Adversarial Machine Learning for Enhanced Spread Spectrum Communications

Mohamed K. M. Fadul and Donald R. Reising
Electrical Engineering Department
University of Tennessee at Chattanooga
{mohammed-fadul, donald-reising}@utc.edu

K.T. Arasu and Michael R. Clark
Open Innovation Center
Riverside Research
{karasu, mclark}@riversideresearch.org

MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM) | 978-1-6654-3956-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/MILCOM52596.2021.9652911

***Abstract*—Recently deep learning has demonstrated much success within the fields of image and natural language processing, facial recognition, and computer vision. The success is attributed to large, accessible databases and deep learning's ability to learn highly accurate models. Thus, deep learning is being investigated as a viable end-to-end approach to digital communications design. This work investigates the use of adversarial deep learning to ensure that a radio can communicate covertly, via Direct Sequence Spread Spectrum (DSSS), with another while a third (the adversary) is actively attempting to detect, intercept and exploit their communications. The adversary's ability to detect and exploit the DSSS signals is hindered by: (i) generating a set of spreading codes that are balanced and result in low side lobes as well as (ii) actively adapting the encoding scheme. Lastly, DSSS communications performance is assessed using energy constrained devices to accurately portray IoT and IoBT device limitations.**

***Index Terms*—Deep Learning, DSSS, Adversarial Learning, IoT, Internet of Battlefield Things (IoBT)**

I. Introduction

Deep Learning (DL) has demonstrated success within the fields of image and natural language processing, facial recognition, as well as computer vision. This success is due in large part to the presence of and access to large training databases (e.g., MNIST, COCO, & KITTI databases) and DL's ability to learn models capable of achieving high levels of accuracy beyond those of human derived mathematical models. Over the past few years, DL has been put forward as a viable solution for many challenges facing modern communication systems such as: spectrum management [1], [2], modulation and radio identification [3]–[6], as well as design [3], [7]–[9]. The focus herein is on the use of DL for communications system design.

It has been asserted that Artificial Intelligence (AI) and its subset DL is critical to future communications system design and operation due to ever increasing complexity [9]. This complexity is attributed to the inherited engineering trade-offs that were made during the design and implementation of previous communication systems as well as impairments (e.g., the channel, radio hardware, interference) that exist during operations. The resulting technological complexity must be navigated and overcome during the development and fielding of new communications systems. Although not a new challenge, this complexity is exacerbated by: (i) the explosive proliferation of wireless communication systems due to increasing Internet of Things (IoT) and Internet of Battlefield Things (IoBT) deployments [10]–[13], (ii) the employment of AI within military and other radio applications (e.g., cognitive radio) [9], [11], as well as (iii) increasing threats within the electromagnetic spectrum [14]. Such complexities can overwhelm traditional communications system design approaches that rely upon tractable mathematical models, especially when such models are difficult to create or missing altogether. However, DL has shown to be a viable alternative in such cases, because it thrives as more information becomes available. An additional benefit to DL is its flexibility, which is enabled by the relative ease at which it can be re-trained in near real-time when environmental conditions change. Such flexibility is lacking within human developed communications systems that require months to achieve an equivalent re-design [9].

The published works [3], [7]–[9], [15]–[17] demonstrate DL driven communications. In comparison to conventional approaches, these works show that DL achieves superior performance for: (i) maximizing capacity when multiple radios communicate over a common channel, (ii) integration of expert knowledge (e.g., channel equalization) within the DL model, (iii) modulation recognition, (iv) signal compression, (v) signal detection, and (vi) adapting transmitter impairments (e.g., the channel and RF front-end) [3], [7]–[9]. The works in [15]–[17] are of particular interest to the work presented herein due to their contributions within the area of DL driven: DSSS, communications security, or the combination thereof. In [16], Shakeel presents two DL approaches that improve DSSS system security by generating communications signals devoid of certain well-understood mathematical features and without secret information. Wei et al. [15] studies blind estimation of a DSSS system's length 31 Pseudo-Random Noise (PRN) sequences and motivates our use of adversarial training. The research presented in [17] teaches Neural Networks (NNs) to communicate securely using a secret key.

In this work, we leverage DL to implement a Generative Adversarial Network (GAN) inspired approach to ensure that a transmitter (a.k.a., Alice) and receiver (a.k.a., Bob) can communicate—using Direct Sequence Spread Spectrum (DSSS)—while actively inhibiting an adversary's (a.k.a., Eve's) ability to detect and reconstruct the intercepted messages' information. Our contributions are:

- A special set of 15 bit "select" spreading codes are generated

to adhere to balance and low side lobe stipulations that are unachievable using Gold, Kasami, and Weil code generation techniques. These "select" spreading codes improve DL model convergence during training.
- The introduction of spreading codes with the messages to enhance DL driven featureless DSSS signaling, which improves jamming resiliency, and Low Probability of Detect/Intercept (LPD/LPI) performance beyond that in [16].
- The use of adversarial training and a learned, shared-encryption scheme that improves DSSS system security.
- The use of multiple spreading codes instead of one shared code to improve the system's LPD.
- The discovery that low Peak Side Lobe (PSL) spreading sequences improve model convergence.
- The transmitter energy is constrained to more accurately capture the resource limitations that exist within many IoT and IoBT applications, which contradicts the work in [16].

The paper is organized as follows: Sect. II describes the adversary, Sect. III covers spreading code generation as well as the DL architecture and training, experimental results are presented in Sect. IV, and Sect. V concludes the paper.

II. Threat Model

This work adopts a threat model that is inspired by the works in [17]–[21]. Eve is categorized as an on-path adversary because it is the most capable. Eve possesses the ability to observe and transmit in real-time, which allows it to spoof, inject, remove, and alter traffic. Therefore, it is assumed that Eve has access to or knowledge of the software applications needed to modify its own transceiver settings and the computational resources required to carry out its attack [19]. We also assume that Eve makes use of the same DL architecture as Alice and Bob, thus making Eve a peer of Alice and Bob in terms of capability. Eve is not an authorized user within the targeted DSSS communications system, thus Eve does not have inherent access to the spreading codes nor any of the radios and support devices that form the communications network. Lastly, it is assumed that the hardware and links comprising the DSSS communications network are not initially compromised.

Of particular interest is Eve's ability to continuously eavesdrop on the communications between Alice and Bob with the goal of "learning" the spreading code or codes through non-cooperative, direct, and persistent detection and interception of the DSSS signal. If Eve is able to learn the spreading codes, then Eve will be able to access the transmitted message [20].

III. Methodology

A. PRN Sequence Generation

DSSS is a spread spectrum technique, wherein the original data signal is multiplied with a PRN spreading code [22]. PRN is enabled by perfect periodic auto-correlation sequences. The celebrated waveform, known as m-sequence, is a type of pseudo-random bit sequence generated using maximal linear feedback shift register. Popular spreading codes are Gold codes [23], the 1023-length of which, are used for the Global Positioning System (GPS) Coarse/Acquisition (C/A) signal. Gold codes are generated using two m-sequences (a.k.a., preferred pair). A preferred pair of m-sequences of length L generates a family of $(L+2)$ Gold codes of length L. A relatively newer family of spreading sequences are Weil codes [24], [25], which exist for any prime length sequence. For length-p sequences, the corresponding family of Weil codes has size K=$0.5(p$ - $1)$. The GPS civilian-use signal uses Weil codes. Kasami codes are generated using an m-sequences of length 2^N – 1 where N is an even positive integer, yielding $2^{(N/2)}$ codes whose cross-correlations meet the Welch bound [26], [27]. All these aforementioned sequences are generated using finite fields, primitive polynomials and trace functions [28]. Our AI inspired study requires several hundreds of spreading sequences of a very modest length of 15 bits, which is consistent with the works [15], [17]. These 15 bit codes must satisfy the two stipulations of: (i) being "balanced" and (ii) possessing "low side lobes". Good balance requires the number of ones and zeros to be nearly equal. A simple linear function maps x to $2x-1$, which sends 0 to -1 and 1 to 1. We use this mapping to deal with bi-polar sequences (with entries 1 and -1) instead of binary sequences. For our 15 bit codes, balanced spreading codes are comprised of exactly eight 1's and seven -1's.

The largest absolute value of all side lobes is denoted as the PSL. Low PSL is extremely important when selecting the spreading sequences for training the Neural Network (NN) because low PSL improves model convergence. Gold, Weil and Kasami codes are incapable of generating hundreds of 15 bit codes that meet the required stipulations. Thus, we generate the required number of L=15 spreading codes using a brute-force approach while stipulating "balance" and "low PSL" requirements. There are only thirty perfect sequences (i.e., the PSL equals 1). When the sequences adhere to the balanced stipulation, none of them satisfy PSL equal to 3, however when the PSL is equal to 5 a total of 3,270 spreading sequences can be generated.

B. Model Architecture and Training

The multi-agent system considered in this work is comprised of three parties: Alice, Bob, and Eve. Fig. 1 shows a block diagram of the considered communication scenario. Alice tries to send the message to Bob correctly using a spreading code from a codebook that is available to both Alice and Bob. As the adversary, Eve is attempting to detect,

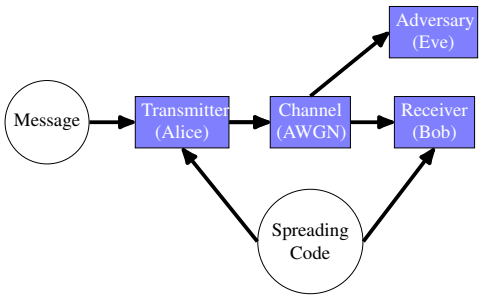

Fig. 1. Block diagram of the multi-agent DSSS system.

intercept, and exploit the communications between Alice and Bob with the explicit goal of recovering the message sent by Alice. Although Eve does not have access to the codebook, Eve attempts to learn the spreading code used by Alice and Bob without prior knowledge of the specific spreading code used nor those remaining in the codebook. An AWGN channel model is used with zero mean and βI co-variance matrix where β depends on the channel signal-to-noise ratio (SNR).

The output of Alice, Bob, and Eve are respectively denoted as $A(\theta_A, M, K)$, $B(\theta_B, Y, K)$, and $E(\theta_E, Y)$ where θ_A, θ_B, and θ_E are NN parameters, M is the original message, Y is the output of the channel, and K is the spreading code. To minimize Eve's reconstruction error, we use the loss function,

$$L_E(\theta_A, \theta_E) = \mathbf{E}_{M,K}(d\{M, E[\theta_E, A(\theta_A, M, K)]\}), \quad (1)$$

$$d(M, M') = \sum_{i=1}^{N} |M_i - M_i'|, \quad (2)$$

where $\mathbf{E}_{M,K}$ is the expected value over the distribution of M and K [17]. Eve's optimum parameters are determined by minimizing its loss function with respect to Alice as follows,

$$O_E(\theta_A) = \arg\min_{\theta_E} \left[L_E(\theta_A, \theta_E)\right], \quad (3)$$

and the loss function for Bob is given by,

$$L_B(\theta_A, \theta_B) = \mathbf{E}_{M,K}\{d[M, B(\theta_B, A(\theta_A, M, K), K)]\}. \quad (4)$$

The whole system is treated as a Multiple-Input and Multiple-Output (MIMO) NN. Using the combined loss function given in (5), the system—a.k.a., Alice, Bob & Eve—is jointly trained with the purpose of minimizing Bob's reconstruction error while increasing Eve's.

$$L_{AB}(\theta_A, \theta_B) = L_B(\theta_A, \theta_B) - L_E\left[\theta_A, O_E(\theta_A)\right] \quad (5)$$

The subtraction in (5) shows that the optimization of Bob's reconstruction error is working against Eve's. Similar to Eve, the optimum parameters for Alice and Bob are determined by,

$$(O_A, O_B) = \arg\min_{\theta_A, \theta_B} \left[L_{AB}(\theta_A, \theta_B)\right]. \quad (6)$$

Alice and Bob are initialized using random parameters and iterative training conducted. The model is then trained for Eve until the optimum parameters O_E are computed. Finally, the model is trained for Alice and Bob using the combined loss function L_{AB} to find the optimum parameters (O_A, O_B) that enable Alice and Bob to defeat the best version of Eve.

The architecture of Alice, Bob, and Eve is detailed in Table I. The architectures for Bob and Eve are similar except the dimension of the first layer. The 1D convolutional layer's dimension is given by: number of filters $\times$ size of the filter $\times$ stride. The size of the message M is 16 bits with entries of either a -1 or 1. Alice's batch normalization layer is used to apply a transmission energy constraint. The work in [16] did not apply energy constraints, which can significantly impact system performance and assessment results by allowing individual entries within the spread signal to exceed a magnitude

TABLE I
THE CONFIGURATION AND VALUES FOR THE NEURAL NETWORKS ASSOCIATED WITH THE GAN INSPIRED ARCHITECTURE USED REPRESENT THE TRANSMITTER (ALICE), RECEIVER (BOB), AND ADVERSARY (EVE).

Agent	Layer	Dimension	Activation
Alice	Concatenate	16 + 15 = 31	none
	Dense	256	relu
	Lambda / Expand Dimension	none	none
	Convolution 1D	2 X 4 X 1	sigmoid
	Convolution 1D	4 X 2 X 2	sigmoid
	Convolution 1D	4 X 1 X 1	sigmoid
	Convolution 1D	1 X 1 X 1	tanh
	Lambda / squeeze Dimension	none	none
	Dense	L	linear
	Power Normalization	L	none
Bob	Concatenate	L + 15	none
	Dense	256	relu
	Lambda / Expand Dimension	none	none
	Convolution 1D	2 X 4 X 1	sigmoid
	Convolution 1D	4 X 2 X 2	sigmoid
	Convolution 1D	4 X 1 X 1	sigmoid
	Convolution 1D	1 X 1 X 1	tanh
	Lambda / squeeze Dimension	none	none
	Dense	16	linear
Eve	Input	L	none
	Dense	256	relu
	Lambda / Expand Dimension	none	none
	Convolution 1D	2 X 4 X 1	sigmoid
	Convolution 1D	4 X 2 X 2	sigmoid
	Convolution 1D	4 X 1 X 1	sigmoid
	Convolution 1D	1 X 1 X 1	tanh
	Lambda / squeeze Dimension	none	none
	Dense	16	linear

of 1. Such a result can bias Bob and Eve to favor these entries during training and message recovery. Thus, a transmission energy constraint of,

$$||\mathbf{x}||_2^2 \leq n, \quad (7)$$

is adopted, where $\mathbf{x}$ is the spread signal at the output of Alice and n is the number of bits in $\mathbf{x}$ [3]. This constraint ensures that the magnitude of each entry in $\mathbf{x}$ does not exceed 1. The size of the spreading code K is set to 15 bits and chosen if it has the properties described in Sect. III-A. Both the original message, M, and spreading code, K, are assumed to be uniformly distributed.

IV. EXPERIMENTAL RESULTS

A. Assessment of Low Probability of Detection

This experiment assesses the LPD nature of the DSSS communications system shown in Fig. 1. Our work integrates Auto-Correlation-based Detection (ACD) to provide Eve with a method by which to detect the DSSS signals sent by Alice. Selection of the ACD detector is motivated by its superior low SNR performance with the lowest computation [29]. In accordance with our threat model, Sect. II, we assume that Eve is continuously sensing the channel and applying the ACD method periodically. Detection performance is measured using probability of detection, P_d. If the input signal at Bob and Eve is $s[n]$, then the Power Spectral Density (PSD) of $s[n]$ is,

$$\tilde{P}[k] = \frac{1}{N_s}|S[k]|^2, \quad (8)$$

where $S[k]$ is the Fourier transform of $s[n]$, and N_s is the number of samples comprising $s[n]$. After calculating the PSD, $\tilde{P}[k]$, the auto-correlation function is computed,

$$r[m] = \frac{1}{N_s} \sum_{k=0}^{N_s-1} \tilde{P}[k] \exp\left(\frac{i2\pi km}{N_s}\right), \tag{9}$$

where $m{=}0, 1, \ldots, N_s - 1$ [29]. Using the auto-correlation, the decision variable is determined,

$$D_s = \frac{\frac{1}{\Upsilon} \sum_{l=1}^{\Upsilon} |r[l]|^2}{\frac{1}{\eta - \Upsilon} \sum_{j=1}^{\eta-\Upsilon} |r[j]|^2}, \tag{10}$$

where η is the length of the auto-correlation sequence, Υ is the number of auto-correlation peaks above τ_r, which is,

$$\tau_r = \frac{1}{2} \max\ r[m], \tag{11}$$

where $r[j]{=}r[m] \geq \tau_r$, and $r[l]{=}r[m]{<}\tau_r$. DSSS signal detection is conducted using two hypotheses,

$$\mathcal{H}_0 : D_s < \lambda, \tag{12}$$

$$\mathcal{H}_1 : D_s \geq \lambda, \tag{13}$$

where the null hypothesis, $\mathcal{H}_0$, represents the case of "no signal detected" and the alternative hypothesis, $\mathcal{H}_1$, is the case of "signal detected". The detection decision threshold is,

$$\lambda = \left| \sqrt{\frac{2(\sigma^2)^2}{2N_s + 1} \log(P_f)} \right|, \tag{14}$$

where σ is the variance of $s[n]$ and $P_f{=}0.1$ is the threshold probability of false alarm.

The network in Fig. 1 is trained for three different LPD assessments: *Scenario #1:* A spreading code is randomly chosen from a uniformly distributed set of spreading codes and used to spread 25,000 messages, *Scenario #2:* A spreading code is randomly chosen from the 3,270 set of "select codes" and used to spread 25,000 messages, and *Scenario #3:* A total of 2,638 spreading codes are chosen from the 3,270 set of "select codes" and each used to spread 10 messages.

The first two scenarios are assessed using a blind test set of spread signals constructed using 100,000 messages and the spreading code chosen for the selected LPD assessment scenario. These spread signals are not part of the spread signal set used to train either of the networks. As with the first two scenarios, the third scenario is also assessed using blind testing. The set of spread signals, used to test the third scenario, are generated by selecting 600 spreading codes and using each of them to spread a total of 100 messages, thus this test set is comprised of 60,000 spread signals. It is important to note that the 600 spreading codes, used for testing, are not part of the 2,638 spreading codes used to generate the training signals set. Every spread signal passes through the channel layer, which adds scaled AWGN to achieve SNRs from -21 dB up to 0 dB in steps of 3 dB. These noisy spread signals are the inputs to the ACD detector. For each SNR, percent detection, P_d, is calculated by dividing the number of times a spread signal is correctly detected by the total number of spread signals transmitted by Alice. Fig. 2 shows percent detection performance for each of the three LPD assessment scenarios when Eve employs an ACD detector. For scenario #1 and #2, Eve is able to detect the presence of the corresponding DSSS signals with accuracy of 75% or higher at SNRs of -12 dB or higher. For scenario #3, Eve is able to detect the presence of the DSSS signal with an accuracy of 70%, and no higher, at SNRs of -6 dB or higher. Although the LPD nature of the spread signals of scenario #3 is better than that of the other two scenarios, Eve is still capable of determining that a DSSS signal is present within the channel. Thus, Alice and Bob need to take a more proactive approach to ensure that Eve is not able to exploit the detected DSSS signals.

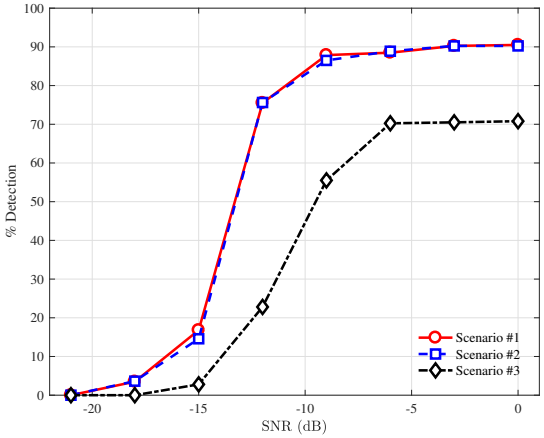

Fig. 2. Eve's average percent detection, P_d, performance for each of the three LPD assessment scenarios: one spreading code drawn from a uniform distribution (solid red line, ○), one spreading code drawn from the set of 3,270 "select" codes (dashed blue line, □), and 600 "select" spreading codes (dashed black line, ◇) generated using an ACD detector at SNRs from -21 dB to 0 dB in 3 dB increments.

B. DSSS Training Evolution

This experiment depicts the training evolution of Alice, Bob, and Eve in the adversarial scenario described in Sect. III and illustrated in Fig. 1. The network architecture shown in Fig. 1 and Table I is implemented in Tensorflow 2.0. The architectures of Alice, Bob, and Eve are similar with the exception for the dimension of their first layers. Alice's first layer accepts a 31 bit input that is constructed by concatenating the $M{=}16$ bit message with the $K{=}15$ bit spreading code. Bob's network accepts a 271 bit length input, which is Alice's $L{=}256$ bit DSSS signal concatenated with the $K{=}15$ bit spreading code. Eve's network accepts a $L{=}256$ input, because Eve *only* has access to Alice's transmitted DSSS signal.

An alternating approach is used to train the networks representing Alice, Bob, and Eve. In this approach, Alice and Bob are trained for one mini-batch, and Eve is trained for two mini-batches per training iteration. The mini-batch size is set to 4,096 to speed up the computations. The networks in Fig. 1 are trained using the Tensorflow Adam optimizer

using a learning rate of 800×10^{-6}. For each training step, Eve's loss function is calculated using the $\ell 1$-norm in equation (1). Bob's loss function consists of two components: (i) the reconstruction error that is computed using the $\ell 1$-norm and (ii) a value based upon Eve's reconstruction error,

$$\left(\frac{M}{2} - \varepsilon_E\right)^2 \left(\frac{M}{2}\right)^{-2}, \tag{15}$$

where ε_E is Eve's reconstruction error. Equation (15) returns zero whenever ε_E=8. The loss function, employed by Bob, is configured to reduce Eve's probability of correctly reconstructing the message to no better than a random guess.

Two spreading code generation cases are used to enable comparative assessment: (i) 3,238 spreading codes are randomly chosen from a uniformly distributed code set and (ii) 3,238 spreading codes are randomly chosen from the "select" set of 3,270 codes that are generated to satisfy the balance and low side lobe stipulations. For each case, the chosen spreading codes are divided into training and testing sets comprised of 2,638 and 600 spreading codes, respectively. Alice generates DSSS signals by using every code, in the training and test sets, to spread ten M=16 bit messages that are randomly drawn from a uniform distribution. Training performance is measured by counting the number of bit errors present in the reconstructed messages output by Bob and Eve at the end of every training step. The total number of training steps is,

$$N_T = \frac{N_e}{N_b}, \tag{16}$$

where N_e is the total size of the training set for each assessment case and N_b is the size of the mini-batch.

Bob's and Eve's reconstruction error is shown in Fig. 3 for both spreading code generation cases after 1,000 epochs and at an SNR of 9 dB. Bob's reconstruction error is designated as ε_U^B and ε_S^B for the uniform and "select" spreading codes, respectively. Similar to Bob, Eve's reconstruction error is denoted as ε_U^E for the uniform spreading codes and ε_S^E for the "select" spreading codes. In the beginning of the assessment (i.e., less than 25 epochs), Bob's and Eve's reconstruction error decreases for both cases, thus showing Eve's ability to "break" the basic encoding used by Alice and Bob. However, Alice and Bob learn an alternative encoding scheme that depends more upon the spreading code, that is unavailable to Eve, after 100 epochs for the uniformly generated spreading codes and 30 epochs for the "select" spreading codes. For these assessments, Bob's and Eve's models are converging as the number of bit errors within the reconstructed messages approach zero and $M/2$=8, respectively. An error threshold of 0.05 for Bob and $M/2 \pm 2$ for Eve is used to confirm convergence of their respective models [30]. Alice and Bob's use of the "select" spreading codes result in a faster convergence, approximately 75 epochs faster when compared to the uniform spreading codes, of Bob's ability and Eve's inability to reconstruct the messages sent by Alice, Fig. 3. Additionally, the "select" spreading codes resulted in model convergence for all 15 runs versus 11 of 15 when using uniform spreading codes.

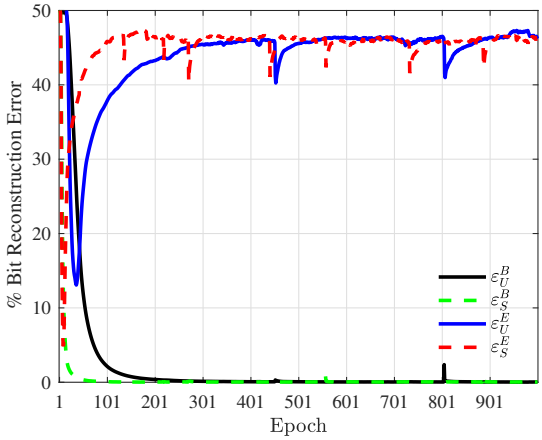

Fig. 3. Reconstruction error after 1,000 epochs for both spreading code assessment cases when: the message is comprised of M=16 bits, the spreading codes are K=15 bits in length, and the DSSS signals are of length L=256 at an SNR of 9 db. Bob's reconstruction error is designated as ε_U^B and ε_S^B for the uniformly and "select" generated spreading codes respectively. Similar to Bob, Eve's reconstruction error is designated as ε_U^E for the uniform spreading codes and ε_S^E for the "select" spreading codes.

C. Block Error Rate

For this experiment, Bob's network is trained using the AWGN channel in Fig. 1 at an SNR of 9 dB and DSSS signals, of length L=[32, 64, 128, 256], that are generated using the "select" spreading codes. Performance is measured using Block Error Rate (BLER) in which the message is considered "successfully reconstructed" if the message bits, recovered by Bob, are the same as the corresponding bits within the original message sent by Alice. Fig. 4 shows Bob's BLER performance at SNR∈[0, 30] dB in 3 dB steps for each of the four investigated spread signal lengths. It can be seen that as L increases Bob's BLER decreases, which is due to the transmit energy constraint that we have placed upon Alice. This constraint captures the energy limitations that exist within many IoT and IoBT devices. The energy constraint is ensured by Alice's batch normalization layer, which maintains the same total energy across the spread signals regardless of

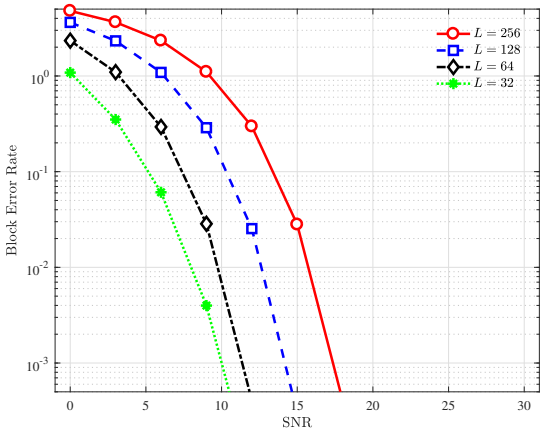

Fig. 4. Bob's BLER performance when Alice transmits DSSS signals of length L equal to 32 (∗), 64 (◇), 128 (□), and 256 (○) bits at SNR ∈ [0, 30] dB in 3 dB steps. Alice uses the "select" spreading codes.

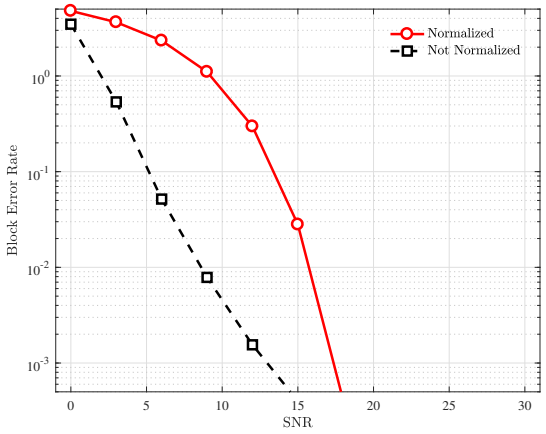

Fig. 5. Bob's BLER performance when Alice's transmit energy is constrained (dashed black line, □) versus when it is not (solid red line, ○) at SNR ∈ [0, 30] dB in 3 dB steps. Alice uses the "select" spreading codes.

their length, L. Thus, for the results shown in Fig. 4, the energy per transmitted bit decreases as L increases, which makes the spread signal and Bob's ability to recover the message more susceptible to the AWGN channel's negative impacts. This observation is supported by the results shown in Fig. 5, which shows BLER performance when the energy constraint is enforced versus when it is not.

V. Conclusion

In this work, a GAN inspired DL approach is implemented to facilitate DSSS-based communications between two cooperative entities, Alice (the transmitter) and Bob (the receiver), within the presence of an adversary (a.k.a., Eve) actively attempting to detect, intercept and exploit their communications. The 15 bit spreading codes employed by Alice and Bob are intentionally generated to adhere to specific balance and low side lobe stipulations that are unachievable by Gold, Kasami, and Weil codes. These "select" spreading codes hinder Eve's ability to detect the DSSS signals, but not stop it completely. However, when these codes are combined with Alice and Bob actively adapting the encoding scheme, then Eve loses the ability to effectively reconstruct Alice's messages from the detected and intercepted DSSS signals. Additionally, the BLER performance is assessed as the length of the spread signal increases and when Alice's transmit energy is constrained to accurately emulate IoT and IoBT device limitations.

References

[1] Defense Advances Research Projects Agency (DARPA), "Spectrum Collaboration Challenge — Using AI to Unlock the True Potential of the RF Spectrum," https://archive.darpa.mil/sc2/, 2017.

[2] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," in *IEEE Int'l. Conf. on Communications (ICC)*, 2018.

[3] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Trans on Cognitive Communications & Networking*, vol. 3, no. 4, Dec 2017.

[4] Defense Advances Research Projects Agency (DARPA), "Radio Frequency Machine Learning Systems," https://www.darpa.mil/program/radio-frequency-machine-learning-systems, 2019.

[5] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms," in *ACM Int'l Symposium on Mobile Ad Hoc Networking & Computing*, ser. Mobihoc, 2019.

[6] M. Fadul, D. Reising, and M. Sartipi, "Identification of ofdm-based radios under rayleigh fading using rf-dna and deep learning," *IEEE Access*, vol. 9, 2021.

[7] F. Restuccia and T. Melodia, "Polymorf: Polymorphic wireless receivers through physical-layer deep learning," in *Proc. of the Twenty-First Int'l. Symposium on Theory, Algorithmic Foundations, & Protocol Design for Mobile Networks & Mobile Computing*, ser. Mobihoc '20. Association for Computing Machinery, 2020.

[8] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, 2019.

[9] J. Downey, B. Hilburn, T. O'Shea, and N. West, "In the Future, AIs—Not Humans—Will Design Our Wireless Signals," *IEEE Spectrum Magazine*, Apr 2020.

[10] L. Cameron, "Internet of Things Meets the Military and Battlefield: Connecting Gear and Biometric Wearables for an IoMT and IoBT," 2017. [Online]. Available: https://www.computer.org/publications/tech-news/research/internet-of-military-battlefield-things-iomt-iobt

[11] V. Insinna, "DARPA Challenges Industry To Make Adaptive Radios With Artificial Intelligence," Sep 2016. [Online]. Available: https://www.defensenews.com/2016/09/08/darpa-challenges-industry-to-make-adaptive-radios-with-artificial-intelligence/

[12] U.S. Army Research Laboratory Combat Capabilities Development Command. Internet of battlefield things. [Online]. Available: https://www.arl.army.mil/business/collaborative-alliances/current-cras/iobt-cra/

[13] Statista, "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)," https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, 2019.

[14] Government Accountability Office (GAO), "Electromagnetic Spectrum Operations: DOD Needs to Take Action to Help Ensure Superiority," https://www.gao.gov/assets/gao-21-440t.pdf, Mar 2021.

[15] Y. Wei, S. Fang, X. Wang, and S. Huang, "Blind estimation of the pn sequence of a dsss signal using a modified online unsupervised learning machine," *Sensors*, vol. 19, no. 2, 2019.

[16] I. Shakeel, "Machine learning based featureless signaling," in *IEEE Military Communications Conf. (MILCOM)*, 2018.

[17] M. Abadi and D. Andersen, "Learning to protect communications with adversarial neural cryptography," *arXiv preprint:1610.06918*, 2016.

[18] T. Clancy and N. Goergen, "Security in Cognitive Radio Networks: Threats and Mitigation," in *Int'l. Conf. on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, 2008.

[19] T. Xie, G. Tu, C. Li, and C. Peng, "How Can IoT Services Pose New Security Threats In Operational Cellular Networks?" *IEEE Trans on Mobile Computing*, 2020.

[20] J. Vlok, "Detection of Direct Sequence Spread Spectrum Signals," Ph.D. dissertation, University of Tasmania, Oct. 2014.

[21] T. Kang, X. Li, C. Yu, and J. Kim, "A survey of security mechanisms with direct sequence spread spectrum signals," *Journal of Computing Science & Engineering*, vol. 7, Sep 2013.

[22] M. Simon, J. Omura, R. Scholtz, and B. Levitt, *Spread Spectrum Communications Handbook*. McGraw-Hill, Inc., 2002.

[23] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (corresp.)," *IEEE Trans on Information Theory*, vol. 13, no. 4, 1967.

[24] J. Rushanan, "Weil sequences: A family of binary sequences with good correlation properties," in *IEEE Int'l. Symposium on Information Theory*, 2006.

[25] ——, "The spreading and overlay codes for the l1c signal," *Navigation*, vol. 54, no. 1, 2007.

[26] T. Kasami, "Weight distribution formula for some class of cyclic codes," *Coordinated Science Laboratory Report no. R-285*, 1966.

[27] L. Welch, "Lower bounds on the maximum cross correlation of signals (corresp.)," *IEEE Trans on Information Theory*, vol. 20, no. 3, 1974.

[28] S. W. Golomb and G. Gong, *Signal design for good correlation: for wireless communication, cryptography, and radar*. Cambridge University Press, 2005.

[29] K. Mourougayane, B. Amgothu, and S. Srikanth, "A robust multistage spectrum sensing model for cognitive radio applications," *AEU - Int'l. Journal of Electronics & Communications*, vol. 110, Aug 2019.

[30] I. Shakeel, "Machine learning based featureless signaling," in *IEEE Military Communications Conf. (MILCOM)*, 2018.