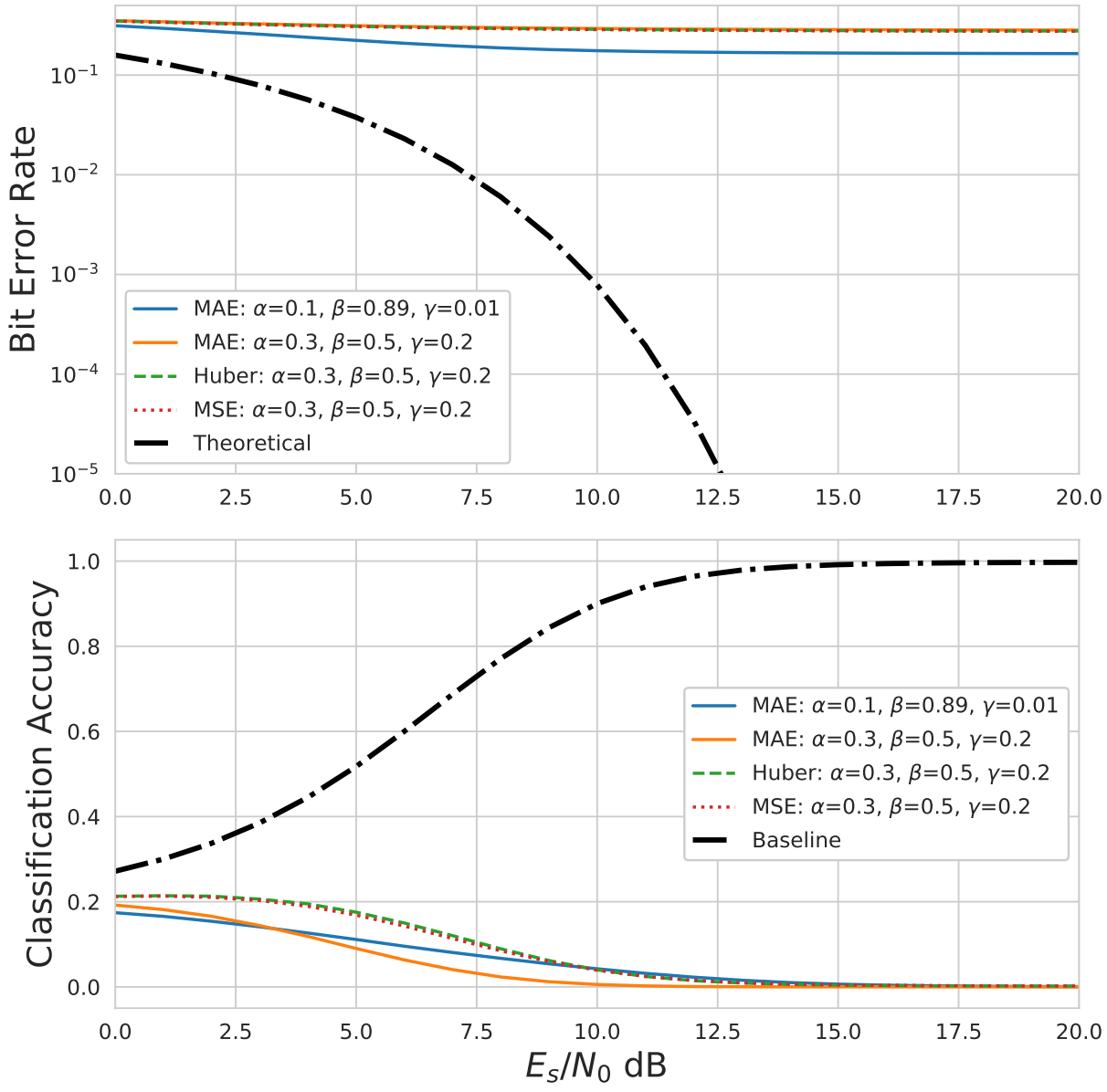


Figure 4.14: The BER and eavesdropper classification accuracy for QPSK adversarial signals when trained with the deception loss performed on the PSDs of the perturbation and original signal using MAE, Huber, and MSE loss. The signals correspond to those shown in Figure 4.13. While the evasion success is good, the communication is not suitable for use.

Figure 4.13 shows the PSDs of the perturbation-based methods. While it appears that they do a decent job of matching the perturbation to the original signal (especially the MAE method when γ is larger), this completely destroys the shape of the adversarial signal which no longer resembles a benign signal. Additionally, the communication is completely unsuccessful with this method. As is shown in Figure 4.14, the BER is extremely high and does not decrease that much as SNR increases. One trial was run for MAE loss with an increased priority on the communication but this only slightly improved the BER while having a very drastic effect on the perturbation shape, which now lies greatly out of band.

The method utilizing the combined, adversarial signal did not fair much better. Figure 4.15 shows the resulting PSDs. While the adversarial signal appears more regularly shaped than the perturbation-based PSD method, it does not look as benign as the adversarial signals generated with the FFT-based methods. Additionally, as is seen in Figure 4.16, the BER is once again very bad. MAE and Huber loss are better than that of MSE loss but still much worse than would be needed for proper communications.

The reason for this poor performance is in part because of the loss of phase shift information. While trying to minimize the difference between the adversarial signal's PSD and a benign signal's PSD makes sense given that this is the metric being considered most important, doing so makes it much harder to maintain the same phase shift. This is due to the fact that the PSD is based off the magnitude of the FFT while the phase shift is based off the angle. In the FFT-based approaches, the proper phase angles can still be learned since this information is inherent in the FFT used to calculate the loss; however, for the PSD-based approaches, this information is lost when the magnitude is calculated for the PSD leading to a worse adversarial signal in terms of spectral integrity. This is first shown in Figure 4.17 where the phase angle of the perturbation, combined adversarial, and original signal is plotted across the entire normalized frequency domain. The hope would be that

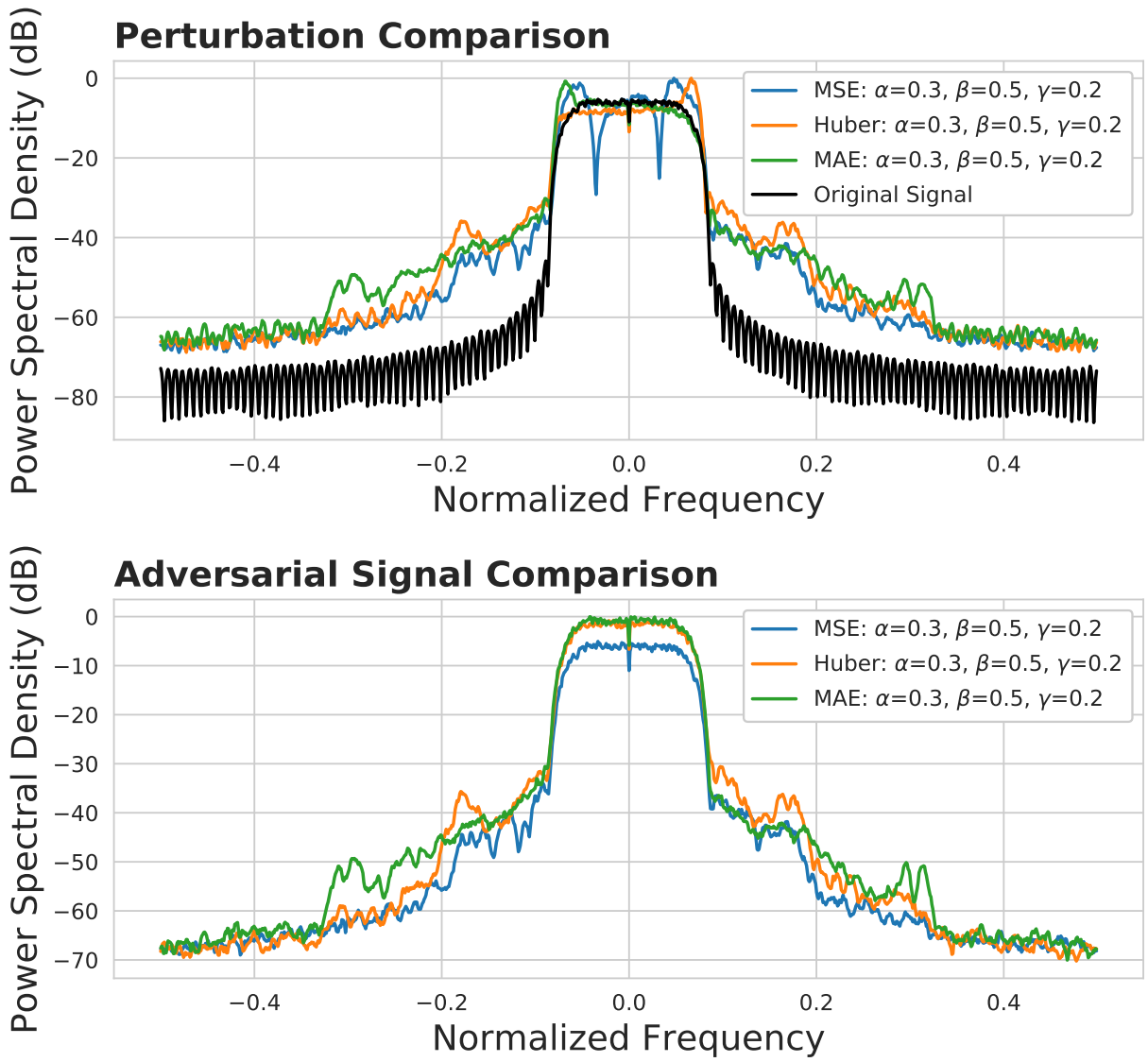


Figure 4.15: The PSDs for the perturbations, original signal, and combined adversarial signals created by the MSE, Huber, and MAE loss methods for the combined signal PSD-based approach. Loss constants of $\alpha = 0.3$, $\beta = 0.5$, and $\gamma = 0.2$ are used for all.

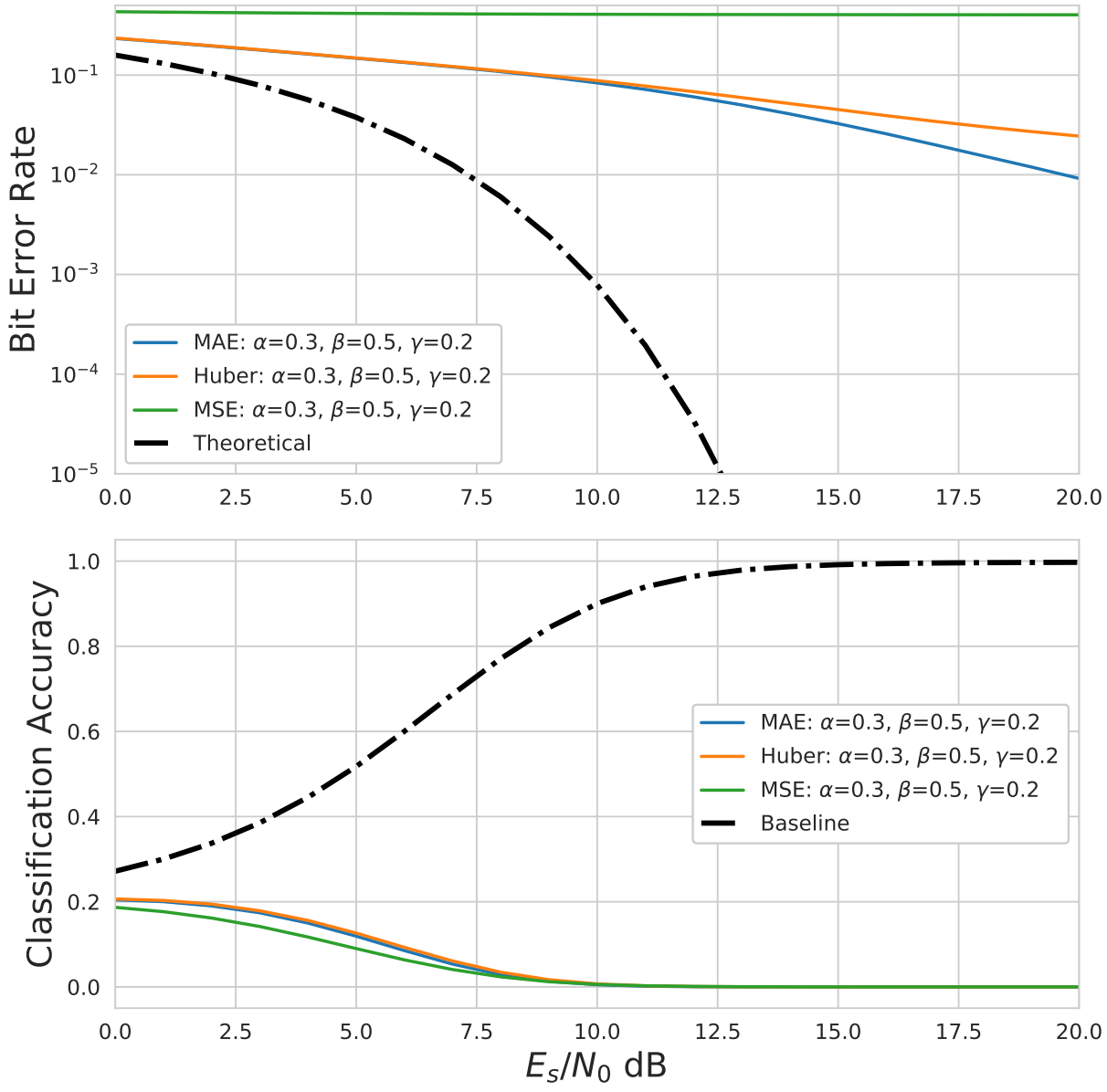


Figure 4.16: The BER and eavesdropper classification accuracy for QPSK adversarial signals when trained with the deception loss performed on the PSDs of the combined signal and clean signal using MAE, Huber, and MSE loss. The signals correspond to those shown in Figure 4.15. While the evasion success is good, the communication is not suitable for use.

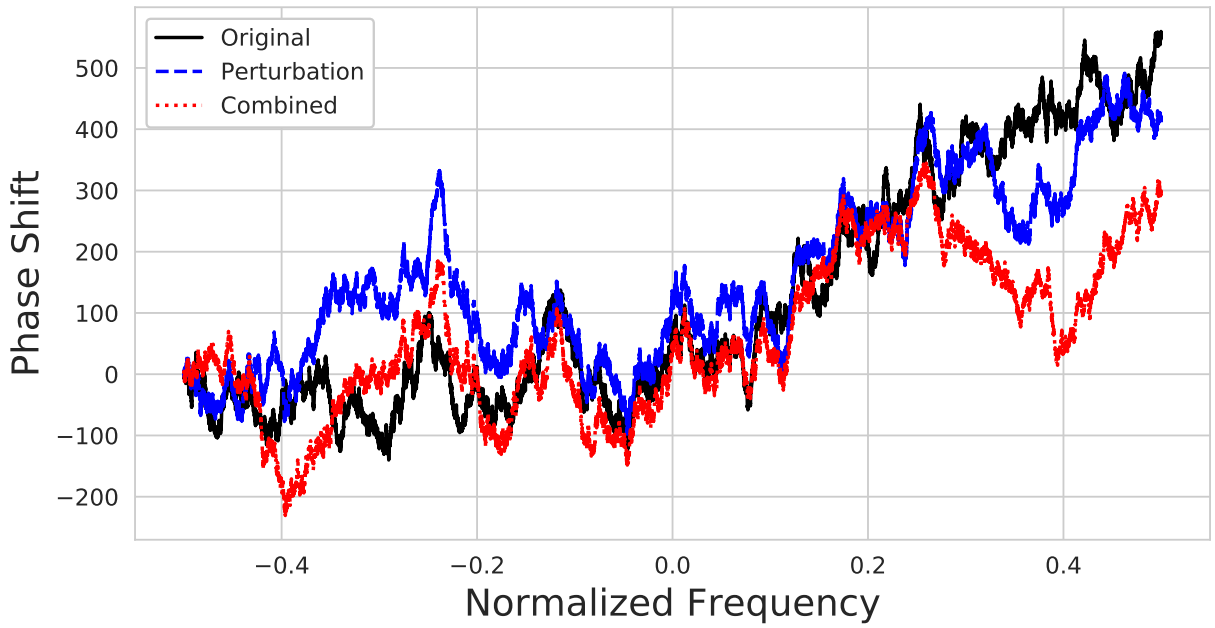


Figure 4.17: The unwrapped phase shift plot for the perturbation and combined adversarial signal created with the PSD-based approach on the combined signal, as well as the original signal. The phase of the combined signal and original appear to be different, especially at both ends of the plot.

the adversarial signal and original signal would have similar phase shift but that is not the case. The phase of the original signal has a linear nature whereas the combined adversarial signal does not. Further, Figure 4.18 shows the wrapped phase shift for the middle 64 bins of the normalized frequency domain for both the adversarial and original signal. While there is some similarity, they are shifted from one another (the adversarial signal is shifted up). These plots indicate loss of phase information as would be expected since the PSD represents just the magnitude of the FFT.

An additional reason behind the poor performance when using these PSD method is the added complexity in learning to shape the signal. While the FFT methods only required calculating the loss based off an FFT of the signals, the PSD methods require taking the magnitude of multiple windows of the resulting FFT, averaging this, and calculating the loss from the average. This is a much more complex process and one that therefore makes the

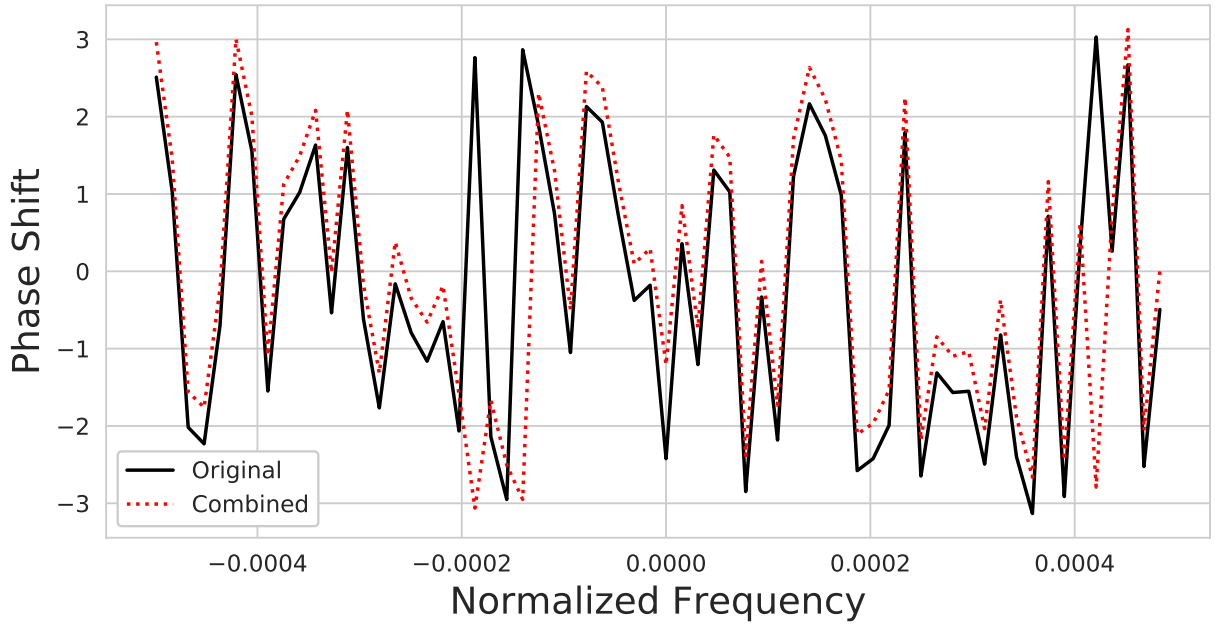


Figure 4.18: The wrapped phase shift plot of the adversarial signal and original signal for the middle slice of the normalized frequency. The signals are the same as those represented in Figure 4.17. While there are similarities, the the phases of the two signals are shifted apart.

back-propagation and learning process much more convoluted and difficult.

The extremely poor BER and lack of benign spectral shape for the adversarial signal together show that *the PSD-based methods are not as ideal for the deception loss as the FFT-based methods.* Given that both the magnitude and phase shift are maintained in the FFT-based approaches, these methods are much better suited to be used for the spectral deception loss.

Additional Modulation Schemes

The above results considered were all utilizing QPSK modulation; however, it is important that these results are shown to be generalized to other modulation schemes. For this reason, trials were conducted for BPSK, 8-PSK, and 16-QAM using the FFT-based approach on the combined signal, given that this approach resulted in adversarial signals that better

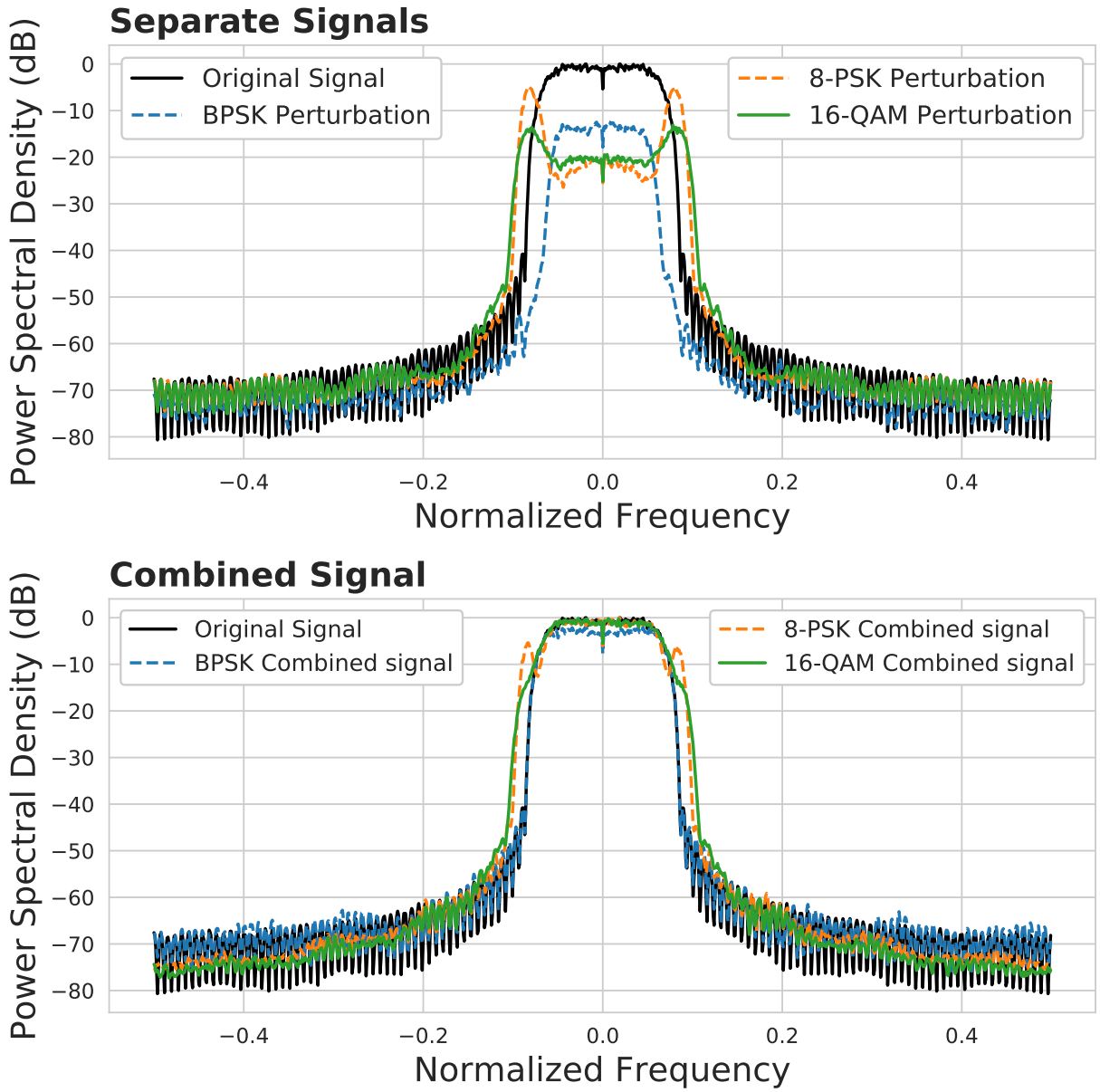


Figure 4.19: The PSD plots for BPSK, 8-PSK, and 16-QAM adversarial signals. These adversarial signals were created using an AMN trained with a deception loss attempting to minimize the difference between the FFT of the original signal and the combined signal. The loss constants are $\alpha = 0.1$, $\beta = 0.8$, and $\gamma = 0.1$. The attacks are able to successfully shape the frequency content of the adversarial signals to be similar to the original signal just as QPSK did.

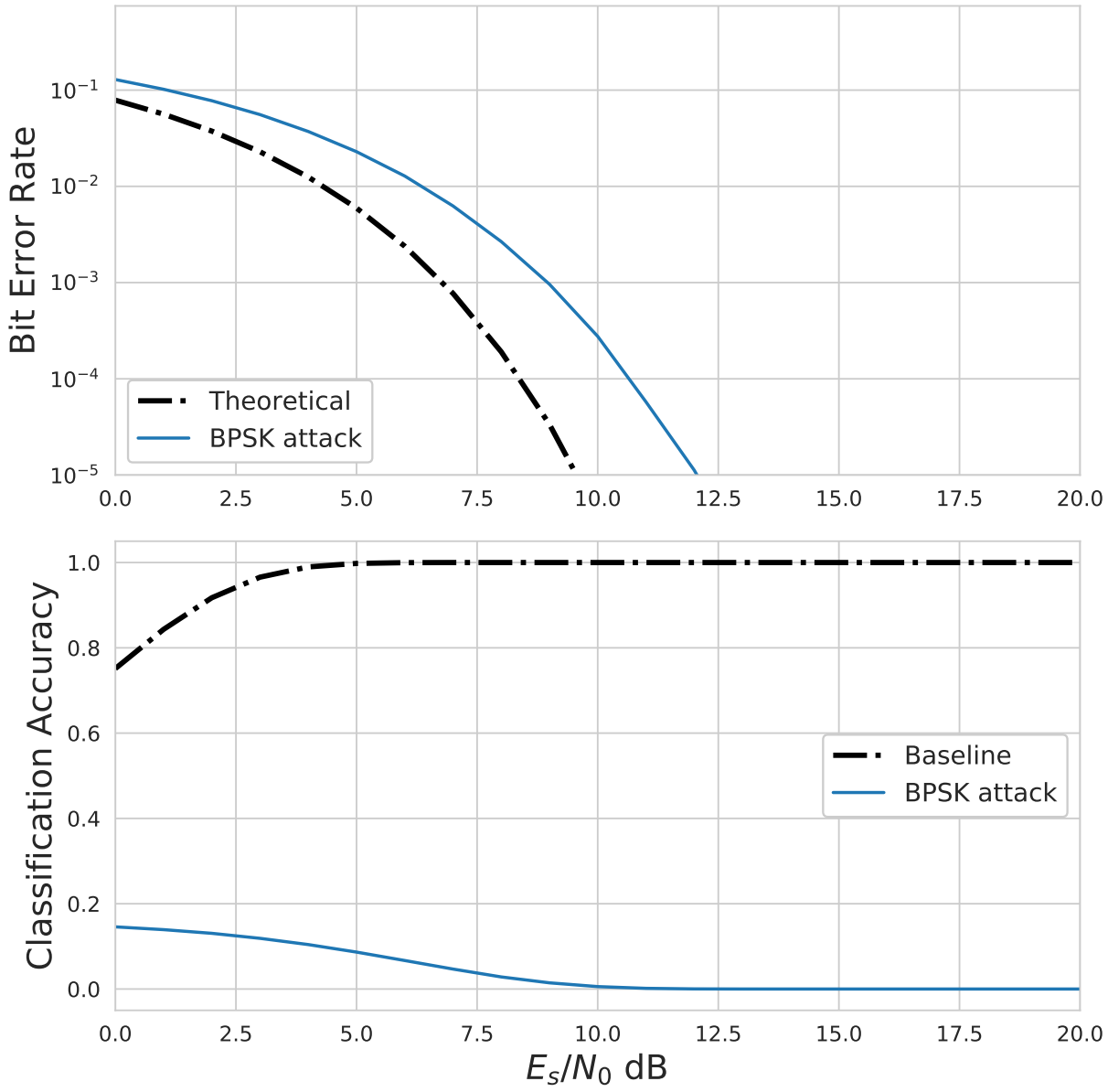


Figure 4.20: The BER and eavesdropper classification accuracy for the BPSK adversarial signal shown in 4.19.

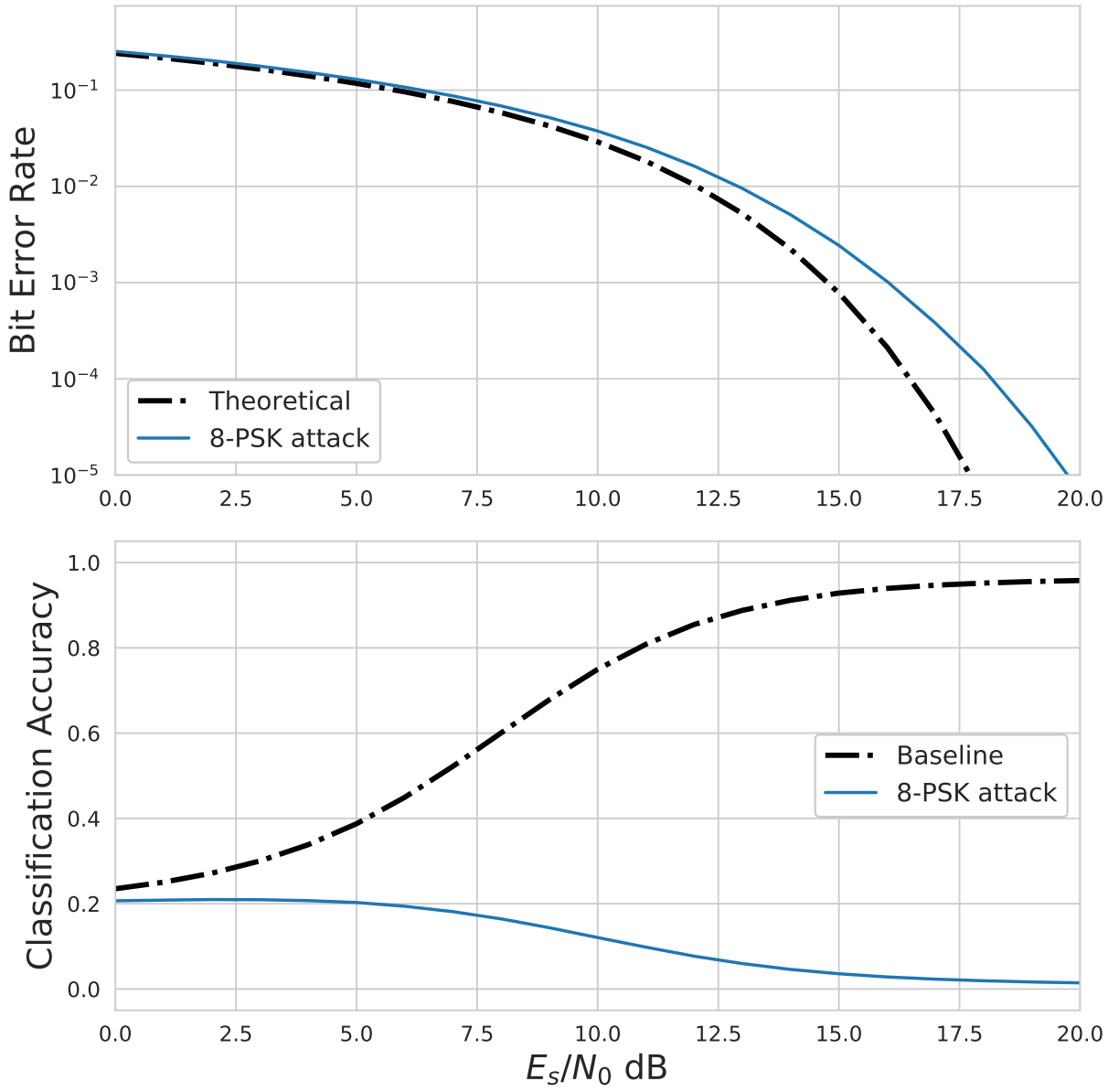


Figure 4.21: The BER and eavesdropper classification accuracy for the 8-PSK adversarial signal shown in 4.19.

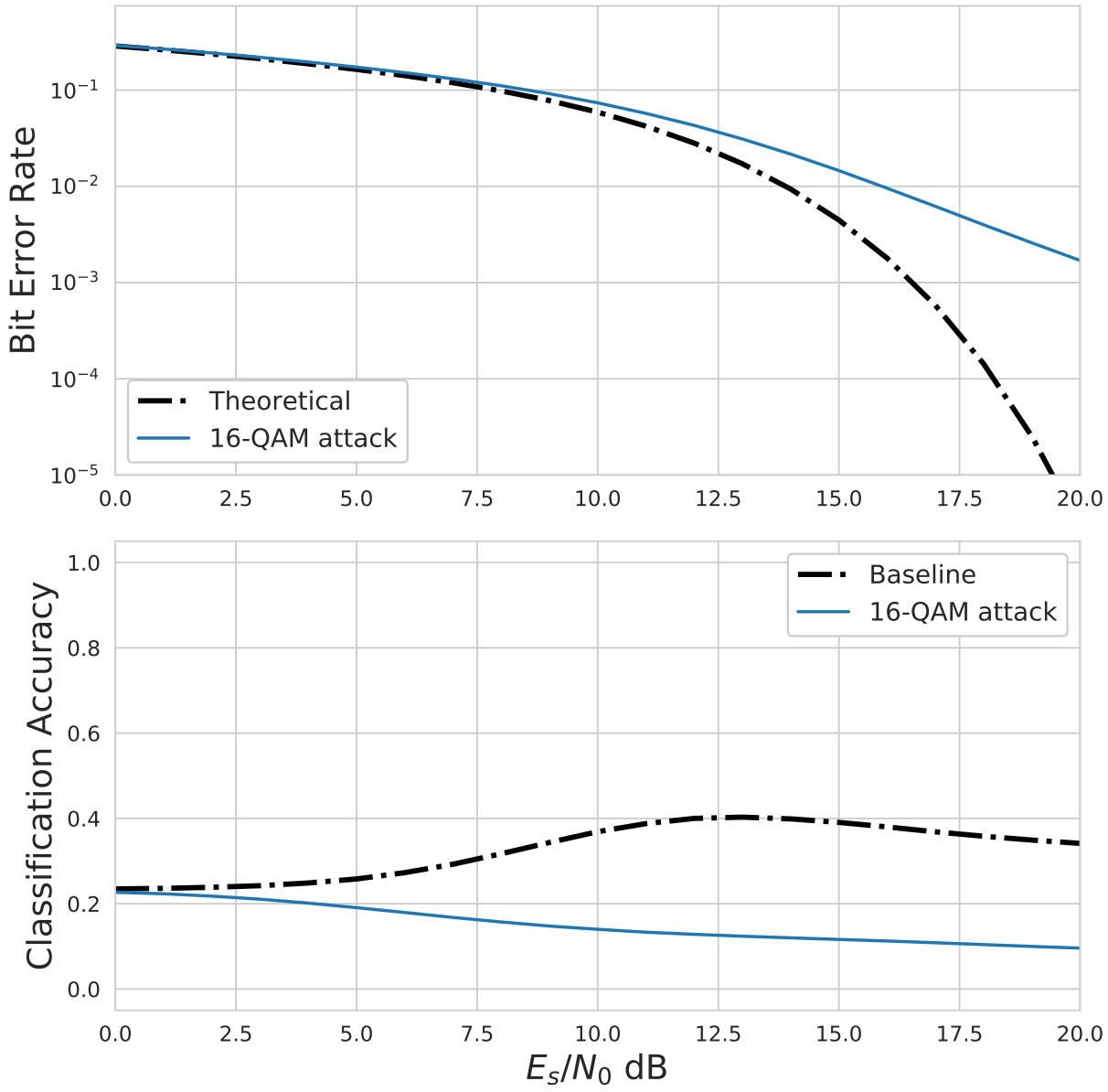


Figure 4.22: The BER and eavesdropper classification accuracy for the 16-QAM adversarial signal shown in 4.19.

matched with the PSD of a clean signal. Figure 4.19 shows the PSDs for BSPK, 8-PSK, and 16-QAM. Figures 4.20, 4.21, and 4.22 then show the BER and classification accuracy for BPSK, 8-PSK, and16-QAM respectively. When looking at the resulting PSD plots, it can be seen that the same trends seen for the spectral characteristics of QPSK continue for the other modulation schemes. 8-PSK and 16-QAM vary slightly from the PSD of the original signal but all three modulation schemes are successfully able to match the PSD of original signal, lying almost entirely in-band. The BER and evasion success for all three modulation schemes further show the success of the attack. Even with the improved spectral integrity, the evasion success and BER indicate successful and allowable metrics for an attack.

The resulting PSD, BER, and classification accuracy for the three additional modulation schemes of BPSK, 8-PSK, and 16-QAM show that *the spectrum deception loss is successful across multiple modulation schemes, not just QPSK.*

4.4 OFDM Consideration

This work has considered linear modulation schemes employed using sequentially transmitted symbols in the time domain to relay information bits. One alternative method of communication that has dominated a wide variety of real world systems is orthogonal frequency division multiplexing (OFDM). OFDM has been adopted for use in IEEE 802.11 standards (WiFi), 4G LTE and 5G cellular systems, WiMAX, Digital Audio Broadcast, and more due to its efficiency in transmitting bits in parallel, allowing for high-speed data transfer [78]. While each modulated symbol was sent in succession in the communication setup considered in this work, OFDM takes a specified number of these modulated symbols and sends them in parallel as one OFDM symbol. The way it does this is through the use of multiple subcarrier frequency bands. Each modulated symbol is transmitted through one of these

subcarriers that are orthogonal to one another to ensure no inter-channel interference (ICI) [78]. Because of this, multiple modulated symbols can be transmitted at one time. Given the prevalence of OFDM systems, it is important to ensure that this work can be utilized by a system that employs OFDM. While true testing of this is left to future work, a brief discussion about the feasibility of the work in this thesis on such a system follows.

When it comes to spectral integrity, both methods introduced should still apply and be successful. Since OFDM still uses schemes, such as the linear modulation schemes from this work, as the underlying form of modulating the symbols, the approach of perturbing the symbols would still be effective. OFDM works by inputting all of the modulated symbols into an inverse fast Fourier transform (IFFT) to create the combined OFDM signal. By perturbing the symbols before this is done, the attack should have no affect on the orthogonality of the subcarriers and resulting OFDM signal. Similarly, the FFT method of deception loss which operates on the samples should also remain successful when implemented on OFDM signals assuming the spectral integrity is sufficiently prioritized. There will be some spectral regrowth as was seen in prior work but the spectral deception loss that utilizes the FFT of the signals can account for this. Given the added importance that maintaining orthogonality puts on creating adversarial signals with similar spectral content to the benign signals, the deception loss may need to be prioritized more. Otherwise the orthogonality could be destroyed, resulting in difficulty interpreting the signal at the receiver; however as long as this is addressed by the weighting of the loss functions, the FFT should be well-suited to handle shaping the signal to meet this need. The concept of calculating the FFT of the received signal is still valid and is actually part of the process of extracting the information at the receiver in an OFDM system. While the spectral shape and frequency content of the resulting OFDM signal would be different than what was observed by the signals in this work, the simple concept of minimizing the difference between the FFT of the original

signal and that of the adversarial signal would allow the adversarial signal to still mirror this new spectral behavior. The more complicated spectral behavior of an OFDM signal (due to requiring orthogonality) further indicates that the spectral integrity component of the loss will need to be more highly prioritized than was done for this work on non-OFDM systems, potentially decreasing the communication and evasion success, but this method should still allow for a properly structured signal in the frequency domain. In summary, the concept of ensuring spectral integrity using the methods introduced in this chapter would still be applicable to OFDM systems.

There are some additional aspects of OFDM that could have an impact on the communications aware attack in general. For example, the idea of perturbing a single sample in the time domain in the current work only affects one symbol and therefore only a few bits of the transmitted bit sequence. However, since the resulting transmitted signal of an OFDM system is essentially a large number of symbols superimposed and sent in parallel, perturbing a signal sample could affect many more bits. Future work should examine how this aspect could change the behavior of how an AMN learns to perturb a signal. This effect would not necessarily be an issue if the attack is carried out on the symbols instead of the samples since the perturbation would be applied before the IFFT combines the modulated symbols. Additionally, OFDM could provide an additional attack vector for the communications aware attack. Since a transmitted signal is comprised of a large number of subcarriers transmitting in parallel, it's possible that some of these could be reserved to aid the attack rather than transmit legitimate data. Doing so would not drastically decrease data rate given a significant number of subcarriers. This work with OFDM would also assume a different eavesdropper that can distinguish modulation schemes when the transmitter is utilizing OFDM. This would likely prove a more difficult process and could require additional knowledge at the eavesdropper, such as number of subcarriers, or a deeper CNN architecture.

4.5 Conclusion and Future Work

This chapter first examined the necessity of forcing the perturbation, and therefore the resulting combined adversarial signal, to be more in-band. A variety of potential solutions were presented. The first investigated perturbing the symbols rather than the samples. This was shown to be successful in both keeping the adversarial signal in-band and maintaining solid communications and evasion abilities; however, in the scenario that the AMN does not have access to the symbols, this is not a feasible approach. A new loss function for training the AMN, coined the spectral deception loss, was therefore presented. This loss looks to force the frequency content of the adversarial signal and perturbation to be more similar to that of a clean signal. A variety of methods were tested. However, an approach that looks to minimize the difference between the FFT of the adversarial signal and that of the original signal was seen to be the most successful. Further, this result was shown to be generalized to all modulation schemes considered in the assumed environment.

While these results show promise, there is still potential future work to develop the concept further. The various deception loss methods presented in this work are intended as starting points and improving upon these may offer greater success. For example, one simple adaptation could come in the form of completing a more exhaustive parameter search over the configurations for the deception loss, such as the δ value used in the Huber loss. Additionally, other functions than the FFT investigated here could be used to determine and quantify the difference between the original signal and the adversarial signal. For example, while the PSD approach was not successful, one that explicitly uses both the PSD and the phase shift could prove fruitful. Finally, while mean absolute error (MAE), mean squared error (MSE), and Huber are good for determining the difference between corresponding elements in an array of data, such as with time domain samples, they may not be the most appropriate for

the frequency domain. Other functions, such as Fréchet distance [79], may provide better comparisons of similarity and should be further studied.

The predominant method used in this work to determine success of the loss was to qualitatively observe if the perturbation was concentrated in the main lobe of the signal. While this may be sufficient in determining whether a human operator can detect the adversarial signal, future work should examine whether this adapted attack framework would be effective in evading detection by a machine learning algorithm aimed at detecting these attacks. Further, the developed methods could also be tested against an eavesdropper that utilizes matched filtering as a preprocessing step to see how this affects the evasion success when compared to previous communications aware attacks that had out-of-band perturbations. Additionally, previous work has assumed oversampling of the signal by the eavesdropper which provides a larger attack vector for the evasion attack in terms of available bandwidth outside of the signal's main lobe. Future work should loosen this assumption in order to better test the success of the deception loss. Recent work has focused on strategies that make the classifier networks more hardened against attacks such as by utilizing curriculum training [80]. Future work should examine the success of evasion attacks against such defensive techniques when employing the deception loss.

Chapter 5

Conclusions

The current work introduced and analyzed important extensions to communications aware attacks in order to provide a methodology for better securing communications against malicious parties. Chapter 2 first described the attack environment considered throughout this work, that of a transmitter attempting to evade modulation classification of a malicious eavesdropper that intercepts the RF communication between the transmitter and its intended receiver. It then illustrated the necessity of communications aware attacks given the limitation of gradient-based methods that don't consider the communication link, and therefore don't provide strong communications reliability, and provided a short discussion on the attack framework introduced in [12]. Chapter 3 presented updates to this attack methodology, namely the current work replaced the previous Adversarial Residual Network (ARN), that outputs just the perturbation, to be a new Adversarial Mutation Network (AMN) that provides the full, combined, adversarial signal. Further, the loss methods used during training were altered, most importantly the power loss that more actively limits the perturbation power and forces the adversarial signal to remain similar to the original signal. It was shown that these updates caused the resulting adversarial signal to appear more benign when examining the frequency content using a power spectral density (PSD) plot. While still containing some out-of-band power, there was roughly a 20 dB in improvement in this region of previous work. In the main focus of Chapter 3, it was seen that when implementing forward error correction (FEC), something that most real-world systems utilize,

that the AMN was more successful in crafting intelligent perturbations. When trained with coded symbols, the communication link was improved during testing, illustrated by drop in bit error rate (BER). Further, the trade-off between evasion success and communication reliability was improved such that decreasing the eavesdropper's accuracy did not degrade communications by as much as in implementations that don't consider FEC. Importantly, this improvement was learned by the AMN inherently, meaning that it learned to make the intelligent perturbations without being provided information about the existence or structure of the FEC coding in use. This allowed the attack framework and AMN architecture to be transferable and successful on other FEC coding schemes, including convolutional codes, and additional modulation schemes without requiring any changes. Beyond being transferable to other modulation schemes and FEC codes, it was also shown that the attack was successfully transferable when executed on eavesdroppers other than the one used when training the AMN. This is important given that the AMN will likely not have access to the malicious eavesdropper's classifier network. Additionally, it was then demonstrated that explicit information about the coding scheme could be provided to the AMN in the form of architectural configurations that would enhance the success of the attack on the eavesdropper. More specifically, the striding and kernel size of the underlying convolutional neural network (CNN) could be changed to match that of the FEC block size. When tested on Hamming (12,8) there was a 25-50% decrease in eavesdropper classification success over Hamming (7,4) using the same architecture optimized for Hamming (12,8).

Chapter 4 then examined the spectral characteristics of the frequency content for communications aware attacks. Results in Chapter 3 and previous work showed that the adversarial signal exhibited significant out-of-band content. This could lead to detection by an observer, violate the spectrum mask allocated to the communication link, or make the attack vulnerable to preprocessing techniques at the eavesdropper. Chapter 4 presented two potential

solutions to this attack limitation. The first considered perturbing the symbols rather than the samples. This was shown to have much better structural behavior as the signal appeared benign and consisted of exclusively in-band frequency content given that the interpolating was done after the perturbation. Further, while the attack exhibited strong structural integrity, there was very little degradation in the BER of evasion success. However, the assumption that the attacker (typically considered to be the transmitter in this work) has access to the symbols may not be feasible in all applications. In some scenarios, the attacker may not be co-located with the transmitter or the attack might need to be inserted at the very end of the transmission process. To address this constraint, a novel spectral deception loss was introduced. The deception loss is a loss function substituted into the AMN training process in place of the previous power loss that forces the adversarial signal to follow the spectral structure of the original, clean signal. It was shown that when implementing a deception loss into training that attempts to minimize the difference between the Fast-Fourier Transform (FFT) of the original signal and the FFT of either the perturbation or adversarial signal, the attack is successful in meeting this goal of an crafting in-band adversarial signal. However, this was at the detriment of communication reliability as the BER increased when spectral integrity was highly prioritized, though it was shown that this trade-off could be adjusted through use of the loss constants. Additionally, it was shown that implementing a deception loss by attempting to minimize the difference between PSDs, the attack was less successful, both for the frequency content and the BER, due to the loss of phase information. The deception loss based on minimizing the difference in the FFTs had success across all modulation schemes assumed in the environment configuration (BPSK, QPSK, 8-PSK, and 16-QAM).

In conclusion, both Chapter 3 and 4 provide important contributions in the field of wireless security. The utilization of FEC showed improvements in the communication reliability

of an evasion attack, ensuring that the attack remains feasible even under more stringent conditions where communication success is of the highest priority. The consideration of spectral integrity and implementation of the spectral deception loss protect the attack from observation and allow it to masquerade as a benign communication link. Through the employment of these new features to the communications aware attack, a transmitter can better protect its communications from a malicious eavesdropper.

The remainder of this chapter covers future work. It is broken into two sections. Section 5.1 discusses assumptions made in this work and how they can be removed in future work. Section 5.2 concludes with a discussion on how the work presented in this thesis can be examined for future improvements.

5.1 Removal of Assumptions

Throughout this work, there were a variety of assumptions knowingly made given the understood attack environment. In future work, some of these assumptions can be peeled back in order to test the introduced concepts under more complex scenarios. The first limitation concerns channel effects. In this work, the two channel effects implemented were a time offset between the eavesdropper and transmitter and an AWGN channel providing noise between the transmitter, receiver, and eavesdropper. While this provides a good proxy for a simplistic communication channel, more work can be done in the future to add additional effects such as carrier frequency offset or Rayleigh fading, as was done in [41]. Additionally, the assumption of synchronization between the transmitter and receiver could be removed as well.

Throughout Chapter 3, the SNR range considered was 5 dB to 15 dB since this incorporated the region where there is a trade-off between the BER of FEC-enabled signals and non-

coded signals. Potential future work could study the behaviors of the AMN when trained with FEC only in a range of SNRs where the FEC provides improvement. Given that in the current work, FEC was not always beneficial for a given SNR, the AMN may have struggled more to craft intelligent perturbations. There may be interesting insights when strictly considering the range of SNR where FEC is beneficial. Additionally, other FEC codes could be considered. This work assumed block codes as an initial step given that their structure should be well-suited for the learning process and architecture of the AMN. While it was shown to work for convolutional codes as well, others should be tried. For instance, the quickly growing standard of 5G communications use polar and low parity density check codes [81]. The framework could be tested with these codes to analyze the AMN's ability to learn more complex codes.

While this work considered an attack on an AMC system deployed at a malicious eavesdropper, this work is intended to be generalized for other RFML applications. For this reason, future work can implement the attack frameworks and methods introduced in Chapters 3 and 4 to other applications such as signal detection or specific emitter identification. In this vein, this work considered an eavesdropper that had previously detected and isolated the signal. Due to this, it considered the full received signal. Removing this assumption may have significant impact on the attack, especially on the work presented in Chapter 4. If the eavesdropper must first detect the signal, the resulting spectral shape from the frequency content may have an impact on what is perceived as the signal. This could both provide more limitations on the attack, such as removing out-of-band perturbations seen in Chapter 3, or provide a larger attack vector as the attacker could try to increase the bandwidth of the signal or trick the eavesdropper into assuming the out-of-band perturbations are part of the signal. For instance, if the side content of the received signal created by the attack is increased in power, the eavesdropper could detect the entire band as the true signal, open-

ing itself up to a more destructive attack. Additionally, this work assumed that the signal was 8 times oversampled. This provides the attack with increased bandwidth in which the perturbation could make use of the added samples without as much detriment to the communication success. Future work could consider lowering the samples per symbol to analyze the effect this has on the success of the attack.

Finally, this work assumed that the AMN had access to knowledge of the CNN used at the eavesdropper in some capacity. While it was shown in Chapter 3 that this attack could be transferred to eavesdropper's employing different CNNs than what was used in training the evasion attack, this still assumed a general understanding of at least the type of network deployed at the eavesdropper. Future work could continue to look more into how the attack fares against CNNs that are significantly different in architecture and data trained on or how successful the attack is when the eavesdropper employs a completely different method of classification than the attack was trained on, such as an RNN.

5.2 Expansion on Current Work

Throughout this work, the evasion attack considered an untargeted scenario as an initial step. However, in Chapter 2, it was shown that one issue with gradient-based methods were that they required an extremely large perturbation when performing targeted attacks such that the initial signal is destroyed. The communications aware framework was introduced as a solution to this problem. Given that it has shown to be successful in untargeted attacks, future work should expand it to perform targeted attacks.

One issue with the communication loss metric was that calculating BER is not differentiable given the hard decision process. Instead, the error vector magnitude of the clean and perturbed signals was used as a proxy. Future work could consider other methods such as soft

demodulation and soft decoding of the FEC that may provide a better way for the AMN to learn to provide effective communications given that the communications loss would be more of a true representation of the success in transmitting information.

Two additional extensions for Chapter 3 could be furthered with future work. The first is for explicit learning of coding discussed in the latter half of the chapter. The work shown in this thesis was provided as an initial examination into this process. More work should be done to find optimal architectures and configurations to best make use of the explicit information on the coding scheme. Specifically the number of channels in the added convolutional layer should be increased to allow for more dimensionality and therefore more flexibility in crafting a successful adversarial signal. Additionally, the loss functions could be re-implemented such that they too provide information on the coding structure employed. The second component left up to future work is interleaving. Just like FEC, interleaving is an aspect in RF communications that is utilized in a large portion of systems. By introducing this into the training process, the AMN may learn to utilize bursty perturbations or structure the perturbation in a more efficient way to help improve communications while maintaining the same level of evasion success.

For Chapter 4, future work could focus on updates to the spectral deception loss. Specifically, while great for point-to-point (such as in the time domain) differences, loss functions such as mean squared error and Huber may not be suited well for calculating difference in the frequency domain. Other functions such as Fréchet distance [79] could instead be considered. While the PSD method for deception loss was seen to be unsuccessful due to loss of phase information, a deception loss is more directly based off both the PSD and the phase may be more successful than using the FFT. Another route for study could be implementing the communications aware attack on OFDM systems. This could potentially add some additional attack vectors that make use of the large number of orthogonal subcarriers and

would help ensure that the attack can be implemented on real world applications that use OFDM. Finally, since the process of filtering an adversarial signal with a matched filter can be done in a differentiable way using a convolution with the filter taps, the AMN may be able to learn the filtering process. Future work could examine removing the deception loss and instead training with an eavesdropper that is actively filtering the signal to see if the AMN learns to place the perturbation in-band implicitly.

Bibliography

[1] Q. Rao and J. Frtunikj, "Deep learning for self-driving cars: Chances and challenges," in *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, pp. 35–38, 2018.

[2] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for contiki-ng-based iot networks exposed to nsl-kdd dataset," *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2020.

[3] M. Chale, N. Bastian, and J. Weir, "Algorithm selection framework for cyber attack detection," *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2020.

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

[5] Sivic and Zisserman, "Video google: a text retrieval approach to object matching in videos," in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1470–1477 vol.2, 2003.

[6] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[7] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion at-

tacks in the context of wireless communications," *IEEE Trans. on Info. Forensics and Security*, 2019.

[8] 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," *IEEE Military Commun. Conf. (MILCOM)*, 2018.

[9] S. Gecgel, C. Goztepe, and G. K. Kurt, "Jammer detection based on artificial neural networks: A measurement study," *ACM Workshop on Wireless Security and Machine Learning (WiseML 2019)*, 2019.

[10] Z. Langford, L. Eisenbeiser, and M. Vondal, "Robust signal classification using siamese networks," *ACM Workshop on Wireless Security and Machine Learning (WiseML 2019)*, 2019.

[11] N. Abuzainab, T. Erpek, K. Davaslioglu, Y. E. Sagduyu, Y. Shi, S. J. Mackey, M. Patel, F. Panettieri, M. A. Qureshi, V. Isler, and A. Yener, "Qos and jamming-aware wireless networking using deep reinforcement learning," *IEEE Military Commun. Conf. (MILCOM)*, 2019.

[12] B. Flowers, R. M. Buehrer, and W. C. Headley, "Communications aware adversarial residual networks for over the air evasion attacks," *IEEE Military Commun. Conf. (MILCOM)*, 2019.

[13] R. Politanskyi and M. Klymash, "Application of artificial intelligence in cognitive radio for planning distribution of frequency channels," in *2019 3rd International Conference on Advanced Information and Communications Technologies (AICT)*, pp. 390–394, 2019.

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

[15] S. Ravindran and R. Jose, "Direction of arrival and channel estimation using machine learning for multiple input multiple output system," in *2019 International Conference on Communication and Electronics Systems (ICCES)*, pp. 1327–1330, 2019.

[16] A. Khan, S. Wang, and Z. Zhu, "Angle-of-arrival estimation using an adaptive machine learning framework," *IEEE Communications Letters*, vol. 23, no. 2, pp. 294–297, 2019.

[17] N. Rastegardoost and B. Jabbari, "A machine learning algorithm for unlicensed lte and wifi spectrum sharing," in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–6, 2018.

[18] J. Kim and J. P. Choi, "Sensing coverage-based cooperative spectrum detection in cognitive radio networks," *IEEE Sensors Journal*, vol. 19, no. 13, pp. 5325–5332, 2019.

[19] M. Bari, A. Khawar, M. Doroslovacki, and T. C. Clancy, "Recognizing fm, bpsk and 16-qam using supervised and unsupervised learning techniques," *49th Asilomar Conf. on Signals, Systems and Computers*, 2016.

[20] N. E. West and T. J. O'Shea, "Deep architectures for modulation recognition," *IEEE Int. Symp. on Dynamic Spectrum Access Networks (DySPAN)*, 2017.

[21] K. Karra, S. Kuzdeba, and J. Petersen, "Modulation recognition using hierarchical deep neural networks," *IEEE Int. Symp. on Dynamic Spectrum Access Networks (DySPAN)*, 2017.

[22] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," *Commun. in Computer and Info. Science*, vol. 629, 2016.

[23] J. L. Ziegler, R. T. Arn, and W. Chambers, "Modulation recognition with gnu radio, keras, and hackrf," *IEEE Int. Symp. on Dynamic Spectrum Access Networks (DySPAN)*, 2017.

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

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

[26] W. C. Headley, J. D. Reed, and C. R. C. M. da Silva, "Distributed cyclic spectrum feature-based modulation classification," *IEEE Wireless Commun. and Netw. Conf.*, 2008.

[27] D. T. Kawamoto and R. W. McGwier, "Rigorous moment-based automatic modulation classification," vol. 1, 2016.

[28] A. Hazza, M. Shoaib, S. A. Alshebeili, and A. Fahad, "An overview of feature-based methods for digital modulation classification," *1st Int. Conf. on Commun., Signal Processing, and their Applications (ICCSPA)*, 2013.

[29] M. M. T. Abdelreheem and M. O. Helmi, "Digital modulation classification through time and frequency domain features using neural networks," *IX Int. Symp. on Telecommunications (BIHTEL)*, 2013.

[30] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar, "Adversarial machine learning," *ACM Workshop on Security and Artificial Intelligence*, pp. 43 – 58, 2011.

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

[32] 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 Symp. on Security and Privacy (EuroS&P)*, 2016.

[33] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *Int. Conf. on Learning Representations*, 2017.

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

[35] S. Baluja and I.Fischer, "Learning to attack: Adversarial transformation networks," *Proc. of AAAI-2018*, 2018.

[36] S. Kokalj-Filipovic and R. Miller, "Adversarial examples in rf deep learning:detection of the attack and its physical robustness," *arXiv preprint arXiv:1902.06044*, 2019.

[37] 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.

[38] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 36–52, 2018.

[39] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "How to make 5g communications "invisible": Adversarial machine learning for wireless privacy," *arXiv preprint arXiv:2005.07675*, 2020.

[40] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Channel-aware adversarial attacks against deep learning-based wireless signal classifiers," *arXiv preprint arXiv:2005.05321*, 2020.

[41] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels," *Conf. on Info. Sciences and Systems (CISS)*, 2020.

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

[43] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *arXiv preprint arXiv:1807.02567*, 2018.

[44] Y. Shi, K. Davaslioglu, and Y. Sagduyu, "Over-the-air membership inference attacks as privacy threats for deep learning-based wireless signal classifiers," *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2020.

[45] C. Cardoso, A. R. Castro, and A. Klautau, "An efficient fpga ip core for automatic modulation classification," *IEEE Embedded Systems Letters*, vol. 5, no. 3, pp. 42–45, 2013.

[46] H. Rahbari and M. Krunz, "Full frame encryption and modulation obfuscation using channel-independent preamble identifier," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2732–2747, 2016.

[47] M. DelVecchio, B. Flowers, and W. C. Headley, "Effects of forward error correction on communications aware evasion attacks," *IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Comm. (PIMRC)*, 2020.

[48] S. Bair, M. DelVecchio, B. Flowers, A. J. Michaels, and W. C. Headley, "On the limitations of targeted adversarial evasion attacks against deep learning enabled modulation recognition," *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2019.

[49] M. DelVecchio, V. Arndorfer, and W. C. Headley, "Investigating a spectral deception loss metric for training machine learning-based evasion attacks," *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2020.

[50] K. Morita, A. Tashita, M. Nii, and S. Kobashi, "Computer-aided diagnosis system for rheumatoid arthritis using machine learning," in *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 2, pp. 357–360, 2017.

[51] E. A. Bayrak, P. Kırcı, and T. Ensari, "Comparison of machine learning methods for breast cancer diagnosis," in *2019 Scientific Meeting on Electrical-Electronics Biomedical Engineering and Computer Science (EBBT)*, pp. 1–3, 2019.

[52] B. Bektaş and S. Babur, "Machine learning based performance development for diagnosis of breast cancer," in *2016 Medical Technologies National Congress (TIPTEKNO)*, pp. 1–4, 2016.

[53] J. Manasa, R. Gupta, and N. S. Narahari, "Machine learning based predicting house prices using regression techniques," in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 624–630, 2020.

[54] F. Wang, Y. Zou, H. Zhang, and H. Shi, "House price prediction approach based on deep learning and arima model," in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 303–307, 2019.

[55] C. E. Thornton, R. M. Buehrer, A. F. Martone, and K. D. Sherbondy, "Experimental analysis of reinforcement learning techniques for spectrum sharing radar," in *2020 IEEE International Radar Conference (RADAR)*, pp. 67–72, 2020.

[56] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2087–2091, 2017.

[57] D. Hong, Z. Zhang, and X. Xu, "Automatic modulation classification using recurrent

neural networks," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 695–700, 2017.

[58] S. Hu, Y. Pei, P. P. Liang, and Y. Liang, "Robust modulation classification under uncertain noise condition using recurrent neural network," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, 2018.

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

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

[61] P. D. White, R. M. Buehrer, and W. C. Headley, "Fhss signal separation using constrained clustering," in *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, pp. 159–164, 2019.

[62] H. Franco, C. Cobo-Kroenke, S. Welch, and M. Graciarena, "Wideband spectral monitoring using deep learning," *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2020.

[63] A. T. Vo, H. S. Tran, and T. H. Le, "Advertisement image classification using convolutional neural network," in *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 197–202, 2017.

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

[65] 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 *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 19–35, 2018.

[66] H. Kwon, H. Yoon, and K. Park, "Selective poisoning attack on deep neural network to induce fine-grained recognition error," in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 136–139, 2019.

[67] Cheng Soon Ong, "Towards open machine learning: Mloss.org and mldata.org," in *2011 IEEE International Workshop on Open-source Software for Scientific Computation*, pp. 12–12, 2011.

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

[69] K. Davaslioglu and Y. E. Sagduyu, "Trojan attacks on wireless signal classification with adversarial machine learning," *IEEE Int. Symp. on Dynamic Spectrum Access Networks (DySPAN)*, 2019.

[70] 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.

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

[72] H. Zhang, Q. Wang, X. Luo, Y. Yin, Y. Chen, Z. Cui, and Q. Zhou, "A user-adaptive deep machine learning method for handwritten digit recognition," in *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, pp. 108–111, 2018.

[73] T. D. Vo-Huu and G. Noubir, "Mitigating rate attacks through crypto-coded modulation," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '15, (New York, NY, USA), p. 237–246, Association for Computing Machinery, 2015.

[74] M. Z. Hameed, A. Gyorgy, and D. Gunduz, "Communication without interception: Defense against modulation detection," *IEEE Global Conf. on Signal and Info. Processing (GlobalSIP)*, 2019.

[75] Freescale Semiconductor, Inc., "Implementing data whitening and crc verification in software in kinetis kw01 microcontrollers," *https://www.nxp.com/docs/en/application-note/AN5070.pdf*.

[76] J. D. Gaeddert, "Liquiddsp dataset," *https://github.com/jgaeddert/liquid-dsp/*.

[77] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Int. Conf. on Learning Representations*, 2015.

[78] N. D. Tripathi and J. H. Reed, *Cellular Communications: A Comprehensive and Practical Guide*. Wiley, 2014.

[79] P. Chen, G. Li, K. Xu, and J. Wan, "Applying the frechet distance to the specific emitter identification," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pp. 1027–1030, 2016.

[80] M. Z. Hameed, A. Gyorgy, and D. Gunduz, "The best defense is a good offense: Adversarial attacks to avoid modulation detection," 2019.

[81] Rohde and Schwartz, "White paper: Radio fundamentals for cellular networks," *https://www.mobilewirelesstesting.com*, 2018.