

Machine Learning in NextG Networks via Generative Adversarial Networks

Ender Ayanoglu, *Fellow, IEEE*, Kemal Davaslioglu, *Member, IEEE*,
Yalin E. Sagduyu, *Senior Member, IEEE*

Abstract

Generative Adversarial Networks (GANs) are Machine Learning (ML) algorithms that have the ability to address competitive resource allocation problems together with detection and mitigation of anomalous behavior. In this paper, we investigate their use in next-generation (NextG) communications within the context of cognitive networks to address *i*) spectrum sharing, *ii*) detecting anomalies, and *iii*) mitigating security attacks. GANs have the following advantages. First, they can learn and synthesize field data, which can be costly, time consuming, and nonrepeatable. Second, they enable pre-training classifiers by using semi-supervised data. Third, they facilitate increased resolution. Fourth, they enable the recovery of corrupted bits in the spectrum. The paper provides the basics of GANs, a comparative discussion on different kinds of GANs, performance measures for GANs in computer vision and image processing as well as wireless applications, a number of datasets for wireless applications, performance measures for general classifiers, a survey of the literature on GANs for *i*–*iii*) above, and future research directions. As a use case of GAN for nextG communications, we show that a GAN can be effectively applied for anomaly detection in signal classification (e.g., user authentication) outperforming another state-of-the-art ML technique such as an autoencoder.

Index Terms

Generative adversarial networks (GANs), conditional GANs, generative modeling, spectrum sharing, anomaly detection, outlier detection, wireless security, unsupervised learning.

I. INTRODUCTION

The number of radios and their use are increasing exponentially. Over the last several years, wireless data transmission has grown up by approximately 50% per year [1], [2]. This increase in demand has largely been driven by novel applications such as streaming videos and social media on smart devices, and has strained the ability of the limited available wireless spectrum to support it. The coming era of the Internet-of-Things (IoT) and its anticipated goal of connecting tens of billions

E. Ayanoglu is with the Center for Pervasive Communications and Computing (CPCC), Dept. of EECS, UC Irvine, CA, USA. K. Davaslioglu is with UTS Inc., MD, USA, and Y. E. Sagduyu is with Intelligent Automation, A BlueHalo Company, MD, USA.

of devices via wireless will make this situation even more challenging for next-generation wireless communication networks (NextG). Classical methods for medium access control and physical layer operation, designed to handle a relatively small number of users in a given space-time-frequency resource, will soon be overwhelmed and alternatives must be sought.

Wireless systems of the future are anticipated to share the available spectrum rather than operating with exclusive assigned frequencies. It is generally expected that this sharing will be across all possible dimensions, including space, time, and frequency, and will involve a huge quantity of interactions among a very large number of radios. Flexible methods will be needed to efficiently use the limited available resources, quickly adapting themselves to changing environments and Quality-of-Service (QoS) requirements. This will be true even as wireless systems move to higher frequencies (such as anticipated for mmWave and THz bands) where available bandwidth is more abundant. Due to shorter propagation distances and the increased prevalence of blockages, it will require networks to continually reconfigure themselves via handoffs, cooperative relaying, beamforming, and so on. Given the exponentially increasing demand for wireless connectivity and the inevitable increase in the complexity of the networks that will supply it, it is no surprise that research is turning towards data-driven techniques of Artificial Intelligence (AI) and Machine Learning (ML) techniques for help in addressing these issues by learning from the spectrum data and solve complex tasks.

The Defense Advanced Research Projects Agency (DARPA) recently built a testbed named *Colosseum* for testing shared-spectrum communication systems [1], [2]. The vision is to emulate more than 65,000 unique interactions, such as text messages or video streams, among 128 radios at once. This testbed is built with the full expectation that the type of complicated spectrum sharing scenario described above is likely and can be efficiently managed by AI [1], [2].

There are two possible ways of sharing spectrum. One is an open access model, that is similar to an unlicensed band, such as the Industrial, Scientific, and Medical (ISM) band. A more desirable model from an interference viewpoint is known as the hierarchical access model, or the cognitive radio network. When the cognitive radio concept was first proposed, learning was already considered to be essential for its operation [3], [4]. There are three reasons why considering ML for cognitive radio is important. First, as discussed in [1], [2], due to the expected numbers of wireless devices and the complexity of the services they provide, the learning process suggested in [3], [4] cannot be managed by classical model-based techniques, and will need to employ ML algorithms. Second, for complex radios with many inputs and many outputs, a very large number of actions will be necessary to account for all possible radio states. In the past, most cognitive radio research focused

on radios that are hard-coded to manage these states. ML provides an opportunity for a learning engine to auto-generate these actions, rather than pre-programming them [5]. Third, ML provides a framework in which to incorporate memory from past actions and results into current operations, and thus more quickly adapt to changing conditions in the future [5].

As ML techniques rely on the availability of representative data, it has become imperative to generate and curate data samples for training and testing of spectrum operations. To that end, synthetic data can be generated to help with ML training and testing processes, when there is a lack of sufficient number of real data samples. Building upon recent advances in deep neural networks, Generative Adversarial Networks (GANs) have been originally introduced to generate synthetic data that cannot be distinguished from real data [6]. One direct use case of GANs is to generate synthetic training data samples for data augmentation purposes to strengthen the training process of ML algorithms. Later, the use of GANs has been extended to support domain adaptation, defend against adversarial attacks, enable unsupervised learning of data, detect anomalies embedded in rich data representations, and security applications. As far as defending against adversarial attacks, GANs are capable of generating adversarial samples. These can be used for network attacks that may be missed by the ML detector. Or, they can be added to the training set to help the ML detector adjust its classification boundary in order to acquire better detection capabilities.

While these use cases also apply in wireless communication systems, their effective application hinges upon careful account of wireless communication characteristics:

- 1) Training data for wireless communications is typically limited. There are sensing limitations regarding sampling rate (due to hardware effects) and time spent for sensing (balancing the sensing-communication tradeoff). Therefore, GANs can be used to augment the training data for wireless applications, such as spectrum sensing and wireless signal detection and classification (e.g., jammer identification) when the Radio Frequency (RF) data typically involves uncertainties due to noise, channel, traffic, and interference effects [7].
- 2) Wireless data is heavily environment dependent. For example, the training data collected in a laboratory environment (e.g., indoor) does not necessarily match channel characteristics in test time (e.g., outdoor). Therefore, the adaptation of test or training data to channel conditions raises the need for domain adaptation with GANs [8].
- 3) Signal spoofing is a key element of wireless security that poses threats for infiltration through signal/user authentication systems, and can be alternatively used to set up decoys/honeypots from the defense point of view [9], [10].

4) Wireless communication systems are subject to attacks due to the open nature of wireless spectrum and anomalies due to hardware impairments and intended/unintended interference. GANs can be used to protect wireless communications against attacks and detect anomalies.

The unique characteristics of wireless communication systems that need to be considered for the GAN applications are summarized as follows.

- 1) The input to the GAN is (real) RF data that is highly complex and dynamic, subject to noise, channel, traffic, and interference effects. Statistical modeling may not be effective to represent the spectrum data. Therefore, the complex structures of deep neural networks in the GAN formulation can be effectively used to represent the spectrum data [11]–[13].
- 2) The format of the RF data is not universal (like pixels in computer vision) and may vary from time series of in-phase/quadrature components (I/Qs) or Received Signal Strength Indicators (RSSIs) to frequency domain representations.
- 3) The generator and the discriminator of the GAN may need to be distributed at transmitter and receiver, respectively, as data transmission and the corresponding reception need to be separated through a wireless channel in online applications. In this context, the message exchange between the generator and the discriminator may be executed over noisy wireless channels.
- 4) While training the GAN, the RF data itself as well as communication channels between the generator and the discriminator may change due to the limited coherence time of channel conditions [9], [10].

Contributions of the paper are as follows. Sec. II introduces the basics of GANs and a number of alternative GAN structures, together with their underlying mathematics. GAN structures were originally developed for computer vision and image processing. This paper is on adopting them for applications in NextG wireless communications. To that end, Sec. III first discusses performance measures used in computer vision and image processing. It introduces performance measures that can be used in wireless applications since the former measures cannot be used in this field. It introduces public datasets that can be employed in wireless applications. It also discusses performance measures for general classifiers, including wireless classifiers. It discusses the use of GANs to extend a given dataset, called *data augmentation*. Furthermore, Sec. III introduces a number of GAN techniques that have been used in spectrum sharing, anomaly detection, and security for NextG wireless. A summary of these is available in Table 7. Sec. IV presents an anomaly detection problem for the modulation classification task and presents our results where we solve this problem using two unsupervised deep

learning methods, a GAN-based and an autoencoder-based algorithm. We also provide interpretations of the results, tradeoffs between the two approaches, and implementation details. Sec. V provides a discussion of a number of potential future research directions, including their challenges. Finally, Sec. VI provides our conclusions.

II. PRELIMINARIES ON GENERATIVE ADVERSARIAL NETWORKS

GANs were introduced in 2014 [6], [14], and belong to the class of deep-learning-based generative models. Generative modeling is an unsupervised ML task that involves automatically discovering the regularities or patterns in input data in such a way that the model can be used to generate new samples that cannot be discriminated from the original dataset. Facebook’s Di-

rector of Artificial Intelligence Research Yann Le Cunn is famously known to have said in June 2016 that “GANs and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion.” Given the many subsequent applications of GANs in different fields, some of which we discuss below, it is difficult to argue with this opinion.

GANs train a generative model by decomposing the problem into two sub-models, as depicted in Fig. 1. The first is a generator network trained to generate new examples, while the second is a discriminator network that tries to classify examples as either real (from the domain) or fake (generated). The two models are jointly trained that continues until the generator network learns to synthesize plausible samples that cannot be distinguished from real data.

GANs and their variations have been used in a variety of applications. A few examples include creation of synthetic human faces [15]–[17] and transforming images from one domain (e.g., real scenery) to another domain (paintings by famous painters) [18]. Other examples of image-to-image translation in [18] include conversion of horse pictures to pictures of zebras and vice versa, pictures of summer scenery to pictures of the same scenery in winter, etc. Further applications include face aging [19], creation of super-resolution images from those of low resolution [20], music generation [21], audio synthesis [22], and video synthesis [23]. More relevant to this paper, GANs have been applied to problems in spectrum sensing and security in wireless networks, e.g., [8], [9], [11], [24]–[27], and these will be discussed in more detail below. A list of the many different applications of

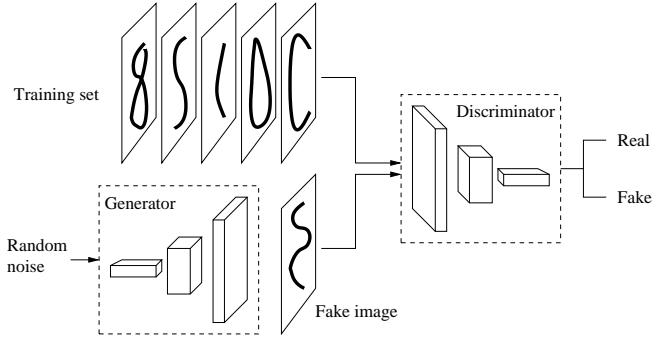


Fig. 1: Generative Adversarial Network (GAN) with image generation application [14].

GANs is available in [28].

To describe how GANs are trained, let \mathbf{x} belong to the manifold of the real input data with distribution $p_{\text{data}}(\mathbf{x})$, and let \mathbf{z} belong to the latent or noise prior space with distribution $p_{\mathbf{Z}}(\mathbf{z})$. Further, let G be a differentiable function representing the generator with input \mathbf{z} , and let D be a differentiable function representing the discriminator with input \mathbf{x} or $G(\mathbf{z})$, where the output of D is mapped to the interval $[0, 1]$. Now consider the function

$$V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \quad (1)$$

The first term increases when the real samples are more correctly classified, while the second term increases as the discriminator more successfully identifies the generated samples as fake. Thus, the discriminator D acts to maximize (1). On the other hand, G attempts to minimize (1), that is equivalent to minimizing just the second term, which increases the likelihood that the discriminator is fooled. This leads to a minimax optimization in which D and G work against each other to achieve the equivalent of a Nash equilibrium point [6], [29]:

$$\min_G \max_D V(D, G). \quad (2)$$

The two optimizations in (2) are carried out by employing neural networks and backpropagation via gradient ascent and gradient descent, which is possible since D and G are differentiable.

Given a batch $\{\mathbf{x}_i, \mathbf{z}_i\}_{i=1}^n$ of training data and samples from the latent space, we can convert the optimization expressed via (1) and (2) into the optimization of two cost functions, for D and G , respectively, as

$$J_D = -\frac{1}{n} \left(\sum_{i=1}^n \log D(\mathbf{x}_i) + \sum_{i=1}^n \log(1 - D(G(\mathbf{z}_i))) \right), \quad J_G = -\frac{1}{n} \sum_{i=1}^n \log(1 - D(G(\mathbf{z}_i))). \quad (3)$$

Details on implementing (3) with a neural network and gradient ascent and descent to train the GAN can be found in [6, Algorithm 1]. Note that by substituting the criterion in (1)–(2) with that in (3), an implicit assumption of ergodicity, or ensemble averages being equal to time averages, is made. This assumption, common in signal processing and communications, will be made again in the sequel.

Additional information that is correlated with the input data, such as class labels, can be used to improve GAN performance, either in the form of more stable or faster training, or generated images that have better quality. Such conditional GANs (CGANs) are trained in such a way that both the generator and the discriminator models are conditioned on the class label, so that when the trained generator is used as a standalone model to generate samples in the domain, samples

of a given type can be generated [30]. For example, in the synthesis of faces one could focus on generating a female face, or one could convert a summer scene into a winter scene, etc. Popular face aging applications are also based on this principle. To describe this approach, let y represent extra information on which the generator and discriminator are conditioned. One can perform the conditioning by feeding y into both the discriminator and generator as additional input layer. As mentioned above, the information y could be in the form of a class label, leading to a so-called class-conditional GAN, or some other kind of input such as a single image, in the case the GAN is performing an image-to-image translation task. Letting y represent a potentially multidimensional label from a distribution $p_Y(y)$, the optimization in (1) and (2) becomes

$$\min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim p_{\text{data}}(\mathbf{x}, \mathbf{y})} [\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{y} \sim p_Y(\mathbf{y}), \mathbf{z} \sim p_Z(\mathbf{z})} [\log(1 - D(G(\mathbf{z}, \mathbf{y}), \mathbf{y}))] \quad (4)$$

for CGANs, with (3) updated in a straightforward fashion based on (4).

A number of other GAN formulations exist beyond those mentioned above. For example, in the Least-Squares GAN (LSGAN) [31], the objective functions become

$$\min_D \{ \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [(D(\mathbf{x}) - a)^2] + \mathbb{E}_{\mathbf{z} \sim p_Z(\mathbf{z})} [(D(G(\mathbf{z})) - b)^2] \}, \quad \min_G \mathbb{E}_{\mathbf{z} \sim p_Z(\mathbf{z})} [(D(G(\mathbf{z})) - c)^2], \quad (5)$$

where a and b represent labels for real data and fake data, respectively, and c be the value that G wants D to believe for fake data. Reference [27] employs an LSGAN with $a = -b = 1$ and $c = 0$. Another variation is known as Wasserstein GAN (WGAN) [32], which employs the loss function

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [D(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim p_Z(\mathbf{z})} [D(G(\mathbf{z}))], \quad (6)$$

where \mathcal{D} is a set referred to as the set of 1-Lipshitz functions. Reference [33] removes the condition $D \in \mathcal{D}$ but adds an additional term for stability, repeatability, and predictable behavior during the training process. Reference [11] employs a WGAN with the additional term proposed in [33]. An alternative way to enforce the Lipshitz constraint is to add a penalty on the gradient norm for random samples $\hat{\mathbf{x}} \sim p_{\hat{\mathbf{X}}}$, and the updated loss function can be expressed as

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [D(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim p_Z(\mathbf{z})} [D(G(\mathbf{z}))] + \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim p_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]. \quad (7)$$

This approach is known as WGAN with gradient penalty (WGAN-GP) [33] and it provides stability and predictable behavior during the GAN training process.

A. Anomaly Detection by Using GANs

One area of particular importance where GANs have been applied is in anomaly detection, which is the task of discovering anomalies, or patterns in the data that do not conform to “normal behavior.” While the use of GANs is based on modeling normal behavior using the adversarial training process,

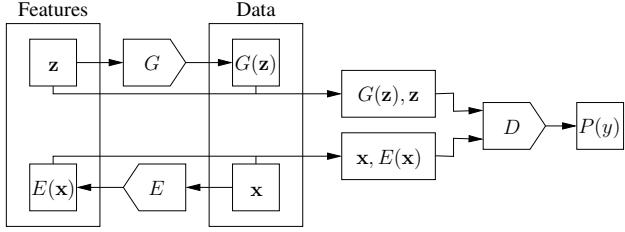


Fig. 2: The structure of BiGAN [34].

an *anomaly score* based on this model can be calculated and employed for anomaly detection. Several GAN-based approaches for anomaly detection have been published, all of which employ the technique of Adversarial Feature Learning idea [34]. This idea makes use of a novel architecture referred to as bidirectional GANs, or BiGANs.

BiGANs add an inverse mapping from the data space to the latent distribution, from which regular GANs generate artificial samples. The overall model is depicted in Fig. 2. A BiGAN includes an encoder E which maps data \mathbf{x} to latent representations \mathbf{z} , in addition to the generator G employed by the standard GAN architecture [34]. The BiGAN discriminator D discriminates not only in data space (\mathbf{x} versus $G(\mathbf{z})$), but jointly in both the data and latent space (tuples $(\mathbf{x}; E(\mathbf{x}))$ versus $(G(\mathbf{z}); \mathbf{z})$), where the latent component is either an encoder output $E(\mathbf{x})$ or a generator input \mathbf{z} .

The two modules E and G do not directly “communicate” with one another: the encoder never “sees” generator outputs ($E(G(\mathbf{z}))$ is not computed), and vice versa. However, it is shown in [34] that the encoder and generator must learn to invert one another in order to fool the BiGAN discriminator. In other words, E learns the inverse of the generator $E = G^{-1}$. The encoder E is a nonlinear parametric function in the same way as G and D , and can be trained using a standard learning algorithm such as gradient descent. A latent representation \mathbf{z} may be thought of as a “label” for \mathbf{x} , but one which came for “free,” without the need for supervision. As discussed in [34], BiGANs make no assumptions about the structure or type of data to which they are applied.

In this paper, we will focus on three GAN architectures that conceptually employ the inverse generator concept of BiGANs. These are the AnoGAN [35], Efficient GAN-Based Anomaly Detection (EGBAD) [36], and a GAN + autoencoder approach [37]. AnoGAN is a deep convolutional GAN that learns a manifold of normal variability, together with a scoring scheme that labels anomalies based on the mapping from the image to the latent space. For results on medical imaging data, see [35]. Given a query image \mathbf{x} , the algorithm iterates through points in the latent space to find a representation $G(\mathbf{z})$ that is close to \mathbf{x} . The algorithm begins by choosing a random point \mathbf{z}_1 in the latent space, and generates a data sample $G(\mathbf{z}_1)$. The algorithm then proceeds through a number of

points $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_\Gamma$ based on the following loss function:

$$\mathcal{L}(\mathbf{z}_\gamma) = (1 - \lambda)\mathcal{L}_R(\mathbf{z}_\gamma) + \lambda\mathcal{L}_D(\mathbf{z}_\gamma), \quad (8)$$

where

$$\mathcal{L}_R(\mathbf{z}_\gamma) = \|\mathbf{x} - G(\mathbf{z}_\gamma)\|_1 \quad (9)$$

is the residual loss, and

$$\mathcal{L}_D(\mathbf{z}_\gamma) = \|f(\mathbf{x}) - f(G(\mathbf{z}_\gamma))\|_1 \quad (10)$$

is the discriminator loss, where f is the output of one of the layers of the multilayer perceptron and $0 < \lambda < 1$ is an interpolation coefficient. The residual loss enforces similarity between the query data \mathbf{x} and the generated data $G(\mathbf{z}_\lambda)$, while the discrimination loss constrains the generated data to lie near the learned manifold \mathcal{X} [38]. For each $\gamma = 2, 3, \dots, \Gamma$, \mathbf{z}_γ is calculated by iteratively minimizing (8) via backpropagation steps. The iteration is on \mathbf{z}_γ , the coefficients of G and D are not changed. The approach of using a linear combination of two loss functions (9) and (10) that employ the L_1 norm for training a deep neural network is different than the approaches discussed earlier for training GANs and CGANs. This approach originated with [39] and was adapted in [35].

Implementing the AnoGAN optimization over Γ steps results in a relatively high computational load. The EGBAD approach [36] was developed as a more efficient alternative and is based on the methods in [34], [40], which enable learning an encoder E by mapping input samples to their latent representation during adversarial training. GANomaly [37] is designed to be an improvement over [34]–[36] in terms of both performance and speed. The algorithm trains a generator network to learn the manifold of the input samples while at the same time training an autoencoder to encode the data in their latent representation. This approach uses a discriminator, a decoder, and two encoders, but the encoders have the same architecture. We describe this approach in more detail below.

As shown in Fig. 3–4, the generator network has three elements in series: an encoder G_E , a decoder G_D , and another encoder E . The combination of G_E and G_D forms an autoencoder that functions as the generator G . The encoder E has the same structure as G_E . G_E takes the data sample \mathbf{x} and generates an encoded version \mathbf{z} in the latent space. Then G_D employs \mathbf{z} to create $\hat{\mathbf{x}}$, which is a reconstructed form of \mathbf{x} . Finally, $\hat{\mathbf{x}}$ is used to generate another point in the latent space, $\hat{\mathbf{z}}$. Three loss functions are defined, which are combined to generate the overall generator loss. The first is the adversarial loss \mathcal{L}_{adv} ,

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}(\mathbf{x})} \|f(\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}(\mathbf{x})} f(G(\mathbf{x}))\|_2 = \|f(\mathbf{x}) - f(\hat{\mathbf{x}})\|_2, \quad (11)$$

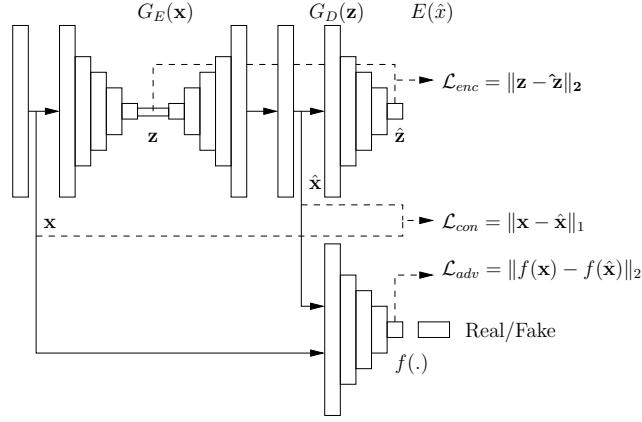


Fig. 3: Simplified single-dimensional architecture representation of GANomaly. Multiple blocks in series represent two-dimensional convolutional encoders and single blocks are for providing two-dimensional input-output [37].

where f is the output of an intermediate layer in the multi-layer perceptron. The second loss function is the contextual loss and is given by

$$\mathcal{L}_{\text{con}} = \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \|\mathbf{x} - G(\mathbf{x})\|_1 = \|\mathbf{x} - \hat{\mathbf{x}}\|_1, \quad (12)$$

and the third is the encoder loss

$$\mathcal{L}_{\text{enc}} = \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}(\mathbf{x})} \|G_E(\mathbf{x}) - E(G(\mathbf{x}))\|_2 = \|\mathbf{z} - \hat{\mathbf{z}}\|_2. \quad (13)$$

Finally, the generator loss is given as

$$\mathcal{L} = w_{\text{adv}} \mathcal{L}_{\text{adv}} + w_{\text{con}} \mathcal{L}_{\text{con}} + w_{\text{enc}} \mathcal{L}_{\text{enc}}, \quad (14)$$

where w_{adv} , w_{con} , and w_{enc} are coefficients between 0 and 1 that sum to 1. During the test stage, the model uses \mathcal{L}_{enc} given in (13) for scoring the abnormality $A(\hat{\mathbf{x}})$ of a given image:

$$A(\hat{\mathbf{x}}) = \|G_E(\hat{\mathbf{x}}) - E(G(\hat{\mathbf{x}}))\|_1. \quad (15)$$

In order to make the anomaly score easier to interpret, [37] proposes to compute it for every sample $\hat{\mathbf{x}}$ in the test set \hat{D} to obtain the set $S = \{s_i : A(\hat{\mathbf{x}}, \hat{\mathbf{x}} \in \hat{D}\}$ of individual anomaly scores,

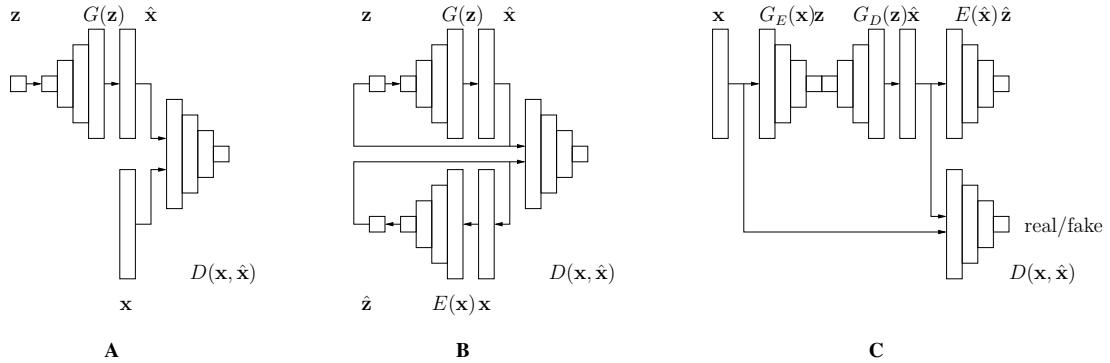


Fig. 4: Comparison of GANs for anomaly detection for computer vision and image processing applications, A: AnoGAN [35], B: EGBAD [36], C: GANomaly [37]. Two-dimensional convolutional encoders and input-output devices are depicted as single-dimensional blocks for simplicity without any loss in functionality.

Abbreviation	Application	Comment	Reference
GAN	General	Employs (1)-(3). Structure in Fig. 1.	[6], [14]
CGAN		Conditional GAN. Employs (4).	[30]
LSGAN		Least Squares GAN. Employs (5).	[31]
WGAN		Wasserstein GAN. Employs (6).	[32]
WGAN-GP		Gradient Penalty WGAN. Employs (7).	[33]
BiGAN	Anomaly Detection	Bidirectional GAN. Structure in Fig. 2.	[34]
AnoGAN		Anomaly GAN. Employs (8)-(10).	[35]
EGBAD		Efficient GAN-Based Anomaly Detection.	[36]
GANomaly		GAN+Autoencoder. Employs (11)-(16). Structure in Fig. 3.	[37]

Table 1: Different versions of GANs used in computer vision and image processing applications. Note that Fig. 4 compares the structures of AnoGAN, EGBAD, and GANomaly. Which of GAN, CGAN, LSGAN, WGAN, or WGAN-GP is preferable depends on the application. For the tradeoffs among the versions used in anomaly detection, see the text.

and then apply a scaling to force the scores to lie within the range $[0, 1]$:

$$s'_i = \frac{s_i - \min(S)}{\max(S) - \min(S)}. \quad (16)$$

Fig. 4 shows a comparison of the architectures of the three GAN-based anomaly detection algorithms discussed above. Experiments show that EGBAD is faster than AnoGAN, and that GANomaly achieves better performance and speed than EGBAD [37], [41]. Table 1 provides a summary of GAN versions used for computer vision and image processing applications.

III. APPLICATION OF GANs TO 5G AND BEYOND

In what follows, we will first discuss, in Sec. III-A, measures of performance and specific issues that have to do with applying GANs first to computer vision and image processing and then to wireless applications. We will discuss public datasets available for enabling the training and benchmarking the performance of GANs in the wireless domain. In the same section, we will discuss measures for general classifier performance. We will specifically use these measures in wireless classifier applications. In Sec. III-B, we will discuss the technique of data augmentation by using GANs, which can be employed in the following three sections. Then, in Sec. III-C–Sec. III-E, we will discuss fast, accurate, and robust methods for analyzing large quantities of spectrum data in order to identify opportunities for *i*) spectrum sharing, *ii*) detecting anomalies, and *iii*) mitigating security attacks, respectively.

A. Performance Measures for GANs and General Classifiers

We will now discuss performance measures for GANs, first for computer vision and image processing, and then for wireless applications, and finally, general classifiers. There are basically two measures of the quality of images generated by GANs for computer vision and image processing applications. These are Inception Score (IS) [38] and Fréchet Inception Distance (FID) [42], [43]. IS

offers a way to objectively and quantitatively evaluate the quality of generated images by a GAN. It is generally considered that this score is well-correlated with scores from human observers. Similarly, FID is used to evaluate the quality of images generated by GANs. FID compares the mean and standard deviation of one of the deeper layers in a Convolutional Neural Network (CNN) named Inception-v3. These layers are closer to output nodes and are believed to mimic human perception of similarity in images. Clearly, these two measures are useless for wireless applications. They need to be replaced with measures meaningful in the context of wireless communications.

For wireless applications, the goal is to calculate the similarity of two probability densities. A commonly used measure towards that end is the Jensen-Shannon distance, which is the square root of the Jensen-Shannon divergence [44]. Jensen-Shannon divergence, on the other hand, is a symmetric form of the well-known Kullback-Leibler divergence [45].

In passing, we would like to state that to facilitate the training of GANs in the wireless domain and benchmark the performance of GANs, there are increasingly more public datasets available. Examples of such sets are RADIOML 2016.04C, RADIOML 2016.10A, RADIOML 2018.01A by DeepSig [46], RFMLS 2016a by DARPA [47], CBRS by NIST [48], and synthetic data generated by GNU Radio [49].

To measure the performance of any classifier, Probability of Detection (P_D) and Probability of False Alarm (P_{FA}) are calculated. Then, the Receiver Operating Characteristics (ROC), which plot P_D against P_{FA} are drawn [50]. A commonly used measure is Area Under Curve of the ROC (AUROC). Accuracy (ACC) is defined as the ratio of the correct classifications to the total classifications. It can be calculated as Accuracy = (True Positives + True Negatives) / (True Positives + False Positives + True Negatives + False Negatives). Furthermore, the variables Precision, Recall, and F1 Score determine the success of a binary classification problem. Precision equals the ratio of the true positives to total (true and false) classified positives. Recall equals the number of true positives to the sum of true positives and false negatives. Precision is a good measure to employ when the cost of the false positive is high whereas Recall is a good measure when the cost of the false negative is high. F1 Score, the harmonic mean [51] of Precision and Recall [52], is employed when it is desired to have a balance between Precision and Recall. All three measures take values between 0 and 1. For a given classification system, it is desirable that each be as close to 1 as possible. Please see Sec. IV for combinations of these quantities and the definitions of Density and Coverage in [53].

An important lesson learned in this subsection is that while it is desirable to employ GAN struc-

tures from computer vision and image processing for NextG wireless applications, the performance measures in the two fields are different.

B. Data Augmentation via GANs for NextG

By recognizing that supervised ML requires significant number of training data samples and it is expensive and likely even infeasible to collect a sufficiently comprehensive and representative set of training data, [8] provides *training data augmentation* by adding synthetic data to an existing training set. This is achieved by a workflow of three steps. In the first step, a CGAN is trained using real training samples. In this step, the generator learns to synthesize new data samples. In the second step, the synthetic data samples are used along with the real samples to train a classifier. In the third step, as new data comes in, the classifier is used for classification purposes. The authors of [8] show that this approach significantly improves the adaptation time and accuracy of the resulting spectrum sensing.

In an attempt to come up with a good modulation recognition technique, [54] employs Auxiliary Classifier GANs (AC-GANs) [55] after a density transformation of the signal called Contour Stellar Image to enhance the performance of CNNs. Although employing a training sequence whose length is 10% of another algorithm, [54] achieves up to 6% gain in the recognition performance of the test sequence.

Reference [13] investigates training data augmentation for deep learning RF systems by asking basic questions about the augmentation process. The paper concentrates on the Automatic Modulation Classification (AMC) problem. These questions are: *i*) how useful a synthetically trained system will be when deployed without considering the environment within the synthesis, *ii*) how can augmentation be leveraged within the RF ML domain, and *iii*) what impact knowledge of degradations to the signal caused by the transmission channel contributes to the performance of a system. The results show that for data generation to a higher fidelity, the propagation path from the Digital-to-Analog Converter (DAC) to the Analog-to-Digital Converter (ADC) must be investigated and modeled. Second, although augmentation provides savings in terms of time and money, the authors suggest a cost analysis to achieve a balance between the two. Finally, a methodology is established for the quantity of data needed.

A lesson that should be derived here is that GANs can generate synthetic training data and thus augment the training set in ML. We will discuss this topic in more detail in Sec. V.

C. Spectrum Sharing

As NextG networks rely on the co-existence of heterogeneous networks such as cellular and non-terrestrial networks, and radar systems, spectrum sharing is expected to play a major role in NextG communications and radar systems, while relying on ML techniques to identify and make use of spectrum opportunities. One particular example is Citizens Broadband Radio Service (CBRS) band at 3.5 GHz where the cellular wireless system needs to share the spectrum with the tactical radar while avoiding mutual interference [56]. As such, the use of ML for spectrum sensing problems is currently a growing area of interest, as evidenced in the survey [57] which summarizes results from [58]–[67]. The work reported in [58]–[61] focuses on ML issues including optimization of supervised classifiers, reduction of the feature space dimension, reduction of interference, and minimization of the number of sensors, respectively. Attacker detection and minimization of sensing time, delay, and operations in collaborative spectrum sensing are discussed in [62], [63]. Techniques that target Spectrum Sensing Data Falsification (SSDF) are developed in [64]–[67]. Furthermore, [68]–[71] discuss the use of neural networks for spectrum sensing under low Signal-to-Noise Ratio (SNR) conditions. References [72], [73] discuss employing neural networks for spectrum sensing in the face of parameter uncertainties in the transmitted signals.

Spectrum sharing techniques can be broadly characterized as belonging to open sharing or hierarchical access models [74]–[76]. In the open sharing model, each network accesses the same spectrum without any interference constraint from one network to its peers. The unlicensed band is an example of this model. The hierarchical or cognitive radio model consists of a primary network and a secondary network that accesses the primary spectrum without interfering with the Primary Users (PUs). We will focus on the concept of cognitive radio networks in order to achieve efficient spectrum sharing via ML, using the classification from [77] to organize potential approaches to spectrum sharing in cognitive radio networks. In this classification, spectrum sharing techniques are categorized as either underlay, overlay, or interweave.

Table 2 summarizes the basic characteristics of underlay, overlay, and interweave techniques [77]. In the underlay technique, a Secondary User (SU) needs to be aware of the channels for all active PUs in order to know if the interference it generates will be below an acceptable limit. This consideration limits its transmit power. For the overlay technique, in addition to the PU channel characteristics, the SU also knows the codebooks and messages of the active PUs. The SU can transmit at

Underlay	Overlay	Interweave
SU knows the channel strengths of PU	SU knows the channel gains, codebooks, and messages of PU	When PU is not using the spectrum, SU knows the spectral holes in space, time, or frequency
As long as the interference is below an acceptable limit, SU can simultaneously transmit with PU	SU can simultaneously transmit with PU; interference to PU can be offset by using part of SU's power to relay PU's message	SU can simultaneously transmit with PU only in the case of false spectral hole detection
SU's transmit power is limited by the interference constraint	SU can transmit at any power; the interference to PU can be offset by relaying PU's message	SU's power is limited by the range of its spectral holes

Table 2: Characteristics of underlay, overlay, and interweave techniques for cognitive radio networks [76], [77].

any power, but to offset the interference it causes to the PUs, it relays their transmitted messages. In the interweave technique, the goal is to opportunistically communicate in the spectrum holes, or more generally the space-time-frequency voids that are not in use by either licensed and unlicensed users. As depicted in Fig. 5, these voids change with frequency and time. The interweave technique requires periodic monitoring of the spectrum for detection of user activity, so that the SU can transmit opportunistically over the space-time-frequency voids with minimal interference. All of the above approaches can benefit from SUs employing directional antennas or beamforming for more flexible control of the interference. Fig. 6 depicts spectrum sensing by a Software Defined Radio (SDR) in an actual setting.

For this research topic, implementing the spectrum sensing task using a GAN-based technique as in [8] is an interesting and potentially useful first step. It provides *domain adaptation* by creating synthetic data that enables the classifier to quickly adapt to changes in the spectrum environment. The approach in [8] can be extended to exploit voids in all possible dimensions including space, time, and frequency, which is of particular importance to interweave cognitive radios. To extend the research in this direction, an approach based

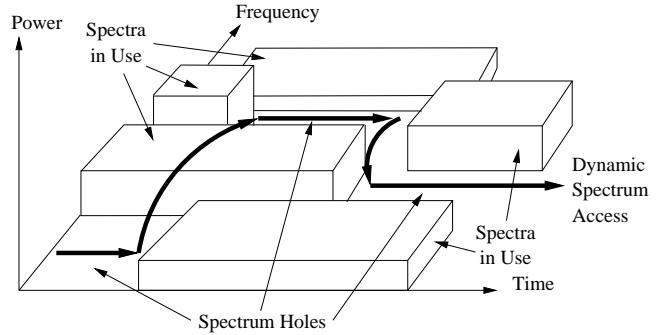


Fig. 5: Holes in frequency and time domains [78].

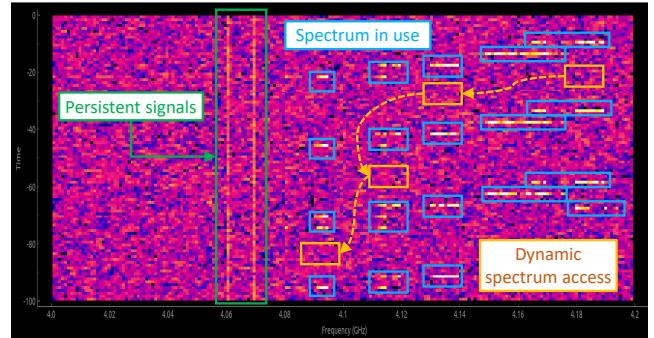


Fig. 6: Spectrum usage in a shared band and spectrum holes in the frequency (vertical) and time (horizontal) domains enabling dynamic spectrum access. For the figure, spectrum sensing is done by HackRF Software Defined Radio (SDR), and QSpectrumAnalyzer library is used for the real-time data visualization.

on CGANs that emphasizes specific subsets of the space, time, and frequency spectrum using specific domain knowledge can be employed. For example, an approach based on CGANs, conditioned on specific subsets of the space, time, and frequency using specific domain knowledge can be employed. Such a CGAN will be able to generate synthetic data that is specific to certain “labels” that correspond for example to known locations (e.g., hot spots), time periods (e.g., rush hour), or occupancy patterns of certain frequency bands. A further step in this direction is the investigation of the performance of the algorithms in the presence of low SNR as in [68]–[71] and parameter uncertainties as in [72], [73].

The approach in [8] consists of the following three steps: *i*) training a GAN using real training samples so that the generator learns to synthesize new samples, *ii*) training of a classifier with augmented data so that the original and limited training set is expanded with synthetic data, and, *iii*) regular operation by classifying data using the trained model. It is possible to extend this approach by training a CGAN for more typical scenarios, such as known locations, time periods, and occupancy patterns of certain bands, as discussed in the previous paragraph. In this context, unlike [8], it is possible to make a comparison of this ML-based approach with conventional narrowband spectrum sensing techniques based on energy detection, cyclostationary feature detection, matched filter detection, and wideband spectrum sensing techniques such as compressed sensing [57], [79] in terms of performance and complexity, specifically in the three areas discussed in Sec. I.

In addition to [13] discussed in Sec. III-B, another work on the AMC problem is provided in [80]. This work uses a modified version of CGAN and AC-GAN, which is operated in a semi-supervised (not unsupervised) mode. The modifications are needed to solve the following problems: *i*) the complexity of the modulation signals cause non-convergence due to the mapping from the high-dimensional parameter space to the low-dimensional classification space, and *ii*) lack of diversity of the generated samples cause mode collapse. Mode collapse is defined as the difficulty in convergence, especially during the training process, due to conditional constraints. The modifications consist of an encoder block E , a transform block T , and the splitting of the classification and discrimination functions of the conventional discriminator D by adding an explicit classifier C [80, Fig. 1]. The encoder E and the transform T are inspired by Conditional Variational Autoencoder (CVAE) [81] and Spatial Transformer Network (STN) [82], respectively. The definition of the loss functions involving all the blocks and their optimization expressions are given in [80, Eq. (10)-(20)]. To evaluate the performance of the system, the synthetic dataset in [49] is used. This dataset contains signals of eleven modulation modes at SNRs ranging from -20 dB to 20 dB, with channel distortion, frequency offset,

phase offset, and Gaussian noise added. Comparisons with a number of well-known deep learning methods show gains in classification accuracy of up to 12%.

An important implementation technique for NextG networks is Cooperative Spectrum Sensing (CSS), where a number of SUs cooperate to improve sensing performance. CSS can be implemented using either a centralized or a distributed approach. In centralized sensing, the SUs send their sensing information to a Fusion Center (FC) that makes the sensing decision and broadcasts this information back to all SUs. In distributed sensing, the SUs share sensing information among each other and make independent sensing decisions locally. In both approaches, the SUs can either transmit their individual sensing decisions or soft information in the form of partial statistics. The information combining schemes used by the FC or SUs can be categorized as either hard or soft combining schemes. The most common hard combining algorithms are AND combining, OR combining, M -out-of- N combining, and quantized hard combining, while soft combining algorithms include selection combining, maximal ratio combining, equal gain combining, and square law combining [83]. The soft combining approach provides the best detection performance, but at the cost of additional control channel overhead.

It is desirable to extend the technique described above to include CSS. In one implementation, GANs can be placed at the SUs and classifications can be combined at the FC. An alternative is to have the SUs transmit partial statistics to be used by a single GAN at the FC. In this regard, it is worthwhile to investigate schemes that will employ combinations of centralized versus distributed CSS with hard or soft decisions. The number of combinations is high but it should be possible to reach quick conclusions. The main interest in this activity is to determine the optimal placement for GANs in interweave networks, and how much performance improvement they can provide.

There is an important lesson that can be drawn from this subsection. Due to the tremendous number of users and applications in NextG networks, spectrum sharing will need to be carried out in a dynamic environment employing principles of cognitive networking. The goal is to determine the voids based on time, frequency, and space and exploit them to accomodate more users. Two specific techniques for spectrum sharing based on GANs are discussed in [8], [80] and summarized in this subsection.

D. Detecting Anomalies

In a GAN, the generator captures the distribution of the training data, and the discriminator can detect false from real, making a GAN an attractive ML technique for anomaly detection. This observation has led to a number of anomaly detection techniques employing GANs [84]–[93]. These

Model	Precision	Recall	F1 Score
OC-SVM [105]	0.7457	0.8523	0.7954
DSEBM-r [105]	0.8521	0.6472	0.7328
DSEBM-e [105]	0.8619	0.6446	0.7399
DAGMM-NVI [106]	0.9290	0.9447	0.9368
BiGAN [36]	0.9363	0.9512	0.9437
PCA+BiGAN [103]	0.9442	0.9592	0.9516

Table 3: Performance of the five anomaly detection algorithms on the KDDCUP-99 dataset [103]. OC-SVM: One-Class Support Vector Machine [105], DSEBM-r: Deep Structured Energy-Based Model - using reconstruction error [105], DSEBM-e: Deep Structured Energy-Based Model - using energy [105], DAGMM-NVI: Deep Autoencoding Gaussian Mixture Model with Neural Variational Inference [106].

applications are typically for various forms of image processing. There are some recent applications of using adversarial networks for anomaly detection in cyber-physical systems [94], [95], fraud detection in banking [96]–[99], driver assistance systems [100], air surveillance [101], prognostics and health management in the aeronautics industry [102], and network traffic anomaly detection [103].

We will briefly discuss [103] since it is related to network anomaly detection. This work proposes BiGAN in conjunction with Principal Component Analysis (PCA) for network anomaly detection using the KDDCUP-99 dataset [104]. This database includes a wide variety of intrusions simulated in a military network environment used for a competition during a conference to build a network intrusion detector, a predictive model capable of distinguishing between “bad” connections, called intrusions or attacks, and “good” normal connections [104].

A number of works exist that study anomaly detection in cognitive radio networks with conventional methods [107]–[113] and with ML [114]–[120]. We refer the reader to [120] for brief descriptions of [115]–[119]. These works provide an introduction to this field, with the important observation that GAN-based methods tend to provide better results in general. However, it is certain that existing research in this area is still limited, and this provides an opportunity for significant contributions.

In [114], an adversarial autoencoder was implemented for wireless spectrum anomaly detection [121], which uses ideas similar to a GAN. We will go beyond this approach and leverage the advantages provided by AnoGAN, EGBAD, and GANomaly. We provide details here on an approach using GANomaly, although similar anomaly scores can be defined for other architectures. Consider a collection of received data vectors X_N , whose elements are for example the I and Q components of a digitally modulated signal, or vectors of the sampled power spectral density. We seek a model that learns the source distribution $p(X_N)$, so that we can detect when the vector’s distribution is different from $p(X_N)$. This is a hypothesis testing problem, where for each vector $\mathbf{x} \in X_T$ in the

test dataset X_T , the two hypotheses are

$$\begin{aligned} H_N: & \text{ Sample data comes from } p(X_N), \\ H_A: & \text{ Sample data does not come from } p(X_N). \end{aligned} \tag{17}$$

We have three assumptions: First, the probability of anomalous behavior in dataset X_N is very low; second, no explicit anomaly labeling is done on the test dataset; and third, no feature extraction is performed before feeding the data to the model. The key insight in employing GAN-based anomaly detection is to bring the data to the latent space, which captures relevant features that can be used to reconstruct the actual input data, with a small anomaly score for normal data. A GAN is trained using (1) and (2), and during the test and operation stage, (11)–(15) are used. Once the training process is complete, the model weights are frozen and new data are input into the model. Anomalies are detected based on the anomaly score or the reconstruction loss of the model.

The term anomaly covers a very broad range of possibilities in the transmitted signal. Since the field of ML progresses based primarily on experimentation, a number of datasets should be created to study the performance of the algorithms under consideration. These can be in the form of time series corresponding to I/Q data from a radio receiver after downshifting in frequency and mixing with the local oscillator. This signal includes the effects of fading and multipath. To create anomalous test signals, there are a wide variety of possibilities. Reference [114] considered four types of normal signals and four types of anomalies. The normal signals were generated as either *i*) a single continuous signal with random bandwidth, SNR, and center frequency, *ii*) pulsed signals with parameters similar to *i*), *iii*) multiple continuous signals with possible frequency overlap, or *iv*) signals with random bandwidths and SNR with deterministic shifts/hops in frequency. The anomalous signals were chosen as either *i*) the same as normal signals in *i*) above, *ii*) random pulsed transmissions in the given band, *iii*) pulsed wideband signals covering the entire frequency band, or *iv*) signals from other classes in the synthetic dataset. Normal and anomalous signals such as these can be considered, as well as other random signal types with arbitrary power spectral densities.

Anomaly detection within the context of cognitive radio in the mmWave band is studied in [120], where the authors generate their own datasets with a number of anomalies and evaluate three methods including two based on GANs for their detection. As mmWave radios are vulnerable to malicious users due to the shared access medium and the complex radio environment, this opens up new threat possibilities. Moreover, since several security critical applications in the NextG networks such as the Vehicle-to-Everything (V2X) networks are based on mmWave communications, this problem is particularly important. In addressing this problem, [120] makes use of three ML techniques:

Variational Autoencoder (VAE) [122], CGAN [30], and AC-GAN [55]. This work created its own dataset by using mmWave equipment in a laboratory environment with implementation using Field Programmable Gate Arrays (FPGAs) and building blocks such as local oscillators, intermediate frequency modules, mmWave radio heads, horn antennas, etc. The operation is at 28 GHz by using Cyclic-Prefix Orthogonal Frequency Division Multiplexing (CP-OFDM) with BPSK, QPSK, 16-QAM, and 64-QAM. Complex I/Q data is collected at baseband after the downconversion process. There are eight channels with 100 MHz bandwidth. The normal behavior is a fixed signal which occupies channel 4 and a sequentially moving signal among channels 8, 6, 3, and 1. This normal behavior is used in the training phase. For testing, three modalities are introduced. Modality 1 is a fixed signal in channel 4, and a moving signal that jumps among channels 5, 7, 2, and 5. Modality 2 is a fixed signal in channel 4 and a moving signal that jumps among channels 7, 5, 2, and 1. Modality 3 is a fixed signal in channel 4 and a moving channel that jumps among channels 5, 7, 6, and 5. Performance results are provided by AUROC and ACC. These results are given in Table 4. CGAN provides better performance than VAE and AC-GAN provides better performance than CGAN, namely, VAE performs an AUROC between 0.9365 and 0.9577, CGAN between 0.9545 and 0.9737, and AC-GAN between 0.9741 and 0.9751. Reference [120] attributes the better performance by GANs to the fact that they learn the relation between random noise and generated data such that the generated data is close to real data, and as a result, they can capture the dynamics in the real data. Whereas, VAEs return the posterior probability that an observation belongs to a cluster. They do so by learning the latent vector corresponding to an input, and as a result, all the dynamics in the real data that VAEs have access to are limited to the latent vectors already modeled. Due to a larger degrees of freedom with the former, it is generally considered that GANs can have better performance. Computational times for the three algorithms, performed on an NVIDIA GeForce GTX 1080 Ti GPU, are given in Table 5. Clearly, there is a big price paid for the better performance of AC-GAN (15 minutes vs. 30 minutes).

In [123], a deep learning based signal or modulation classification solution is described, where i

	Modality	AUROC	ACC
VAE	1	0.9365	0.9356
	2	0.9577	0.9551
	3	0.9232	0.9382
CGAN	1	0.9566	0.9657
	2	0.9737	0.9696
	3	0.9545	0.9668
AC-GAN	1	0.9741	0.9804
	2	0.9751	0.9757
	3	0.9742	0.9660

Table 4: AUROC and ACC performance of the three ML algorithms.

	Training [mm:ss]	Testing [mm:ss]
VAE	15:09	01:00
CGAN	15:16	01:36
AC-GAN	30:42	03:16

Table 5: Computational times of the three ML algorithms.

signal types may change over time, *ii*) some signal types are not known *a priori*, and therefore there is no training data available, *iii*) signals are potentially spoofed such as smart jammers replaying other signal types, *iv*) different signals may be superimposed due to interference from concurrent transmissions. The authors present a CNN that classifies the received I/Q samples as idle, in-network signal, jammer signal, or out-network signal. Traditional approaches for signal classification require expert design or knowledge of the signal. Modulations are classified into *i*) idle, *ii*) in-network user signal, *iii*) jamming signals, and *iv*) out-network user signals where there are 3, 4, and 3 different modulation techniques corresponding to *ii*–*iv*). There are in-network users who try to access the channel opportunistically (SUs), out-network users with priority channel access (PUs), and jammers that all coexist. The authors use the dataset in [124]. There are ten modulations with SNRs from -20 dB to 18 dB in 2 dB increments. Radio fingerprinting via radio hardware imperfections such as I/Q imbalance, time or frequency drift, and power amplifier effects are used to identify the type of a transmitter. In addition to a CNN structure, Minimum Covariance Determinant (MCD) , *k*-means clustering, and Independent Component Analysis (ICA) techniques are employed. Results demonstrate the feasibility of using deep learning to classify RF signals with high accuracy in unknown and dynamic spectrum environments. By using the signal classification results a distributed scheduling protocol is developed where SUs share the spectrum with each other while avoiding interference imposed to PUs and received from jammers.

The use of GANs that exploit the spatial domain to identify anomalous signal sources can be considered, using data received from antennas in different locations. In many applications, normal network operations are consistent with users transmitting from specific locations, such as in a stadium setting, a large hall, a shopping mall, etc. A GAN trained in this setting with an array of antennas or widely separated receivers can be used to differentiate between normal network traffic and signals that arrive from anomalous directions, even if their spectral characteristics are identical to normal users. A similar approach can be used to detect differences between mobile and stationary sources, even if no detectable Doppler shift can be measured. Mobile sources typically have lower power, reduced persistence, a time-varying polarization, as well as time-varying ranges and azimuth/elevation angles. Further differentiation is possible between ground-based and airborne sources. A particularly important direction is to investigate whether incorporating the spatial dimension together with the time and frequency dimensions can significantly improve the ability of GANs to detect anomalous network behavior.

In passing, we would like to state that more research is needed to understand the performance vs.

computational complexity of autoencoders for anomaly detection [125]–[129], especially in terms of their performance for given computational complexity. Tables 4 and 5 provide an interesting comparison in this regard.

The most important lesson learned from this subsection is that the use of GANs result in powerful techniques for anomaly detection in wireless networks. In Sec. IV, we will expand this observation and provide more evidence to this fact by adopting a GAN-based anomaly detection algorithm for NextG wireless applications.

E. Security Applications of GANs in NextG

Another important application area of GANs is wireless security. In these applications, ML can be used to create an attack, defend against an attack, or both. Some specific cases are discussed below.

In [25], a laboratory study is carried out with eight Universal Software Radio Peripheral (USRP) radios as trusted transmitters. The goal is to use the imbalance in the detected I/Q components due to unique hardware differences in each transmitter for fingerprinting, i.e., identifying which transmitter is which. The authors use the generator G of a conventional GAN to create fake transmitters, and then the GAN classifier attempts to distinguish between real and fake transmitters, achieving an accuracy of 99.9%. The authors also develop approaches to distinguish between the legitimate transmitters, one based on a CNN that obtains an accuracy of 81.6%, and another based on a Deep Neural Network (DNN) that scores 96.6%. Although this is a simple experiment, it shows the power of ML in wireless security.

A wireless spoofing algorithm based on a GAN architecture is proposed in [9], split between an adversary transmitter A_T and an adversary receiver A_R placed close to the actual receiver R . Feedback from A_R based on its location allows A_T to know the channel between the actual transmitter T and R as well as between A_T and R with sufficiently high accuracy. Then, a GAN is implemented between A_T and A_R in which the generator G is trained at A_T and the discriminator D is implemented at A_R . A_T in turn adjusts its transmission parameters such that its signal will appear to R as if they come from T . In this process, A_R trains the discriminator D such that the classification error is minimized, i.e., it attempts to achieve

$$\min_D \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] - \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log(D(\mathbf{x}))]. \quad (18)$$

At the same time, A_T trains the generator G such that the classification error is maximized, i.e., it

attempts to achieve

$$\max_G \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (19)$$

The process is continued until convergence. Thus, although they are at different locations, A_T and A_R train G and D while playing the following minimax game

$$\min_D \max_G \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] - \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log(D(\mathbf{x}))]. \quad (20)$$

Simulation results show that, while the probability of successful spoofing is only 7.9% for random signals and 36.2% in an amplify-and-forward architecture, the GAN is able to achieve a success rate of 76.2%.

The problem of thwarting an Intrusion Detection System (IDS) is studied in [26]. While this work is not specifically about wireless networks, the basic results are still applicable to wireless systems. While many ML approaches have been proposed for IDS, the authors of [26] show that such systems are vulnerable to an attack employing a GAN. On the other hand, the paper also shows that if the IDS system based on conventional ML is replaced by one based on GANs, it can be made more robust against adversarial perturbations.

In [27], an interesting problem involving two Unmanned Aerial Vehicles (UAVs) trying to communicate in the presence of an active jammer that eavesdrops their transmissions and jams only when the two are at communicating in the same frequency band is described. A three-way GAN is developed with the generator G present at the jammer and two classifiers D_s and D_r present, one at each UAV. The generator G operates without access to the classifiers. For reasons of stability, [27] does not employ the conventional log-based formulation for driving the backpropagation algorithm, but instead, another version based on a least squares formulation called LSGAN [31]. The performance of the algorithm is compared against two GAN-based algorithms with three players and one non-GAN-based game theoretic approach. The algorithm in [27] is shown to outperform the other three in terms of average connection latency, attack probability, and packet delivery ratio in the presence of channel switching and jamming. This paper shows the sophistication of GAN-based approaches in wireless applications.

In addition to the GAN-based wireless security work described above, there are many other papers in the literature that cover various aspects of wireless security by employing other ML techniques, in terms of both designing attacks and ways to mitigate them. A sample listing is [7], [10], [130]–[151].

For spectrum sharing applications, not only traditional security threats such as receiver jamming, but also newly emerged cognitive-radio-specific security threats such as PU Emulation Attacks

(PUEA), SSDF, common control channel jamming, selfish users, and intruding nodes should be considered. A detailed description of these threats can be found in [152]–[156].

Cognitive radio networks rely on a trustworthy spectrum sensing process, the key to which is the ability to distinguish PU signals from SU signals in a robust way. The use of licensed spectrum bands by PUs may be sporadic, so an SU must constantly monitor for the presence of PUs in candidate bands. If an SU detects the presence of a PU in the band under observation, it cannot use the current band. If there is no PU currently active in the band, then SUs employ a medium access control mechanism to share the band. In general, it is expected that no change will be required on the part of the PU transmission system or air interface in order to accommodate the SUs, e.g., [157], and thus it is the responsibility of the SUs to correctly identify PUs with their existing air interface. As a result, an SU’s ability to correctly identify PUs becomes very important not only for avoiding interference to the PUs, but also to be able to increase their own throughput. Distinguishing whether or not a user is a PU is nontrivial, especially when rogue users modify their air interface in order to mimic a PU’s signal characteristics. This is known as a PUEA [158].

In order to mitigate the PUEA problem, it is possible to draw parallels with a phenomenon known for adversarial attacks against classifiers that employ neural networks. This phenomenon became apparent in classification problems in conventional fields such as image processing, where it was observed that certain minor variations in the input, undetectable by humans, can make a classifier fail, see e.g., [159], [160]. An example that appeared in [160] is very well-known. Using a 22-layer deep CNN that is able to correctly identify a panda with 57.7% accuracy, adding a slight amount of noise results in it being characterized as a gibbon with 99.3% confidence. This vulnerability of deep neural networks to adversarial attacks has been well-documented, see e.g., [161]–[165].

A number of approaches exist to deal with the problem of adversarial attacks in the context of image processing; a good review is available in [166]. There is a recent study which finds that generative classifiers, such as GANs, are actually more robust to adversarial attacks [167]. Given the success of ML techniques developed for image processing and successfully adapted to problems in wireless networking, it is possible to leverage some of this prior work in the context of designing GAN-based ML systems to counteract adversarial attacks. The first such technique is described in [168], and is referred to as Defense-GAN. In this technique, the GAN is trained to model the distribution of unperturbed data. At inference time, it finds an output that does not contain any adversarial changes that is “close” in some sense to the given input data. This output is then fed to the classifier, which can employ any model, and the result is used as the classification of the data.

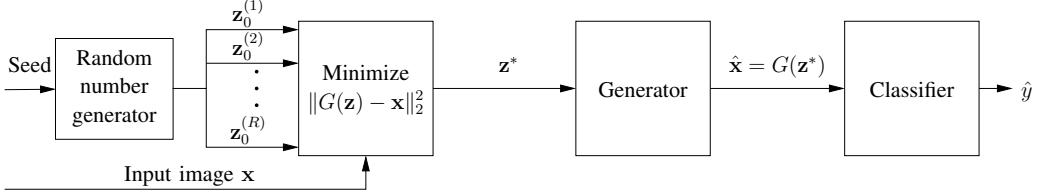


Fig. 7: Defense-GAN algorithm [168].

This technique does not assume knowledge of the process for generating the adversarial examples. Reference [168] empirically shows that Defense-GAN is consistently effective against different attack methods and improves on existing defense strategies.

Fig. 7 illustrates the basic idea in [168]. For a given input observation \mathbf{x} , the system finds the closest element (in the L_2 norm) of the data space that can be produced by the generator operating on an element of the latent space. The resulting element of the data space, presumably free from the affects of the adversarial perturbation, is then classified in the regular way. Intuitively, if \mathbf{x} is not perturbed, then the GAN can generate a sufficiently close local copy of it to be classified as “real.” On the other hand, a perturbed \mathbf{x} will not be among the possible outputs of the generator G trained by using unperturbed data, and the classifier will declare it as “fake.” There is a mathematical development that justifies this intuition in [168]. The paper shows that the Defense-GAN technique is very effective against both black-box and white-box attacks, i.e., attacks that do not have any knowledge of G and the classifier C , as well as those that do have full knowledge of G and C , respectively. Other works that use a similar concept based on GANs are available in [169], [170].

In [10], the effects of a GAN-based spoofing attack to generate synthetic wireless signals that cannot be statistically distinguished from intended transmissions is studied. The spoofing attack can be used for various adversarial purposes such as emulating PUs in cognitive radio networks and fooling signal authentication systems to intrude protected wireless networks. The adversary is modeled as a pair of transmitter and a receiver that build the generator and discriminator of a GAN, respectively. The adversary transmitter trains a deep neural network to generate the best spoofing signals and fool the best defense trained as another deep neural network at the adversary receiver. Thus, the generator and the discriminator of the GAN are at different locations, collaborating over the air such that the GAN-generated signals cannot be reliably discriminated from intended signals. The adversary and the defender may have multiple transmitter or receiver antennas. Spoofing is accomplished by jointly capturing waveform, channel, and radio hardware effects inherent to wireless signals under attack. It is demonstrated in [8] that a GAN-based spoofing attack can be successfully performed and its performance is compared with random signal attack and replay attack. An important observation is that as the attacker’s transmitter gets close to the transmitter, the likelihood of the

# of defense operations / # of all samples	Attacker error probabilities		Transmitter performance	
	Mis detection	False alarm	Throughput	Success ratio
0% (no defense)	3.95%	18.10%	0.012	2.91%
10%	20.79%	25.16%	0.074	17.13%
20%	33.88%	40.69%	0.124	28.84%
30%	40.09%	44.79%	0.170	35.42%
40%	45.18%	43.75%	0.206	41.53%
50%	41.63%	45.10%	0.204	39.23%

Table 6: Results for defensive strategies against jamming attacks based on adversarial learning [134].

attacker to generate high-fidelity synthetic signals for the spoofing attack increases. In summary, the success probability of the GAN-based spoofing attack is very high. This holds for different network topologies and when node locations change from training to test time. This probability improves further when multiple antennas are used at the transmitter.

It is possible to devise defensive strategies against such attacks. A simple strategy is to change the idle and busy labels, so that the legitimate transmitter may not transmit if it senses the channel as idle, or it may transmit even if the channel is sensed as busy [7], [134], assuming that the unobserved transmitter is not a PU. This can be done over a subset of the transmit opportunities. For example, by introducing 10% of false labels, the legitimate transmitter, as the defender, can increase the misdetection performance of the attacker from 3.95% to 20.79% and the false alarm rate from 18.10% to 25.16%. The results for a single-input single-output (SISO) scenario from [134] are shown in Table 6. Clearly, this simple algorithm can result in a significant improvement.

It is possible to extend this work to the case where NACKs, in addition to ACKs, are employed. ACKs are only positive acknowledgements, they do not reveal if there was a transmission. NACKs state there was a transmission but it was not received correctly, possibly due to a collision, but also potentially due to other effects such as fading. This increases the state space of the learning algorithm and may improve the performance of the attacker. A potential direction is to investigate if there is a simple algorithm similar to the one above that provides a defense against such attacks.

For the case of cooperative spectrum sensing, a recent study makes adversarial use of ML to construct a surrogate of the fusion center’s decision model [171]. The authors then propose an algorithm to create malicious sensing data, and show that this type of attack is very effective. Reference [171] shows, *via experiments*, that with existing defenses it can achieve up to an 82% success ratio while only manipulating a small number of malicious nodes. Then, a mechanism referred to as an influence-limiting policy is introduced which achieves a disruption ratio reduction of up to 80% of the attacks introduced in [171]. An important observation that can be made based on this work is that, as in many security problems, specific types of attacks require specific types of solutions.

Reference	Application	Comment
[8] [80]	Spectrum Sharing	Spectrum sensing with domain adaptation and data augmentation. AMC solution via modified CGAN and semi-supervised AC-GAN.
[103] [114] [120] [123]	Anomaly Detection	Use of BiGAN and PCA for anomaly detection. Comparisons in Table 3. Unsupervised spectrum anomaly detection via adversarial autoencoder. For mmWave. Comparison of VAE, CGAN, AC-GAN in Tables 4-5. Modulation classification via CNN: Idle, in- or out-network, jammer.
[25] [9] [26] [27] [168] [10] [7]	Security	Create and then classify fake transmitters via GAN. 99.9% accurate. Spoofing with a GAN where its G and D are not colocated. High success rate. Intrusion detection with improved attack or defense by GANs. Three-way GAN among a jammer and two UAVs trying to communicate. Defense-GAN (Fig. 7). Effective against both black- and white-box attacks. Effects of GAN-based spoofing to generate synthetic signals like real ones. Strategy based on changing labels against GAN-based spoofing attacks. Table 6.

Table 7: Different implementations of techniques using GANs for spectrum sharing, anomaly detection, and security applications in NextG. The main characteristic of each application is specified as a comment. For a detailed discussion of each of the techniques, see the text. PCA: Principal Component Analysis, CNN: Convolutional Neural Network, UAV: Unmanned Aerial Vehicle.

It is possible to pursue attacks similar to those in [171] and attempt to develop relevant solutions. However, it should be emphasized that a key aspect of any defense against a particular attack is first detecting that an attack has occurred, which is exactly the anomaly detection problem discussed above. Strong anomaly detection is the first step in mitigating attacks in spectrum sensing problems, both cooperative and noncooperative. To that end, it is worthwhile to investigate the effectiveness of the GAN-based anomaly detection techniques discussed in Sec. II-A against a number of different attacks such as those described above.

While many conventional countermeasures against security attacks in cognitive radio networks exist [172], there is very limited published work on ML techniques for this purpose [173]. Thus, this is a potentially very fruitful research area.

The most important lesson learned from this subsection is that GANs can be used for security attacks or as a mechanism for defense against any security attacks. We have discussed seven works from the literature towards both ends.

Table 7 summarizes all of the works from the literature discussed in Sec. III.

IV. SIMULATION RESULTS

The problem of outlier detection in signal classification is a critically challenging one for NextG networks. Signal classification is needed for various NextG applications, including user equipment (UE) identification, spectrum sharing and coexistence (as in the CBRS band), and jammer (interference) detection. Specifically, for the case where there is no data about unknown waveforms (that can be considered as outliers), the signal classification problem becomes even more challenging. In this case, in general, the outlier detection algorithms are trained using only inlier waveforms and

tested with inliers and outliers. Recent studies have shown that GANs can help solve the outlier detection problem. AnoGAN has been proposed as an unsupervised anomaly detection algorithm to calculate an anomaly score [35]. Despite its good performance and relaxed assumption that does not require labeled data, it suffers from instabilities in GAN training. In addition, AnoGAN is an iterative algorithm which hinders its real-time application. Fast AnoGAN (f-AnoGAN) extends AnoGAN by using a more stable GAN architecture, WGAN, and it does not require iterations (only one forward pass) [174]. F-AnoGAN uses an encoder to obtain the latent features of a generator. The latent features are input to the generator. The anomaly score is calculated as the mean squared error (MSE) of the input sample and its reconstruction $\|\mathbf{x} - G(E(\mathbf{z}))\|_2^2$ plus the MSE of the discriminator features of the original and those of the reconstructed signal, $\|D_f(\mathbf{x}) - D_f(G(E(\mathbf{x})))\|_2^2$, weighted by a scaling factor κ .

This approach can be applied to anomaly detection in RF spectrum data. As an initial result, we used the RFMLS 2016a dataset [124] that includes eleven different modulations collected over a wide SNR range between -20 dB and 18 dB. The f-AnoGAN algorithm is trained on only one (known) modulation and evaluated using all eleven modulations. We expect the model to label samples from the known (trained) modulation as inlier and samples from all others as outliers. In contrast to [33] that uses WGAN, we used the WGAN-GP which is known to be more stable. The f-AnoGAN model outputs an anomaly score per sample and the AUROC figures are calculated.

In our evaluations, the f-AnoGAN model is trained in three steps:

- 1) First, we trained the generator and discriminator of the WGAN-GP model, both networks are summarized in Table 9 and Table 10 in the Appendix, respectively.
- 2) Second, we trained the encoder (architecture presented in Table 11 in the Appendix).
- 3) Finally, we evaluated the performance of the anomaly detection using the anomaly score.

This three-step procedure closely follows the f-AnoGAN implementation of [174].¹

In evaluating the fidelity and diversity of the generative models, we compared the use of five different measures [53]: *i*) sum of recall and precision (S-RP), *ii*) harmonic mean of recall and precision (H-RP), *iii*) sum of density and coverage (S-DC), *iv*) harmonic mean of density and coverage (H-DC), and *v*) Jensen-Shannon distance (JSD) [175]. In the GAN training, these measures are evaluated every 10 epochs and if the fidelity/diversity measure has improved, the model is saved. For the JSD metric, a lower value indicates a better model, whereas for the remaining four measures,

¹<https://github.com/tSchlegl/f-AnoGAN>

a higher value means a better model. We used 500 epochs for each of the GAN and encoder training steps. For all three networks, we used the Adam optimizer with a learning rate of 0.0002 [176].

As a comparison, we evaluated the performance of a convolutional autoencoder (CAE) system which is again trained on the only inlier modulation. The CAE architecture is presented in Table 12 in the Appendix. We measured the mean and standard deviation of the reconstruction loss in the training dataset. For any test data that has a reconstruction loss larger than a fixed threshold (i.e., the mean reconstruction loss plus one standard deviation), the sample is labeled as an outlier and for those samples with a reconstruction loss smaller or equal to threshold, the sample is labeled as an inlier. Note that to have a fair assessment between performance measures and minimize the effects of network architecture, we have used a CAE model that is very similar to the AE model of the f-AnoGAN.

Table 8 presents the AUROC values for the anomaly detection problem employing different modulations. We observe that the performance of the anomaly detection model depends significantly on the trained (inlier) modulation in both GAN-based and CAE-based approaches. There are several reasons for this. First, unlike image datasets, this dataset (and many other RF datasets) include the effects of channel, noise, and hardware impairments, which make the classification problem fundamentally challenging. Second, some modulations in this dataset are subsets of each other in the constellation plot. For example, AM-SSB modulation is a subset of the AM-DSB modulation. In comparing the results in Table 8, we see that the H-RP, JSD, and S-RP measures perform well across different modulations, whereas the S-DC and H-DC metrics did not provide good performance. We observe that anomalies can be precisely detected (achieving more than a 0.90 AUROC score) when f-AnoGAN models are trained on modulations such as the AM-DSB, CPFSK, GFSK, and WBFM using any of the S-RP, H-RP, and JSD measures. In these modulations, the f-AnoGAN method outperforms the CAE method. However, modulations such as AM-SSB and QAM16 present challenging problems as they are very similar to other classes (AM-DSB and QAM64, respectively). For these modulations, the f-AnoGAN method fails to reliably distinguish the anomalies from inliers, whereas the CAE model performs only as good as a random classifier.

Next, we discuss some implementation details. For the WGAN-GP training, there are two important parameters that we have tuned. The first parameter is the number of discriminator (critic) iterations per generator iteration. We have tested 1, 3, and 5 iterations and found that updating the generator every 3 discriminator updates provided the best results. The second parameter is the gradient penalty coefficient λ . This parameter had a profound effect on the performance. We tested different λ values

Modulation	S-RP	H-RP	S-DC	H-DC	JSD	CAE
AM-DSB	0.963	0.964	0.962	0.965	0.968	0.899
AM-SSB	0.088	0.088	0.088	0.088	0.273	0.502
BPSK	0.797	0.777	0.349	0.716	0.439	0.529
8PSK	0.558	0.597	0.540	0.540	0.636	0.528
CPFSK	0.968	0.970	0.853	0.732	0.939	0.533
GFSK	0.983	0.980	0.821	0.816	0.917	0.561
PAM4	0.692	0.728	0.041	0.707	0.731	0.587
QAM16	0.419	0.424	0.300	0.300	0.447	0.600
QAM64	0.549	0.624	0.285	0.285	0.545	0.597
QPSK	0.664	0.678	0.547	0.547	0.559	0.530
WBFM	0.937	0.938	0.934	0.933	0.939	0.824
Average	0.693	0.706	0.520	0.603	0.672	0.608

Table 8: AUROC scores of the anomaly scores obtained using five different fidelity measures for different modulations. S-RP stands for sum of precision and recall, H-RP is the harmonic mean of precision and recall, S-DC stands for sum of density and coverage, H-DC is the harmonic mean of density and coverage, and JSD stands for Jensen-Shannon distance.

$(\lambda = 1, 5, 10)$ and $\lambda = 10$ provided the best results, which emphasizes the importance of the gradient penalty. For the f-AnoGAN model, the scaling factor κ is a parameter that can be tuned. We have tested $\kappa = 1, 5, 10$ values and found that $\kappa = 1$ provided the best results. Finally, as the f-AnoGAN approach requires only one forward pass and does not need any iteration to infer a sample, we can obtain the average inference time on hardware, which is an important measure for the embedded implementation. Note that this is in contrast to iterative algorithms such as AnoGAN that require setting parameters such as the maximum number of iterations to obtain precise timing measures. Towards this goal, we measured the average end-to-end inference time on hardware, which includes the forward pass of the input sample to the encoder, generator, and discriminator networks, and calculating the respective losses to obtain the anomaly score. The results are evaluated on an NVIDIA GeForce RTX 3060 GPU system. The average inference time is measured as 0.005784 seconds which enables us to process 172.9 samples per second without any parallel processing.

In concluding this section, we would like to emphasize that our simulations can only be considered as very initial results. To be able to come to more in-depth conclusions, substantially more work by the research community should be carried out.

V. FUTURE RESEARCH DIRECTIONS FOR GANs IN NEXTG COMMUNICATION SYSTEMS

This section is on the use of GANs for wireless applications in general terms. This is important because the original development of GANs was in the area of images while their straightforward application to wireless can bring up questions. On the other hand, this feature actually makes the proposed research nontrivial and interesting. It should be understood that what we are proposing is not a simple translation or porting of what exists in another field to wireless communications

but rather a very careful study of how to be inspired from a development in one field and do a careful analysis to adapt it to a completely different area. For example, performance measures used to compare two images do not directly translate into wireless applications. In particular, performance measures used in image processing try to imitate human perception. Clearly, it is questionable that they would be useful in the kind of tasks discussed above in sections Sec III-C–Sec III-E. If this is the case, then what kind of performance measures should be employed? Since ML is not a model-based approach, finding the answer to this question will require experimentation and thus is not a straightforward task. Another consideration comes from the fact that GANs are known to be difficult to train, especially for datasets other than images. Then, is there a way this difficulty can be circumvented? In terms of datasets, there is a perception that datasets that can be employed for wireless applications are limited in quantity. Furthermore, in general, training of neural networks require very long training sequences. On the other hand, the datasets available are limited in size. Some researchers employ GANs to increase the size of the their training sequences in a synthetic manner. But then there is a question as to whether the generated or augmented synthetic signals contain any more information than the training set since the GAN is able to learn the statistics of the training sequence but it is not clear if this augmentation process can add any new statistics. These are some of the questions that employing algorithms from the basic exploration of GANs to their use in wireless applications will face.

A number of open problems remain in this domain. While a large number of problems in spectrum sharing, anomaly detection, and security exist, studies using GANs to address them have so far targeted a relatively small number of specific problems. Scaling these solutions to a larger number of problems will require significant effort. First, specific problems need to be identified, for example, in anomaly detection and security, to be addressed by ML algorithms. Second, these specific problems need to be simulated and ML algorithms need to be trained on them. Third, the computational complexity of this effort should be taken under consideration. There are many successful uses of GANs, especially in image processing. While a number of these algorithms have been successfully used in wireless applications, a number of them are still candidates for use in that setting. For example, we have discussed the use of BiGANs and f-AnoGANs for anomaly detection, however, a number of other anomaly detection techniques such as EGBAD and GANomaly have not yet been studied in the context of wireless anomaly detection. Fourth, reliable diversity and fidelity measures for GANs to quantify the quality of the synthetic data for RF applications need to be further explored. As it is difficult to conduct statistical analyses over complex and high-dimensional data, these metrics

would provide the necessary reliable methods to perform hyper-parameter optimization in training GANs and fairly compare different GAN architectures. Towards this goal, we have investigated five of such metrics in this paper and showed that HR-P provided the best results for the outlier detection problem.

Yet on another subject we touched upon previously, the topic of augmentation is controversial. Some researchers strongly hold the belief that GAN-generated signals cannot be gainfully employed to augment the training dataset. However, an interesting piece of news from NVIDIA, discussed in [177], together with a short video worthwhile seeing, states that it is possible to reduce the training sequence length 10x-20x by using data augmentation and GANs. The particular application is the creation of synthetic images from real ones and therefore the visual effect is strong. This was achieved by employing a special technique with GANs, explained in [178]. Additional works show that, by using special techniques, it is possible to achieve data augmentation outside the statistics of the short training sequence, see, e.g., [179]–[181]. There exist publications that make this statement specifically for wireless applications, see, e.g., [13], [54], [80].

An inevitable question arises when GANs are considered as a member of a large class of ML algorithms. As shown in this paper, they have advantages in terms performance although it is not clear that they are always the best. Considerations such as complexity, accuracy, robustness, etc. should be taken into account in comparing them with other ML algorithms for use in the wireless applications considered in this paper. To the best of our knowledge, such a study does not exist. Any step taken towards this understanding will be very valuable.

GANs can be applied to a number of emerging areas in communications. An example of this is communications in the mmWave and THz bands. As long as the computational complexity of the solution can be managed, what was discussed in this paper is applicable to communication networks in those bands. We will discuss a number of works from the literature related to this fact. Reference [182] discusses channel estimation for very large bandwidths operating in mmWave and THz bands at low SNR. It proposes a GAN-based estimator that first learns to produce channel samples from the unknown channel distribution via training the generative network, and then uses this trained network as a prior to estimate the current channel. This estimator has been shown to work at an SNR of -5 dB better than Least Squares (LS) and Linear Minimum Mean Square Estimate (LMMSE) estimators at SNR values of 20 dB and 2.5 dB, respectively. Furthermore, this estimator reduces the required number of pilots and does not require retraining even if the number of clusters and rays change considerably. Channel estimation in the mmWave band is further discussed for the unmanned aerial

vehicle (UAV) applications in [183], [184] and for vehicular systems in [185].

Another emerging area in communications is mobile edge computing. This is an area GANs have applications in security. As discussed in [186], GANs can be used to design a security attack for mobile edge computing. In this attack, a malicious participant can infer sensitive information from a victim participant even with just a part of shared parameters from the victim [187], [186, Fig. 13]. For other applications of GANs in mobile edge computing, in security and privacy, see, e.g., [187], [188].

VI. CONCLUSION

The proliferation of wireless services, in the forms of wireless local area networks, cellular wireless, applications of the IoT, the tactile internet, autonomous cars, drones, factory automation, and the general expectations under the umbrella of the Fourth Industrial Revolution place enormous demands on wireless spectrum. Although new spectrum in the mmWave band is already being regulated around the world, the expectations are such that spectrum scarcity will be with us beginning in the very near future. Many experts in the field expect that forms of spectrum sharing will be needed towards that purpose. It is well known that a common technique under consideration for dynamic spectrum sharing is cognitive radio which can be used to address the problem of spectrum sharing. However, conventional forms of cognitive radio will likely not be sufficient because of the enormous dynamics of the problem. It is expected that the complexity of the problem will prohibit conventional cognitive radio and forms of ML will have to be employed towards solving this very difficult problem.

This paper addressed a particular ML solution, GANs, towards that end. GANs are capable of generating “fake” data that cannot be differentiated from “real” data, and thus to some extent they capture the distribution of real signals. They are known to address competitive resource allocation problems. Furthermore, they are known to be effective in detecting and mitigating anomalous behavior. A survey of the use of GANs in spectrum sharing, anomaly detection, and addressing security concerns is provided in this paper. In several cases discussed, it is demonstrated that GANs have better performance in addressing the concerns stated above than the other ML algorithms.

It is generally accepted that wireless applications are increasing in quantity while wireless spectrum is scarce and that ML solutions to address this problem are needed. GANs provide a framework for such a set of solutions with proven success. Nevertheless, more work is needed to identify specific problems and develop successful ML solutions, including those that involve GANs.

VII. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers whose comments improved the presentation in the paper.

APPENDIX

In this section, we provide the parameters of the f-AnoGAN and CAE used in the simulations in Sec. IV.

Layer Name	Details
Reshape layer	$(N, 100)$ to $(N, 100, 1, 1)$
2D transposed convolution layer	1024 filters, kernel size of (2,4), stride of (1,1), no padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D transposed convolution layer	512 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D transposed convolution layer	256 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D transposed convolution layer	128 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D transposed convolution layer	64 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D transposed convolution layer	1 filter; kernel size of (2,4), stride of (2,2), padding

Table 9: Generator architecture.

Layer Name	Details
2D convolution layer	64 filters, kernel size of (2,4), stride of (2,2), padding
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	128 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	256 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	512 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	1024 filters, kernel size of (2,4), stride of (2,2), padding
Reshape layer	$(N, 1024, 2, 4)$ to $(N, 8192)$
Linear layer	8192 neurons to 1 neuron

Table 10: Discriminator architecture.

Layer Name	Details
2D convolution layer	64 filters, kernel size of (2,4), stride of (2,2), padding
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	64 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	128 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	256 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	512 filters, kernel size of (2,4), stride of (2,2), padding
Batch normalization	
Leaky-ReLU	$\alpha = 0.2$
2D convolution layer	1024 filters, kernel size of (2,4), stride of (2,2), padding
Reshape layer	$(N, 1024, 2, 2)$ to $(N, 100)$
Linear layer	4096 neurons to 1 neuron
Tanh	

Table 11: Encoder architecture of the f-AnoGAN system.

Layer Name	Details
2D convolution layer Leaky-ReLU	64 filters, kernel size of (2,4), stride of (2,2), padding $\alpha = 0.2$
2D convolution layer Batch normalization Leaky-ReLU	64 filters, kernel size of (2,4), stride of (2,2), padding $\alpha = 0.2$
2D convolution layer Batch normalization Leaky-ReLU	128 filters, kernel size of (2,4), stride of (2,2), padding $\alpha = 0.2$
2D convolution layer Batch normalization Leaky-ReLU	256 filters, kernel size of (2,4), stride of (2,2), padding $\alpha = 0.2$
2D convolution layer Batch normalization Leaky-ReLU	512 filters, kernel size of (2,4), stride of (2,2), padding $\alpha = 0.2$
2D convolution layer Reshape layer	1024 filters, kernel size of (2,4), stride of (2,2), padding ($N, 1024, 2, 2$) to ($N, 100$)
Reshape layer	($N, 100$) to ($N, 1024, 2, 2$)
2D transposed convolution layer Leaky-ReLU Batch normalization	512 filters, kernel size of (1,2), stride of (1,2), padding $\alpha = 0.2$
2D transposed convolution layer Leaky-ReLU Batch normalization	256 filters, kernel size of (1,2), stride of (1,2), no padding $\alpha = 0.2$
2D transposed convolution layer Leaky-ReLU Batch normalization	128 filters, kernel size of (1,2), stride of (1,2), no padding $\alpha = 0.2$
2D transposed convolution layer Leaky-ReLU Batch normalization	64 filters, kernel size of (1,2), stride of (1,2), no padding $\alpha = 0.2$
2D transposed convolution layer Leaky-ReLU	64 filters, kernel size of (1,2), stride of (1,2), no padding $\alpha = 0.2$
2D transposed convolution layer	1 filter, kernel size of (1,2), stride of (1,2), no padding

Table 12: Convolutional Autoencoder (CAE) architecture.

REFERENCES

- [1] P. Tilghman, "If DARPA has its way, AI will rule the wireless spectrum," *IEEE Spectrum* [Online]. Available: <https://spectrum.ieee.org/telecom/wireless/if-darpa-has-its-way-ai-will-rule-the-wireless-spectrum>, May 28, 2019.
- [2] P. Tilghman, "AI will rule the airwaves," *IEEE Spectrum*, pp. 29–33, Jun. 2019.
- [3] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
- [4] J. Mitola, "Cognitive radio – An integrated agent architecture for software defined radio," Ph.D. dissertation, KTH Royal Institute of Technology, Sweden, 2000.
- [5] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of machine learning to cognitive radio networks," *IEEE Wireless Communications*, vol. 14, no. 4, pp. 47–52, Aug. 2007.
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680, [Online]. Available: <https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>.
- [7] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 1, pp. 2–14, Mar. 2019.
- [8] K. Davaslioglu and Y. E. Sagduyu, "Generative adversarial learning for spectrum sensing," in *2018 IEEE International Conference on Communications (ICC)*, 2018.
- [9] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative adversarial network for wireless signal spoofing," in *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, 2019, pp. 55–60.
- [10] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative adversarial network in the air: Deep adversarial learning for wireless signal spoofing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 294–303, Mar. 2021.
- [11] T. Roy, T. O'Shea, and N. West, "Generative adversarial radio spectrum networks," in *Proc. ACM Workshop on Wireless Security and Machine Learning (WiseML)*, May 2019, pp. 12–15.
- [12] M. Patel, X. Wang, and S. Mao, "Data augmentation with conditional GAN for automatic modulation classification," in *Proc. ACM Workshop on Wireless Security and Machine Learning (WiseML)*, July 2020, pp. 31–36.
- [13] W. H. Clark IV, S. Hauser, W. C. Headley, and A. J. Michaels, "Training data augmentation for deep learning radio frequency systems," arXiv:2010.00178v4 [Online]. Available: <https://arxiv.org/abs/2010.00178v4>, Jan. 2021.
- [14] T. Silva, "An intuitive introduction to Generative Adversarial Networks (GANs)," [Online]. Available: <https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/>, Jan. 2018.
- [15] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," arXiv:1710.10196v3 [Online]. Available: <https://arxiv.org/abs/1710.10196v3>, pp. 1–26, Feb. 2018.
- [16] M. Kan, "NVIDIA's AI creates artificial human faces from celebrity photos," [Online]. Available: <https://www.pcmag.com/news/357068/nvidias-ai-creates-artificial-human-faces-from-celebrity-ph>.
- [17] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations (ICLR)*, Apr. 2018, pp. 1–26, [Online]. Available: https://research.nvidia.com/sites/default/files/pubs/2017-10_Progressive-Growing-of/karras2018iclr-paper.pdf.
- [18] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *International Conference on Vision and Visualization (ICCV)*, Feb. 2017, pp. 1–18, [Online]. Available: arXiv:1703.10593 [Online]. Available: <https://arxiv.org/abs/1703.10593>.
- [19] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," arXiv:1702.01983v2 [Online]. Available: <https://arxiv.org/abs/1702.01983v2>, pp. 1–5, May 2017.

- [20] C. Ledig *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” arXiv:1609.04802v5 [Online]. Available: <https://arxiv.org/abs/1609.04802v5>, May 2017.
- [21] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, “MiDiNet: A convolutional generative adversarial network for symbolic-domain music generation,” arXiv:1703.10847v2 [Online]. Available: <https://arxiv.org/abs/1703.10847v2>, Jul. 2017.
- [22] C. Donahue, J. McAuley, and M. Puckette, “Adversarial audio synthesis,” arXiv:1802.04208v3 [Online]. Available: <https://arxiv.org/abs/1802.04208v3>, Feb. 2019.
- [23] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, “Video-to-video synthesis,” arXiv:1808.06601v2 [Online]. Available: <https://arxiv.org/abs/1808.06601v2>, Dec. 2018.
- [24] D. N. Tran, T. D. Tran, and L. Nguyen, “Generative adversarial networks for recovering missing spectral information,” in *2018 IEEE Radar Conference (RadarConf18)*, Apr. 2018, pp. 1223–1227.
- [25] D. Roy, T. Mukherjee, M. Chatterjee, and E. L. Pasiliao, Jr., “Detection of rogue RF transmitters using generative adversarial nets,” in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2019, pp. 1–7.
- [26] M. Usama, M. Asim, S. Latif, J. Qadir, and A. Al-Fuqaha, “Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems,” in *International Wireless Communications Mobile Computing Conference (IWCMC)*, Jun. 2019, pp. 78–83.
- [27] B. Yang and M. Liu, “Attack-resilient connectivity game for UAV networks using generative adversarial learning,” in *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2019, pp. 1743–1751.
- [28] A. Hindupur, “A list of all named GANs!” [Online]. Available: <https://github.com/hindupuravinash/the-gan-zoo>.
- [29] I. Goodfellow, “NIPS 2016 Tutorial: Generative Adversarial Networks,” arXiv:1701.00160v4 [Online]. Available: <https://arxiv.org/abs/1701.00160v4>, Apr. 2017.
- [30] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” arXiv:1411.1784 [Online]. Available: <https://arxiv.org/abs/1411.1784>, Nov. 2014.
- [31] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least squares generative adversarial networks,” in *Proc. IEEE International Conference on Computer Vision (ICCV)*, Jul. 2017, pp. 2813–2821.
- [32] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proc. International Conference on Machine Learning (ICML)*, Aug. 2017, pp. 214–223.
- [33] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in *Advances in Neural Information Processing Systems*, Dec. 2017, pp. 5767–5777.
- [34] J. Donahue, P. Krhenbhl, and T. Darrell, “Adversarial feature learning,” arXiv:1605.09782v7 [Online]. Available: <https://arxiv.org/abs/1605.09782v7>, Apr. 2017.
- [35] T. Schlegl, P. Seebck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” arXiv:1703.05921 [Online]. Available: <https://arxiv.org/abs/1703.05921>, Mar. 2017.
- [36] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, “Efficient GAN-based anomaly detection,” arXiv:1802.06222v2 [Online]. Available: <https://arxiv.org/abs/1802.06222v2>, May 2019.
- [37] S. Akcay, A. A. Abarghouei, and T. P. Breckon, “GANomaly: Semi-supervised anomaly detection via adversarial training,” arXiv:1805.06725v3 [Online]. Available: <https://arxiv.org/abs/1805.06725v3>, Nov. 2018.
- [38] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” arXiv:1606.03498 [Online]. Available: <https://arxiv.org/abs/1606.03498>, Jun. 2016.
- [39] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with deep generative models,” arXiv:1607.07539v3 [Online]. Available: <https://arxiv.org/abs/1607.07539v3>, Jul. 2017.
- [40] V. Dumoulin, M. I. D. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, “Adversarially learned inference,” arXiv:1606.00704v3 [Online]. Available: <https://arxiv.org/abs/1606.00704v3>, Feb. 2017.

- [41] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi, “A survey on GANs for anomaly detection,” arXiv:1906.11632v1 [Online]. Available: <https://arxiv.org/abs/1906.11632v1>, Jun. 2019.
- [42] D. C. Dowson and B. V. Landau, “The Fréchet distance between multivariate normal distributions,” *Journal of Multivariate Analysis*, vol. 12, pp. 450–455, Sep. 1982.
- [43] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” arXiv:1706.08500v6 [Online]. Available: <https://arxiv.org/abs/1706.08500v6>, Jan. 2018.
- [44] Wikipedia, “Jensen-Shannon divergence,” [Online]. Available: https://en.wikipedia.org/wiki/Jensen_Shannon_divergence
- [45] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, 2006.
- [46] DeepSig, “RF Datasets for Machine Learning,” [Online]. Available: <https://www.deepsig.ai/datasets>.
- [47] DARPA, “Radio Frequency Machine Learning Systems (RFMLS),” [Online]. Available: <https://www.darpa.mil/program/radio-frequency-machine-learning-systems>.
- [48] NIST, “RF Dataset of Incumbent Radar Signals in the 3.5 GHz CBRS Band,” [Online]. Available: <https://www.nist.gov/publications/rf-dataset-incumbent-radar-signals-35-ghz-cbrs-band>, Dec. 2019.
- [49] T. J. O’Shea and N. West, “Machine learning dataset generation with GNU radio,” in *Proc. 6th GNU Radio Conference*, Sep. 2016.
- [50] Wikipedia, “Receiver operating characteristics,” [Online]. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic
- [51] Wikipedia, “Harmonic mean,” [Online]. Available: https://en.wikipedia.org/wiki/Harmonic_mean
- [52] Wikipedia, “F-score,” [Online]. Available: <https://en.wikipedia.org/wiki/F-score>
- [53] N. Naeem, S. J. Oh, J. Uh, Y. Choi, and Y. Yoo, “Reliable fidelity and diversity metrics for generative models,” arXiv:2002.09797v2 [Online]. Available: <https://arxiv.org/abs/2002.09797v2>, Jun. 2020.
- [54] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, “Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks,” *IEEE Access*, vol. 6, pp. 15 713–15 722, 2018.
- [55] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier GANs,” arXiv:1610.09585 [Online]. Available: <https://arxiv.org/abs/1610.09585>, Oct. 2016.
- [56] R. Caromi, M. Souryal, and W.-B. Yang, “Detection of incumbent radar in the 3.5 GHz CBRS band,” in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018, pp. 241–245.
- [57] Y. Arjoune and N. Kaabouch, “A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances new challenges, and future research directions,” *Sensors*, vol. 19, no. 1, pp. 1–32, 2019.
- [58] E. Ghazizadeh, B. Nikpour, D. Abbasi, M. Hossein, and N. Pour, “A PSO-based weighting method to enhance machine learning techniques for cooperative spectrum sensing in CR networks,” in *Proc. Swarm Intelligence and Evolutionary Computation Conference*, Mar. 2016, pp. 113–118.
- [59] Y. Lu, P. Zhu, D. Wang, and M. Fattouche, “Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks,” in *2016 IEEE Wireless Communications and Networking Conference*, Apr. 2016, pp. 1–6.
- [60] D. Wang and Z. Yang, “A novel spectrum sensing scheme combined with machine learning,” in *Proc. International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, Oct. 2016, pp. 1293–1297.
- [61] B. Khalifi, A. Zaid, and B. Hamdaoui, “When machine learning meets compressive sampling for wideband spectrum sensing,” in *Proc. International Wireless Communications and Mobile Computing Conference*, Jun. 2017, pp. 1120–1125.
- [62] G. Ding, Q. Wu, Y. Yao, J. Wang, and Y. Chen, “Kernel-based learning for statistical signal processing in cognitive radio networks: Theoretical foundations, example applications, and future directions,” *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 126–136, Jul. 2013.
- [63] A. M. Mikaeil, B. Guo, and Z. Wang, “Machine learning to data fusion approach for cooperative spectrum sensing,” in *Proc. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Oct. 2014, pp. 429–434.

[64] F. Farmani, A. Jannatabad, and R. Berangi, “Detection of SSDF attack using SVDD algorithm in cognitive radio networks,” in *Proc. Computational Intelligence, Communication Systems and Networks (CICSyN)*, Jul. 2011, pp. 201–204.

[65] Yang Li and Q. Peng, “Achieving secure spectrum sensing in presence of malicious attacks utilizing unsupervised machine learning,” in *IEEE Military Communications Conference (MILCOM)*, Nov. 2016, pp. 174–179.

[66] Z. Cheng, T. Song, J. Zhang, J. Hu, Y. Hu, L. Shen, X. Li, and J. Wu, “Self-organizing map-based scheme against probabilistic SSDF attack in cognitive radio networks,” in *Proc. International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct. 2017, pp. 1–6.

[67] G. Nie, G. Ding, L. Zhang, and Q. Wu, “Byzantine defense in collaborative spectrum sensing via Bayesian learning,” *IEEE Access*, vol. 5, pp. 20089–20098, 2017.

[68] Y. Tang, Q. Zhang, and W. Lin, “A neural network based spectrum prediction scheme for cognitive radioartificial neural network based spectrum sensing method for cognitive radio,” in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, Sep. 2010, pp. 1–4.

[69] B. Varatharajana, E. Praveen, and E. Vinotha, “Neural network aided enhanced spectrum sensing in cognitive radio,” *Procedia Engineering*, vol. 38, pp. 82–88, 12 2012.

[70] D. Han, G. C. Sobabe, C. Zhang, X. Bai, Z. Wang, S. Liu, and B. Guo, “Spectrum sensing for cognitive radio based on convolution neural network,” in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2017, pp. 1–6.

[71] K. Yang, Z. Huang, X. Wang, and X. Li, “A blind spectrum sensing method based on deep learning,” *Sensors*, vol. 19, no. 10, pp. 1–17, 2019.

[72] Q. Cheng and Z. Shi, “Deep learning network based spectrum sensing methods for OFDM systems,” arXiv:1807.09414v4 [Online]. Available: <https://arxiv.org/abs/1807.09414v4>, Jan. 2019.

[73] Z. Ye, Q. Peng, K. Levick, H. Rong, A. Gilman, L. Milstein, and P. Cosman, “Neural network detector for spectrum sensing under uncertainties,” arXiv:1907.07326v2 [Online]. Available: <https://arxiv.org/abs/1907.07326v2>, Aug. 2019.

[74] Q. Zhao and B. M. Sadler, “A survey of dynamic spectrum access,” *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, May 2007.

[75] J. M. Peha, “Sharing spectrum through spectrum policy reform and cognitive radio,” *Proc. IEEE*, vol. 97, no. 4, pp. 708–719, Apr. 2009.

[76] S. M. Baby and M. James, “A comparative study on various spectrum sharing techniques,” in *Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science, and Technology (RAEREST)*, 2016, pp. 613–620, [Online]. Available: <https://doi.org/10.1016/j.protcy.2016.08.152>.

[77] A. Goldsmith, S. Jafar, I. Maric, and S. Srinivasa, “Breaking spectrum gridlock with cognitive radios: An information theoretic perspective,” *Proc. IEEE*, vol. 97, no. 5, pp. 203–213, May 2009.

[78] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[79] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, “Cooperative spectrum sensing in cognitive radio networks: A survey,” *Elsevier Physical Communication*, vol. 4, pp. 40–62, 2011.

[80] M. Li, O. Li, G. Liu, and C. Zhang, “Generative adversarial networks-based semi-supervised automatic modulation recognition for cognitive radio networks,” *Sensors (Basel, Switzerland)*, vol. 18, pp. 3913–3934, Nov. 2018.

[81] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in *Advances in Neural Information Processing Systems*, 2015.

[82] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” arXiv:1506.02025v3 [Online]. Available: <https://arxiv.org/abs/1506.02025v3>, Feb. 2016.

- [83] P. M. Pradhan and G. Panda, "Information combining schemes for cooperative spectrum sensing: A survey and comparative performance analysis," *Springer Wireless Personal Communications*, vol. 94, no. 3, pp. 685–711, Jun. 2017.
- [84] S. K. Yarlagadda, D. Guera, P. Bestagini, F. M. Zhu, S. Tubaro, and E. J. Delp, "Satellite image forgery detection and localization using GAN and one-class classifier," arXiv:1802.04881v1 [Online]. Available: <https://arxiv.org/abs/1802.04881v1>, Feb. 2018.
- [85] C. Wang, Y.-M. Zhang, and C.-L. Liu, "Anomaly detection via minimum likelihood generative adversarial networks," arXiv:1808.00200v1 [Online]. Available: <https://arxiv.org/abs/1808.00200v1>, Aug. 2018.
- [86] S. K. Lim, Y. Loo, N.-T. Tran, N.-M. Cheung, G. Roig, and Y. Elovici, "DOPING: Generative data augmentation for unsupervised anomaly detection with GAN," arXiv:1808.07632v2 [Online]. Available: <https://arxiv.org/abs/1808.07632v2>, Aug. 2018.
- [87] I. Haloui, J. S. Gupta, and V. Feuillard, "Anomaly detection with Wasserstein GAN," arXiv:1812.02463v2 [Online]. Available: <https://arxiv.org/abs/1812.02463v2>, Dec. 2018.
- [88] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-GANomaly: Skip connected and adversarially trained encoder-decoder anomaly detection," arXiv:1901.08954v1 [Online]. Available: <https://arxiv.org/abs/1901.08954v1>, Jan. 2019.
- [89] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, "Image anomaly detection with generative adversarial networks," in *Proc. Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Jan. 2019, pp. 3–17.
- [90] C. P. Ngo, A. A. Winarto, C. K. K. Li, S. Park, F. Akram, and H. K. Lee, "Fence GAN: Towards better anomaly detection," arXiv:1904.01209v1 [Online]. Available: <https://arxiv.org/abs/1904.01209v1>, Apr. 2019.
- [91] Y. Intrator, G. Katz, and A. Shabtai, "MDGAN: boosting anomaly detection using multi-discriminator generative adversarial networks," arXiv:1905.01430v1 [Online]. Available: <https://arxiv.org/abs/1905.01430v1>, May 2019.
- [92] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," arXiv:1802.06222v2 [Online]. Available: <https://arxiv.org/abs/1802.06222v2>, May 2019.
- [93] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "BeatGAN: Anomalous rhythm detection using adversarially generated time series," in *Proc. Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, Aug. 2019, pp. 4433–4439.
- [94] D. Li, D. Chen, J. Goh, and S.-K. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," arXiv:1809.04758v3 [Online]. Available: <https://arxiv.org/abs/1809.04758v3>, Jan. 2019.
- [95] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," arXiv:1901.04997v1 [Online]. Available: <https://arxiv.org/abs/1901.04997v1>, Jan. 2019.
- [96] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, "One-class adversarial nets for fraud detection," arXiv:1803.01798v2 [Online]. Available: <https://arxiv.org/abs/1803.01798v2>, Jun. 2018.
- [97] Y.-J. Zheng, X.-H. Zhou, W.-G. Sheng, Y. Xue, and S.-Y. Chen, "Generative adversarial network based telecom fraud detection at the receiving bank," *Elsevier Neural Networks*, vol. 102, pp. 78–86, Jun. 2018.
- [98] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Elsevier Information Sciences*, vol. 479, pp. 448–455, Apr. 2019.
- [99] M. Schreyer, T. Sattarov, C. Schulze, B. Reimer, and D. Borth, "Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks," arXiv:1908.00734v1 [Online]. Available: <https://arxiv.org/abs/1908.00734v1>, Aug. 2019.
- [100] Y. Qiu, T. Misu, and C. Busso, "Driving anomaly detection with conditional generative adversarial network using physiological and CAN-Bus data," in *2019 International Conference on Multimodal Interaction*, Oct. 2019, pp. 164–173, [Online]. Available: <https://dl.acm.org/doi/10.1145/3340555.3353749>.
- [101] F. Gokgoz, "Anomaly detection using GANs in OpenSky network," in *NATO S&T Organization, STO-MP-IST-160*, 2018, pp. W2–2.1–W2–2.7.

[102] M. DuCoffe, I. Haloui, and J. S. Gupta, "Anomaly detection on time series with Wasserstein GAN applied to PHM," *Int. Journal of Prognostics and Health Management (Special Issue on Deep Learning and Emerging Analytics)*, vol. 10, pp. 1–12, Apr. 2019, [Online]. Available: <https://www.phmsociety.org/node/2665>.

[103] R. Patil, R. Biradar, V. Ravi, P. Biradar, and U. Ghosh, "Network anomaly detection using PCA and BiGAN," *Internet Technology Letters*, pp. 1–6, Sep. 2020, [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.235>.

[104] "KDD Cup 1999 Data," [Online]. Available: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

[105] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," in *Proc. 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48, Jun. 2016, pp. 1100–1109.

[106] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. International Conference on Learning Representations (ICLR) 2018*, May 2018, pp. 1–7, [Online]. Available: <https://openreview.net/forum?id=BJJLHbb0->.

[107] M. Afgani, S. Sinanovic, , and H. Haas, "The information theoretic approach to signal anomaly detection for cognitive radio," *International Journal of Digital Multimedia Broadcasting*, vol. 2010, pp. 1–18, 2010.

[108] J. Blesa, E. Romero, A. Rozas, and A. Araujo, "PUE attack detection in CWSNs using anomaly detection techniques," *EURASIP Journal on Wireless Communications and Networking*, pp. 1–13, 2013.

[109] W. Honghao, J. Yunfeng, and W. Lei, "Spectrum anomalies autonomous detection in cognitive radio using hidden Markov models," in *IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Dec. 2015, pp. 388–392.

[110] Q. Yang, X. Zhang, J. Qian, and Q. Ye, "An anomaly node detection method for distributed time synchronization algorithm in cognitive radio sensor networks," *International Journal of Distributed Sensor Networks*, vol. 14, no. 5, pp. 1–9, 2018.

[111] C. Wu and Y. Guo, "Cognitive sensing: Adaptive anomalies detection with deep networks," in *Applied Cloud Deep Semantic Recognition*. Taylor & Francis Group, 2018, pp. 57–85.

[112] Q. Dong, "Primary user emulation attack detection in cognitive radio networks," Ph.D. dissertation, Binghamton University, State University of New York, 2018.

[113] M. S. Khan, M. Jibran, I. Koo, S. M. Kim, and J. Kim, "A double adaptive approach to tackle malicious users in cognitive radio networks," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–9, 2019.

[114] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, "SAIFE: Unsupervised wireless spectrum anomaly detection with interpretable features," arXiv:1807.08316v1 [Online]. Available: <https://arxiv.org/abs/1807.08316v1>, Jul. 2018.

[115] N. Tandiya, A. Jauhar, V. Marojevic, and J. H. Reed, "Deep predictive coding neural network for rf anomaly detection in wireless networks," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.

[116] Z. Li, Z. Xiao, B. Wang, B. Y. Zhao, and H. Zheng, "Scaling deep learning models for spectrum anomaly detection," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 291–300.

[117] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, "Unsupervised wireless spectrum anomaly detection with interpretable features," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, pp. 637–647, 2019.

[118] S. Rajendran, V. Lenders, W. Meert, and S. Pollin, "Crowdsourced wireless spectrum anomaly detection," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, pp. 694–703, Jun. 2020.

[119] A. Toma, A. Krayani, M. Farrukh, H. Qi, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "AI-based abnormality detection at the PHY-Layer of cognitive radio by learning generative models," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, pp. 21–34, Mar. 2020.

[120] A. Toma, A. Krayani, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "Deep learning for spectrum anomaly detection in cognitive mmWave radios," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–7.

- [121] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” arXiv:1511.05644v2 [Online]. Available: <https://arxiv.org/abs/1511.05644v2>, May 2016.
- [122] D. P. Kingma and M. Welling, “An introduction to variational autoencoders,” arXiv:1906.02691v3 [Online]. Available: <https://arxiv.org/abs/1906.02691v3>, Dec. 2019.
- [123] Y. Shi, K. Davaslioglu, Y. E. Sagduyu, W. C. Headley, M. Fowler, and G. Green, “Deep learning for RF signal classification in unknown and dynamic spectrum environments,” arXiv:1909.11800v1 [Online]. Available: <https://arxiv.org/abs/1909.11800v1>, Sep. 2019.
- [124] T. O’Shea, J. Corgan, and C. Clancy, “Convolutional radio modulation recognition networks,” in *Proc. International Conference on Engineering Applications of Neural Networks*, 2016.
- [125] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, “Deep autoencoding models for unsupervised anomaly segmentation in brain MR images,” arXiv:1804.04488v1 [Online]. Available: <https://arxiv.org/abs/1804.04488v1>, Apr. 2018.
- [126] L. Beggel, M. Pfeiffer, and B. Bischl, “Robust anomaly detection in images using adversarial autoencoders,” arXiv:1901.06355v1 [Online]. Available: <https://arxiv.org/abs/1901.06355v1>, Jan. 2019.
- [127] H. S. Vu, D. Ueta, K. Hashimoto, K. Maeno, S. Pranata, and S. M. Shen, “Anomaly detection with adversarial dual autoencoders,” arXiv:1902.06924 [Online]. Available: <https://arxiv.org/abs/1902.06924>, Feb. 2019.
- [128] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, “GEE: A gradient-based explainable variational autoencoder for network anomaly detection,” arXiv:1903.06661v1 [Online]. Available: <https://arxiv.org/abs/1903.06661v1>, Mar. 2019.
- [129] S. Ger and D. Klabjan, “Autoencoders and generative adversarial networks for anomaly detection for sequences,” arXiv:1901.02514v3. Available: <https://arxiv.org/abs/1901.02514v3>, Sep. 2019.
- [130] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan, “Machine learning in wireless sensor networks: Algorithms, strategies, and applications,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1996–2018, Fourth Quarter 2014.
- [131] K. Youssef, L.-S. Bouchard, K. Z. Haigh, H. Krovi, J. Silovsky, and C. P. V. Valk, “Machine learning approach to RF transmitter identification,” arXiv:1711.01559v2 [Online]. Available: <https://arxiv.org/abs/1711.01559v2>, Nov. 2017.
- [132] A. Weinand, M. Karrenbauer, R. Sattiraju, and H. D. Schotten, “Application of machine learning for channel based message authentication in mission critical machine type communication,” arXiv:1711.05088v1 [Online]. Available: <https://arxiv.org/abs/1711.05088v1>, Nov. 2017.
- [133] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. F. Moura, “A deep learning approach to IoT authentication,” in *IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.
- [134] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, “Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies,” in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6.
- [135] B. Chatterjee, D. Das, S. Maity, and S. Sen, “RF-PUF: Enhancing IoT security through authentication of wireless nodes using in-situ machine learning,” arXiv:1805.01374v3 [Online]. Available: <https://arxiv.org/abs/1805.01374v3>, Jun 2018.
- [136] S. Karunaratne and H. Gacanin, “An overview of machine learning approaches in wireless mesh networks,” arXiv:1806.10523v1 [Online]. Available: <https://arxiv.org/abs/1806.10523v1>, Jun. 2018.
- [137] K. Yang, J. Ren, Y. Zhu, and W. Zhang, “Active learning for wireless IoT intrusion detection,” arXiv:1808.01412v1 [Online]. Available: <https://arxiv.org/abs/1808.01412v1>, Aug. 2018.
- [138] M. Sadeghi and E. G. Larsson, “Adversarial attacks on deep-learning based radio signal classification,” arXiv:1808.07713v1 [Online]. Available: <https://arxiv.org/abs/1808.07713v1>, Aug. 2018.
- [139] R. Zhu, T. Shu, and H. Fu, “Statistical inference attack against PHY-layer key extraction and countermeasures,” arXiv:1810.00200v1 [Online]. Available: <https://arxiv.org/abs/1810.00200v1>, Sep. 2018.

[140] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, “Applications of deep reinforcement learning in communications and networking: A survey,” arXiv:1810.07862v1 [Online]. Available: <https://arxiv.org/abs/1810.07862v1>, Oct. 2018.

[141] J. Roux, E. Alata, G. Auriol, M. Kaaniche, V. Nicomette, and R. Cayre, “RadIoT: Radio communications intrusion detection for IoT - A protocol independent approach,” arXiv:1811.03934v1 [Online]. Available: <https://arxiv.org/abs/1811.03934v1>, Nov. 2018.

[142] A. Y. Abyaneh, A. H. G. Foumani, and V. Pourahmadi, “Deep neural networks meet CSI-based authentication,” arXiv:1812.04715v1 [Online]. Available: <https://arxiv.org/abs/1812.04715v1>, Nov. 2018.

[143] U. Challita, A. Ferdowsi, M. Chen, and W. Saad, “Machine learning for wireless connectivity and security of cellular-connected UAVs,” arXiv:1804.05348v3 [Online]. Available: <https://arxiv.org/abs/1804.05348v3>, Nov. 2018.

[144] Y. Shi, T. Erpek, Y. E. Sagduyu, and J. H. Li, “Spectrum data poisoning with adversarial deep learning,” arXiv:1901.09247v1 [Online]. Available: <https://arxiv.org/abs/1901.09247v1>, Jan. 2019.

[145] M. Sadeghi and E. G. Larsson, “Physical adversarial attacks against end-to-end autoencoder communication systems,” arXiv:1902.08391v1 [Online]. Available: <https://arxiv.org/abs/1902.08391v1>, Feb. 2019.

[146] M. Z. Hameed, A. Gyorgy, and D. Gunduz, “Communication without interception: Defense against deep-learning-based modulation detection,” arXiv:1902.10674v1 [Online]. Available: <https://arxiv.org/abs/1902.10674v1>, Feb. 2019.

[147] B. Flowers, R. M. Buehrer, and W. C. Headley, “Evaluating adversarial evasion attacks in the context of wireless communications,” arXiv:1903.01563v1 [Online]. Available: <https://arxiv.org/abs/1903.01563v1>, Mar. 2019.

[148] Y. E. Sagduyu, Y. Shi, and T. Erpek, “IoT network security from the perspective of adversarial deep learning,” arXiv:1906.00076v1 [Online]. Available: <https://arxiv.org/abs/1906.00076v1>, Jun. 2019.

[149] T. M. Hoang, T. Q. Duong, and S. Lambotharan, “Secure wireless communication using support vector machines,” in *2019 IEEE Conference on Communications and Network Security (CNS)*, Jun. 2019, pp. 1–5.

[150] H. Fang, X. Wang, and S. Tomasin, “Machine learning for intelligent authentication in 5G-and-Beyond wireless networks,” arXiv:1907.00429v2 [Online]. Available: <https://arxiv.org/abs/1907.00429v2>, Jul. 2019.

[151] L. Zhang, F. Restuccia, T. Melodia, and S. M. Pudlewski, “Taming cross-layer attacks in wireless networks: A Bayesian learning approach,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1688–1702, Jul. 2019.

[152] T. C. Clancy and N. Goergen, “Security in cognitive radio networks: Threats and mitigation,” in *Proc. International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, May 2008, pp. 1–8.

[153] Y. Zhang, G. Xu, and X. Geng, “Security threats in cognitive radio networks,” in *Proc. IEEE International Conference on High Performance Computing and Communications*, Sep. 2008, pp. 1036–1041.

[154] J. Esch, “An introduction to the paper by Attar, Tang, Vasilakos, Yu, and Leung – A survey of security challenges in cognitive radio networks: Solutions and future research directions,” *Proc. IEEE*, vol. 100, no. 12, pp. 3170–3171, Dec. 2012.

[155] A. Attar, H. Tang, A. V. Vasilakos, F. R. Yu, and V. C. M. Leung, “A survey of security challenges in cognitive radio networks: Solutions and future research directions,” *Proc. IEEE*, vol. 100, no. 12, pp. 3172–3186, Dec. 2012.

[156] S. Holcomb and D. B. Rawat, “Recent security issues on cognitive radio networks: A survey,” in *Proc. IEEE SoutheastCon 2016*, Mar. 2016, pp. 1–6.

[157] Federal Communications Commission, “Notice for Proposed Rulemaking (NPRM 03- 22): Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies,” ET Docket, No. 03-108, Dec. 2003.

[158] R. Chen and J. Park, “Ensuring trustworthy spectrum sensing in cognitive radio networks,” in *Proc. IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, Sep. 2006, pp. 110–119.

[159] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” arXiv:1312.6199v4v2 [Online]. Available: <https://arxiv.org/abs/1312.6199v4>, Feb. 2014.

[160] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” arXiv:1412.6572v3 [Online]. Available: <https://arxiv.org/abs/1412.6572v3>, Mar. 2015.

[161] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” arXiv:1511.07528v1 [Online]. Available: <https://arxiv.org/abs/1511.07528v1>, Nov. 2015.

[162] A. Kurakin, I. Goodfellow, , and S. Bengio, “Adversarial examples in the physical world,” arXiv:1607.02533v4 [Online]. Available: <https://arxiv.org/abs/1607.02533v4>, Feb. 2017.

[163] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” arXiv:1608.04644v2 [Online]. Available: <https://arxiv.org/abs/1608.04644v2>, Mar. 2017.

[164] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, “EAD: Elastic-Net attacks to deep neural networks via adversarial examples,” arXiv:1709.04114v3 [Online]. Available: <https://arxiv.org/abs/1709.04114v3>, Feb. 2018.

[165] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” arXiv:1706.06083v4 [Online]. Available: <https://arxiv.org/abs/1706.06083v4>, Sep. 2019.

[166] S. Qiu, Q. Liu, S. Zhou, and C. Wu, “Review of artificial intelligence adversarial attack and defense technologies,” *Applied Science*, vol. 9, pp. 1–29, Mar. 2019.

[167] Y. Li, J. Bradshaw, and Y. Sharma, “Are generative classifiers more robust to adversarial attacks?” arXiv:1802.06552v2 [Online]. Available: <https://arxiv.org/abs/1802.06552v2>, May 2019.

[168] P. Samangouei, M. Kabkab, and R. Chellappa, “Defense-GAN: Protecting classifiers against adversarial attacks using generative models,” arXiv:1805.06605v2 [Online]. Available: <https://arxiv.org/abs/1805.06605v2>, May 2018.

[169] G. K. Santhanam and P. Grnarova, “Defending against adversarial attacks by leveraging an entire GAN,” arXiv:1805.10652v1 [Online]. Available: <https://arxiv.org/abs/1805.10652v1>, May 2018.

[170] G. Liu, I. Khalil, and A. Khreichah, “GanDef: A GAN based adversarial training defense for neural network classifier,” arXiv:1903.02585v1 [Online]. Available: <https://arxiv.org/abs/1903.02585v1>, Mar. 2019.

[171] Z. Luo, S. Zhao, Z. Lu, J. Xu, and Y. Sagduyu, “When attackers meet AI: Learning-empowered attacks in cooperative spectrum sensing,” arXiv:1905.01430v1 [Online]. Available: <https://arxiv.org/abs/1905.01430v1>, May 2019.

[172] S. Btahtcharjee, S. Sengupta, and M. Chatterjee, “Vulnerabilities in cognitive radio networks: A survey,” *Elsevier Computer Communications*, vol. 36, pp. 1387–1398, 2013.

[173] S. Srinivasan, K. B. Shivakumar, and M. Mohammad, “Semi-supervised machine learning for primary user emulation attack detection and prevention through core-based analytics for cognitive radio networks,” *International Journal of Distributed Sensor Networks*, vol. 15, no. 9, pp. 1–12, Sep. 2019.

[174] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurthb, “f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks,” *Elsevier Medical Image Analysis Journal*, vol. 54, pp. 30–44, May 2019.

[175] K. Davaslioglu, T. Erpek, and Y. E. Sagduyu, “End-to-end autoencoder communications with optimized interference suppression,” 2021.

[176] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” arXiv:1412.6980v9 [Online]. Available: <https://arxiv.org/abs/1412.6980v9>, Jan. 2017.

[177] I. Salian, “NVIDIA research achieves AI training breakthrough using limited datasets,” [Online]. Available: <https://blogs.nvidia.com/blog/2020/12/07/neurips-research-limited-data-gan/>, Dec. 2006.

[178] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” arXiv:2006.06676v2 [Online]. Available: <https://arxiv.org/abs/2006.06676v2>, Oct. 2020.

[179] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation generative adversarial networks,” arXiv:1711.04340v3 [Online]. Available: <https://arxiv.org/abs/1711.04340v3>, Mar. 2018.

[180] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, 2019.

- [181] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, “Differentiable augmentation for data-efficient GAN training,” arXiv:2006.10738v4 [Online]. Available: <https://arxiv.org/abs/2006.10738v4>, Dec. 2020.
- [182] E. Balevi and J. G. Andrews, “Wideband channel estimation with a generative adversarial network,” arXiv:2012.12063v1 [Online]. Available: <https://arxiv.org/abs/2012.12063v1>, Oct. 2019.
- [183] Q. Zhang, A. Ferdowsi, and W. Saad, “Distributed generative adversarial networks for mmWave channel modeling in wireless UAV networks,” arXiv:2102.10836v1 [Online]. Available: <https://arxiv.org/abs/2102.10836v1>, Feb. 2021.
- [184] Q. Zhang, A. Ferdowsi, W. Saad, and M. Bennis, “Distributed Generative Adversarial Networks (GANs) for data-driven millimeter wave communications in UAV networks,” arXiv:2102.01751v2 [Online]. Available: <https://arxiv.org/abs/2102.01751v2>, Jul. 2021.
- [185] X. Li, A. Alkhateeb, and C. Tepedelenlioglu, “Generative adversarial estimation of channel covariance in vehicular millimeter wave systems,” arXiv:1808.02208v1 [Online]. Available: <https://arxiv.org/abs/1808.02208v1>, Aug. 2018.
- [186] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, “Federative learning in mobile edge networks: A comprehensive survey,” arXiv:1909.11875v2 [Online]. Available: <https://arxiv.org/abs/1909.11875v2>, Feb. 2020.
- [187] B. Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep models under the GAN: Information leakage from collaborative deep learning,” in *Proc. 2017 ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2017, pp. 603–618.
- [188] C.-W. Ching, T.-C. Lin, K.-H. Chang, C.-C. Yao, and J.-J. Kuo, “Model partition defense against gan attacks on collaborative learning via mobile edge computing,” in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.