GAN-based Evasion Attack in Filtered Multicarrier Waveforms Systems

Kawtar Zerhouni, Gurjot Singh Gaba, *Member, IEEE*, Mustapha Hedabou,
Taras Maksymyuk, *Member, IEEE*, Andrei Gurtov, *Senior Member, IEEE*, and El Mehdi Amhoud, *Member, IEEE*

***Abstract*—Generative adversarial networks (GAN), a class of deep learning (DL) models, have emerged as a cyber threat to wireless communication systems. A potential cyber-attacker can use GAN to mislead the convolutional neural network (CNN) trained receiver by transmitting fraudulent wireless signals that are statistically indistinguishable from true signals. GANs have previously been used for digitally modulated single-carrier waveforms; however, in this article, we investigate the possibility of using an unsupervised GAN to model filtered multi-carrier waveforms, like orthogonal frequency-division multiplexing (OFDM), filtered orthogonal FDM (F-OFDM), generalized FDM (GFDM), filter bank multi-carrier (FBMC), and universal filtered MC (UFMC). In this paper, we perform an evasion attack using GAN counterfeited filtered multi-carrier signals in an attempt to deceive the target receiver. Our findings demonstrate a 99.7% likelihood of a receiver misclassifying GAN-based fabricated signals as authentic ones, necessitating further research into the timely development of preventive measures.**

***Index Terms*—Conditional generative adversarial networks, deep convolutional neural networks, evasion attack, filtered multicarrier waveforms.**

I. Introduction

Machine learning (ML) finds applications in diverse fields, including wireless communication. However, the majority of existing systems operate under the assumption that data is sourced from regular users and is independently generated from the same distribution [1]. While some algorithms are capable of handling small dense noises and large sparse outliers, only a limited number of them tackle adversarial noises that are deliberately crafted by individuals possessing knowledge of the machine learning system and the underlying data. These adversaries purposefully introduce meticulously crafted noises or directly manipulate the dataset to undermine or deceive the learning system, causing it to make inaccurate decisions which may present a substantial threat, especially in security and safety-critical domains [1]. The adversarial attacks can be categorized into three types based on the adversary's objective: evasion, poisoning, and model stealing [1]–[3]. Evasion attacks are a method of manipulating input test data to deceive ML models and cause them to make incorrect decisions [4], [5]. In the realm of wireless communications, these attacks have been applied to various scenarios, such as fooling classifiers used for spectrum sensing [34], modulation recognition [6], autoencoder-based end-to-end communication systems [7], channel state information (CSI) feedback for massive MIMO [8], channel estimation [9], [10] and initial access in directional communications [11]. Compared to traditional jamming attacks that focus on causing interference during data transmission [12], evasion attacks are often more discreet and energy-efficient, as they only require transmitting low-power signals for a short duration to confuse the ML algorithms in their decision-making process [2].

Several deep generative methods are accessible for generating such perturbations among which generative adversarial networks (GANs) have received tremendous attention due to their capacity to automatically discover and learn patterns in input data then generate realistic examples across a range of problem domains. Ever since they were introduced by Ian J. Goodfellow *et al.* in their seminal work [13] in 2014, their applications have increased rapidly. To date, most work on GANs in the field of wireless communications has focused on single carrier signals, for applications including- but not limited to- data augmentation, wireless channel modeling [14]–[17], physical layer design for communication systems [18], adversarial attacks and their detection [19], [20], as well as anomaly detection [21], [22]. For instance, in [19], a deep GAN was proposed to forge wireless signals as if they were conceived by legitimate users. More precisely, the authors used the I/Q components of quadrature phase shift keying (QPSK) signals propagating through Rayleigh channels to train a GAN composed of dense layers in both the generator and discriminator.

To evaluate the performance of their approach, the authors rely on a deep neural network (DNN) classifier designed to distinguish between signals from an intended transmitter, and unintended one [23]. The GAN based attack success probability is much higher than that of an attack using random or replayed signals [24]. The success probability of this spoofing attack rises with additional antennas at the transmitters (both intended and adversary) while it reduces with further antennas at the receivers [25]. In [20], a GAN based defense mechanism against primary user emulation attack (PUEA) was proposed by leveraging the discriminator of the trained GAN. The authors considered two models; with no prior information about primary users (PUs), called dumb attacker, and with some information about the PUs signal's features,

K. Zerhouni, M. Hedabou, and E. M. Amhoud are with the School of Computer Science, Mohammed VI Polytechnic University, Ben Guerir, Morocco, e-mail: {kawtar.zerhouni, mustapha.hedabou, elmehdi.amhoud}@um6p.ma.

T. Maksymyuk is with the Department of Telecommunications, Lviv Polytechnic National University, Lviv, Ukraine, email: taras.a.maksymiuk@lpnu.ua.

G.S. Gaba and A. Gurtov are with the School of Computer and Information Science, Linköping University, Sweden, email: {gurjot.singh, andrei.gurtov}@liu.se.

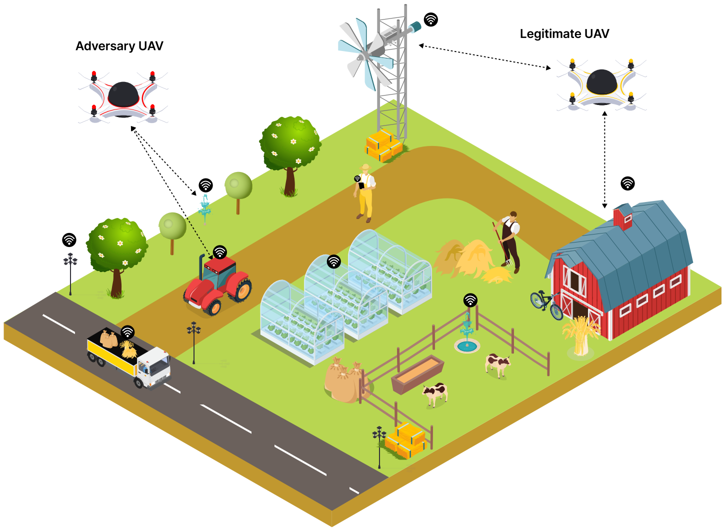


Fig. 1. Aerial data collection by UAVs from IoT devices in a smart farming scenario.

called smart attacker. The previous models were designed so that over time the discriminators eventually outsmart the generators, then use these pre-trained discriminators to distinguish between legitimate and fraudulent PUs.

The GAN model is based on dense layers, and trained on raw I/Q over-the-air data, collected utilizing universal software radio peripheral (USRP) transmitters using QPSK signals each. In [21], a radio anomaly recognition algorithm built on GAN is presented. An anomaly is defined if the signal is produced by either defective equipment or dishonest jammers. The goal being to detect whether a signal is normal or anomalous, a modified convolutional GAN trained on 2D spectrogram representations of an orthogonal frequency division multiplexing (OFDM) received signal is adopted. More exactly, spectrogram images representing normal signals are collected from the training examples, and are used to tune the proposed GAN model. Afterwards, based on the reconstruction error and the discriminator loss, an anomaly score is defined as the statistic test to recognize and isolate the anomalies. The work in [22] presents also an anomaly detection approach by comparing a conditional and an auxiliary cGAN (ACGAN) models, where the loss of the trained discriminator is used to compute an anomaly measure. The models are trained on a generalized state vector, composed of the signal's amplitude drawn from a Stockwell transform (ST) for mmWave OFDM signals.

The task of the GAN based data augmentation in communication systems has also attracted widespread attention, to handle labeled data insufficiency and improve the classification accuracy for automatic modulation classification (AMC) applications [26]. Indeed, in [27] the authors use an ACGAN model [28] to generate synthetic digital modulations, namely BPSK, QPSK, 16QAM, 32QAM, 64QAM, OQPSK, 4ASK, 8PSK, and use them along with real signals to train a convolutional neural network (CNN) classifier, more precisely, AlexNet. ACGAN adds a conditional information (a class label for example) to the input of the generator in an effort to control the data generation process and stabilize the training. The digital modulations are first converted to constellation diagrams, then into contour Stella images (a richer color feature transformation), before being used as training sets for both the ACGAN and AlexNet. The data augmentation approach enhances the performance of the classifier especially when the modulations are relatively identical and the signal-to-noise-ratio (SNR) is low, or the number of examples is insufficient for training.

A similar approach is presented in [29], where the authors train a conditional GAN (cGAN) to generate synthetic digital modulations, using the RML2016.10a dataset [30]. The adopted model uses fully dense layers at both the generator and the discriminator, and is trained directly on the I/Q data. Once the GAN training is performed, the synthetic I/Q are combined with the real dataset in an effort to improve the performance of the state-of-the-art digital modulation CNN classifier proposed in [31]- the synthesized data are more advantageous to enhance the CNN accuracy for low SNR values. The authors of [32] also propose a cGAN model, where both the modulation order and the SNR of the training signals are used as input instructions, so that the same generator of the trained GAN can forge fake data with a specific modulation order and SNR. This work also uses the RML2016.10a dataset, instead of manipulating the raw I/Q data directly, these components are first converted into a 64 × 64 constellation diagrams. Once the GAN's training is finished, the forged data is used to improve the performance of a CNN model comprising of four convolutional layers in addition to one fully-connected layer. The proposed technique boosts the performance of the classification accuracy specially for low SNR and reduces training time in contrast to the ACGAN.

The compelling properties of OFDM have accelerated its infiltration in wireless communication standards, which have also sparked an in depth investigation of multicarrier waveforms [33]. Over the course of the last decade, a number of alternative transmission schemes have been proposed to improve the deficiencies of OFDM. Among the various waveform contenders, the most promising ones are filtered versions of OFDM. These filtered versions utilize different pulse shapes to address the drawbacks associated with conventional OFDM. Given the emergence of filtered multicarrier waveforms, it becomes crucial to reassess the approach of GAN based waveform generation, to take into consideration the diversity and complexity of the advanced multicarrier mechanisms. In fact, filtered multicarrier systems are more complex than single-carrier systems. They involve multiple carriers with specific filtering requirements and inter-carrier relationships. Learning to generate such complex waveforms using ML techniques requires more sophisticated models and increased computational resources. Furthermore, the model needs to generalize well to the variations of filtered multicarrier systems. Addressing these challenges requires careful consideration of model architectures, loss functions, training strategies, and the incorporation of domain knowledge and signal processing expertise into the ML framework. Therefore, the primary objective of this study is to explore the potential of GAN-based modeling for baseband filtered multicarrier waveforms directly from the raw I/Q data. The focus of our investigation centers around the following categories:

- *Per-subcarrier* filtered multicarriers family: filter bank multicarrier (FBMC) and generalized frequency division multiplexing (GFDM) [34]–[36].
- *Per-subband* filtered multicarriers family: universal filtered multicarrier (UFMC) and filtered orthogonal frequency division multiplexing (F-OFDM) [37]–[40].

Our intention is to create an evasion attack [3], where an adversary attempts to fool a machine learning algorithm into making a wrong decision [4]. Fig. 1 illustrates a vulnerable network where an unmanned aerial vehicle (UAV) communicates with ground devices [41]–[44]. In this network, an adversarial UAV seeks to spawn signals which are statistically analogous to those collected from the legitimate one; therefore the GAN training is performed by the adversarial node. Conventional OFDM is the principal waveform candidate for drone-to-ground communication [45]. Nonetheless, its innovative filtered contenders present a better trade-off between time-frequency localization and critical waveform density, hence allow the co-existence of several drones with enhanced throughput [45]. In order to identify the intended UAV transmitters relying on the investigated schemes [46], [47], a CNN classifier is built by collecting their various signals at each receiving device. The end goal of the adversarial UAV is to fool such a classifier with the synthetic waveforms spawned using the GAN generator. There are unique challenges for such approach, since each of these advanced waveforms, introduces distinct filtering techniques which interact differently with the propagation channel- all these embedded effects need to be learned by the GAN. The results reported in this research work prove the practicality of this proposal, and provide a foundation for further investigations.

The paper is structured in the following way: Section II briefly describes the waveforms investigated herein. Section III details the proposed GAN model for the considered filtered multicarrier signals. In Section IV, the training steps and simulation parameters are presented followed by discussions on findings. Finally, Section V concludes the paper and provides our future perspectives.

II. System model

In order to overcome the frequency selectivity of a wireless propagation channel on wideband signals, multicarriers waveforms have been widely adopted. In fact, a multicarrier system subdivides the total bandwidth into smaller subchannels, each one lower than the coherence bandwidth of the propagation channel. In this work, we consider the same signal model adopted in [33], [48], [49]. A multicarrier transmitted signal can then be given as follows:

$$x\left[n\right] = \sum_{r=0}^{N_{symbols}} \sum_{q=0}^{N_c-1} s_{q,r} g_{q,r}\left[n\right]. \quad (1)$$

Data symbols, $s_{q,r}$, are zero-mean independent and identically distributed (i.i.d) random variables, originating from a quadrature amplitude modulation (QAM) constellation, carried on the q-th subcarrier during the r-th period. $g_{q,r}[n]$ is the synthesis function which maps the data symbol to the signal dimension [33]. It is a *Gabor System* expressed as [33]:

$$g_{q,r}\left[n\right] = g_{tx}\left[n - rN\right] e^{j\frac{2\pi q n}{N_c}}. \quad (2)$$

where $g_{tx}[n]$ is the transmit prototype filter from which all other pulse shapes are inferred, through time shifting and frequency modulation [33]. The received counterpart of the signal in Eq. (1) is a sum of attenuated, Doppler shifted, and time stretched/compressed copies, which can be expressed by [48]:

$$r_{ch}\left[n\right] = x[n] * h[n] = \sum_{j=0}^{L_{ch}-1} h[\eta_j] e^{-j2\pi f_{d_j} n} x[n-\eta_j]. \quad (3)$$

L_{ch} stands for the number of multipaths, η_j is a discrete propagation delay, while $h[\eta_j]$ is the time varying channel gain associated to the l-th path. $f_{d_j} = f_c \frac{v_d}{c}$ denotes the Doppler shift, c is the speed of light while v_d is the relative speed between the transmitter and receiver. Considering a symbol time offset δ, a carrier frequency offset κ, and an additive white Gaussian noise (AWGN) $z[n]$ with variance σ_z^2, the final received signal can be expressed as:

$$y\left[n\right] = e^{j\frac{2\pi n \kappa}{N_c}} r_{ch}[n+\delta] + z[n]. \quad (4)$$

The multicarrier contenders of the legacy OFDM, rely on different synthesis functions in an attempt to circumvent the Balian-low theorem (BLT) limit. Indeed, this theorem proves the impossibility of designing a prototype $g_{tx}[n]$ which achieves *orthogonality*, is *localized* both in time and frequency, while attaining *critical density* [33], [50], [51]. OFDM is an orthogonal waveform which attains critical density [52], however, it is not well localized in frequency hence it suffers from high out of band (OOB) radiations. The oldest OFDM variant investigated in this work is FBMC [53]. This waveform introduces a real domain orthogonality, instead of a complex one, and relies on long prototype filters for each subcarrier, to offer very low OOB emissions. Another waveform that uses per-subcarrier filtering is GFDM [34].

In order to avoid the long filter tails of FBMC, GFDM introduces the concept of subsymbols, then adopts a circular filtering of the individual subcarriers. Both these schemes overcome the lack of spectral containment of OFDM at the expense of a higher receiver complexity. The subband filtered family offers a better trade-off between OOB suppression and receiver complexity. Inspired by the resource block based allocation in 4G systems, UFMC proposes filtering a group of subcarriers, also known

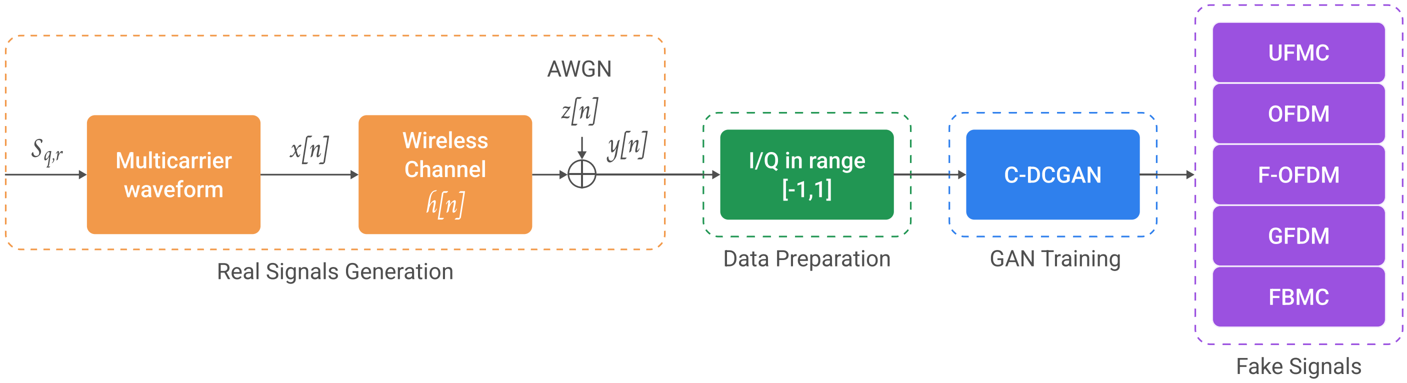


Fig. 2. Steps for fake multicarrier signals generation using a C-DCGAN.

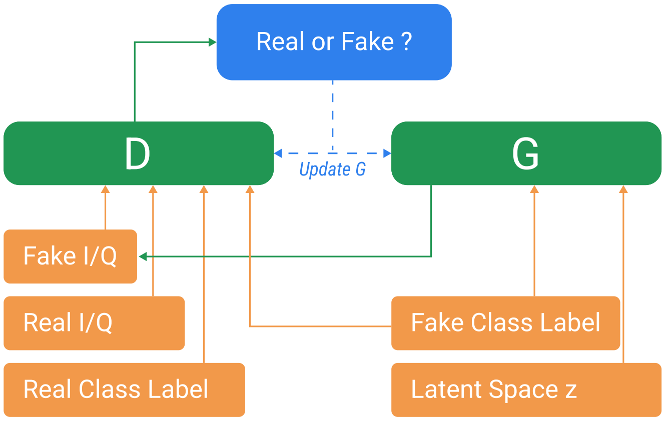


Fig. 3. Class label conditional GAN architecture for multicarrier waveforms.

as subbands, instead of subcarriers; it exhibits moderate OOB compared to FBMC but circumvents its drawback [37]. Finally, F-OFDM uses longer filters compared to UFMC, hence presents a better spectral containment, however it suffers from inter symbol interference (ISI) [40]. A thorough investigation as well as a comparison between the studied waveforms, is presented in [33], [54]–[56] and the references therein.

In order to spawn fake examples, the GAN attempts to learn the statistical characteristics of the real ones. The different filter designs utilized in each waveform result in rich statistical features, which can be learned by the GAN then used to generate fake signals. Fig. 2 depicts the steps of this generation process, where the investigated waveforms are first simulated, then filtered by the channel's transfer function, before being fed to a conditional deep convolutional GAN (C-DCGAN). The latter attempts to learn the distinctive features of each waveform, then spawns fake signals statistically similar to the studied ones. The following section provides details of the C-DCGAN model.

III. GAN ARCHITECTURE FOR MULTICARRIER SIGNALS

GANs are a novel deep-learning based approach to generative modeling, proposed by Ian J. Goodfellow *et al.* in [13]. The goal is to learn regularities in a dataset, then use these patterns to induce different examples that could allegedly be extracted from the dataset at hand. This proposal constructs the problem as a supervised learning task, by using two adversary networks: A generator G, and a discriminator D, each being trained to outperform the other [57]. The generator is a neural network trained to learn the probability distribution of the target data p_u, while the discriminator, also a neural network, attempts to differentiate between examples spawned from G's distribution, p_g, and those taken from the real dataset. The input to the generator is a fixed length vector $\mathbf{z}$, also known as a *latent space*, from a Gaussian distribution. G learns to map $\mathbf{z}$ to the dataset distribution. The discriminator is a classification model that predicts a binary class label of fake or real. The two models are trained jointly in a zero-sum game. When G fools the discriminator, no changes are applied to its model weights, while D parameters are updated. Alternately, when D distinguishes real and fake examples, it is rewarded, while G is penalized with substantial updates, as depicted in Fig. 3. Both the generator and the discriminator have the same loss function presented in Eq. (5), but the first attempts to minimize it, whereas the second seeks to maximize it [13]:

$$\mathfrak{L} = E_{u\sim p_u}[\log(D(u))] + E_{z\sim p_z}[\log(1-D(G(z)))]. \quad (5)$$

$D(u)$ is the discriminator's approximation of the probability that authentic data occurrence u is real, $G(z)$ is the generator's output when presented with the noise z, and $D(G(z))$ denotes the discriminator's estimate of the probability that the forged occurrence is real. Since the generator doesn't see the real instances, it cannot directly affect $\log(D(u))$, hence minimizing the loss function is comparable to minimizing $\log(1-D(G(z)))$. In practice, this function saturates for the generator, to circumvent that challenge, the generator aims at maximizing $\log(D(G(z)))$ [57].

Traditional GANs can spawn original credible samples for a given dataset, nonetheless, they do not offer a way to supervise the types of the fake instances, hence it is a common practice to rely on human perception in computer vision applications to assess the fidelity of the forged data. Waveforms in general, and filtered multicarrier ones in particular, are not directly human interpretable, hence the evaluation of a traditional GAN is not straightforward. We inspect a class conditional variant of the model, using a cGAN, to allow a targeted multicarrier waveforms generation. cGAN is a type of GAN where a supplementary information is associated to the input data, class labels for example [58]. In order to achieve this goal, it is a common practice to introduce an embedding layer combined to a fully connected one, to scale the embedding to the dimension of the input data, then integrate it in the model as an added feature map [57].

In our work, the dataset at hand is a collection of simulated 2 × 128 I/Q signals, with different waveforms. The GAN architecture relies on convolutional and convolutional-transpose layers in the discriminator and generator, correspondingly. The discriminator must take an I/Q sample and return a binary classification, whether the instance is fake or real. Our discriminator model has four 2D convolutional layers, each with a kernel size of 4 and a stride of 2. No pooling layers are used, while the output layer is of one node and a Sigmoid activation function. The discriminator is trained to minimize the binary cross-entropy loss function. We adopt some of the GAN's training best practices. We employ Leaky rectified linear unit (ReLU) instead of ReLU to avoid sparse gradients; we add a dropout layer, and set its rate to 0.3, to improve the model's ability to generalize. Finally, we exploit the Adam variant of the stochastic gradient descent, setting the learning rate to 0.0002 and the momentum to 0.5 [59], [60].

The generator must transform a point from the latent space to a plausible 2×128 I/Q vector. To this end, we adopt the approach of using a dense layer at first followed by as many upsampling layers as needed. More specifically, the first hidden layer of the generator, produces 256 multiple parallel versions of a smaller I/Q vector, 1 × 8, with different learned features each. They are reshaped and passed to a transpose convolution layer (also called deconvolution), which is a combination of an upsampling layer and a convolution one, with a kernel size of (4,4), and a stride of (2,2). This layer will double the size of the I/Q vector from 1×8 to 2×16, followed by a similar layer with a kernel size of (2,4), and a stride of (1,2), scaling the generated vectors up to 2×32, and so on until we reach the desired I/Q vector size. Again, we will apply the Leaky ReLU with a default slope of 0.2. Finally, the output layer of the generator model is a 2D convolution layer with one filter. To stabilize the GAN training, we adopt a Tanh activation function in attempt to guarantee that the output values are in the range of [-1,1] [57]. For both models, the class label is given using an embedding layer of the size of 50. Each of the 5 waveform classes will be mapped to a distinct 50-element vector interpretation.

IV. Methodology, Results and Discussions

We design different simulations in order to build an intuition on how the GAN performs with the filtered multicarrier signals. We construct a comprehensive dataset composed of signals with $N_{symbols} = 10$, where the SNR takes values in the range $[-8 : 20]$ dB with a step of 2 dB. We set the number of examples for each SNR to 5000 per class. To create signals with similar multicarrier symbol length, we fix the number of subcarriers to $N_c = 16$ for all the investigated waveforms excluding GFDM, for which $N_c = 8$. We employ the prototype filters widely adopted in the literature and previously investigated for waveforms classification in [45], [49]. For GFDM, we use the RRC filter where the roll-off factor is $\beta = 0.35$. For FBMC, we apply the Phydyas filter using an overlapping factor of $K = 2$. The Chebyshev filter is adopted for UFMC, finally, for F-OFDM we utilize the truncated sinc [34], [35], [37], [40]. The data symbols are drawn from a QPSK constellation. Table I outlines the simulation parameters in more details. We set the cyclic prefix length $\frac{N_c}{4}$, a value greater than the maximum channel delay to avoid ISI. The multipath channel effects are introduced using the profile of the long term evolution (LTE) channel model, specifically, we opt for the extended typical urban model (ETU) model [61]. The signals are conceived utilizing Matlab, whereas the machine learning models are built and trained in Python, relying on Keras library with TensorFlow backend, over the GPU of Google Colab Pro.

A. Evaluation methods

GANs evaluation is still a problem of research in the computer vision community [62]. In order to judge the performance of a GAN architecture, several quantitative and qualitative metrics have been proposed to inspect not only the quality of the forged data, but also its diversity. Since our goal is to mimic the signals sent from a legitimate user, we will consider that the GAN has converged if it can fool the CNN classifier used at the receivers. Taking advantage of the conditional GAN, we define the following metrics:

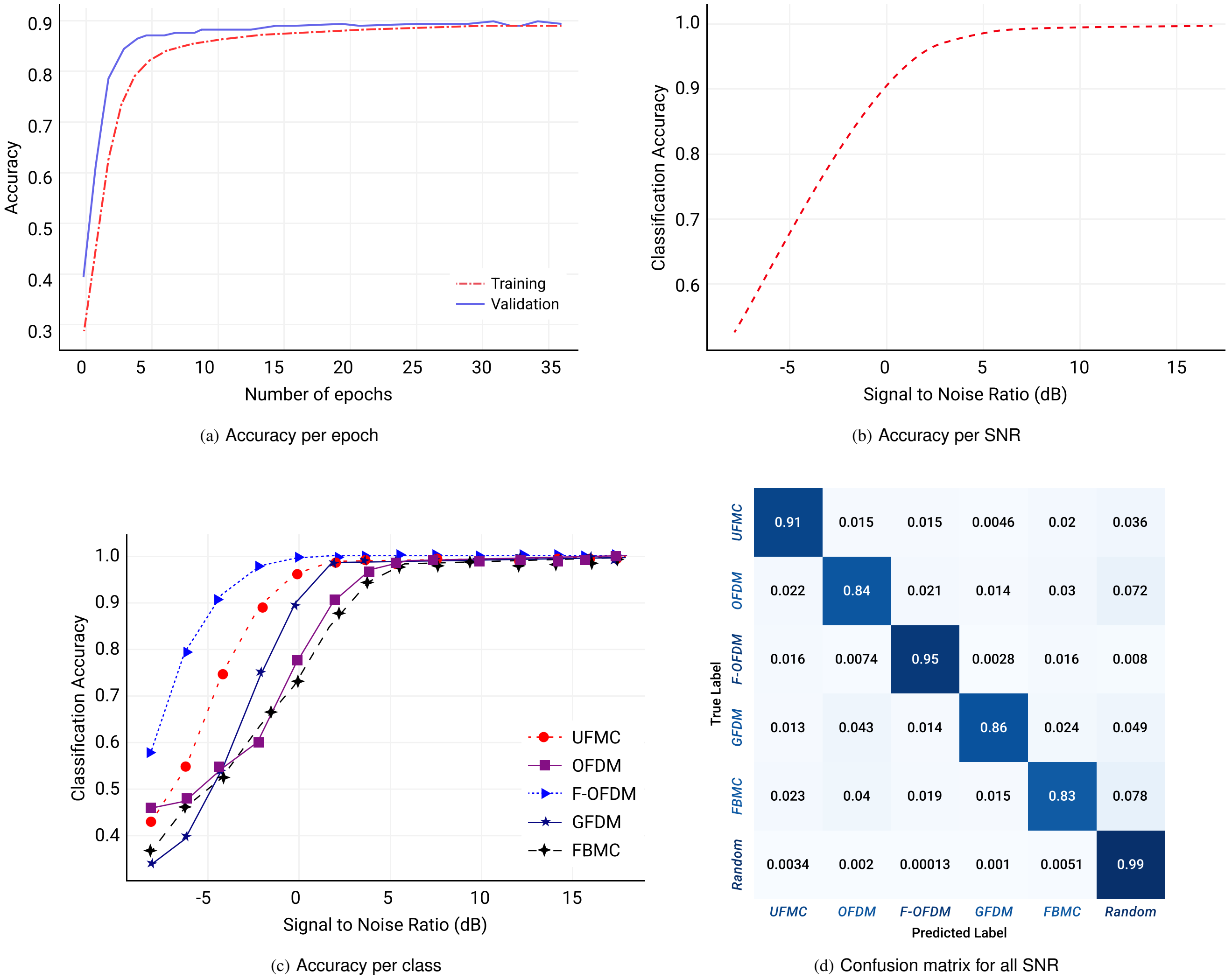
1) The probability of global correct classification:

$$P_{gcc} = \frac{N_{cc}}{N_f \times N_w}, \tag{6}$$

where N_w is the total number of waveforms used to train the GAN. N_f is the number of fake signals per waveform used to evaluate the GAN. N_{cc} is the total number of correct classification across all waveforms. It is a count incremented each time the labels predicted using the CNN classifier match the conditional labels used to generate the fake signals by the generator of the GAN.

2) The per-class correct classification probability:

$$P_{ccc} = \frac{\sum_{i=1}^{N_f} p_{c_i}}{N_f}, \tag{7}$$



(a) Accuracy per epoch

(b) Accuracy per SNR

(c) Accuracy per class

(d) Confusion matrix for all SNR

Fig. 4. Performance of the CNN classifier: Accuracy and Confusion matrix.

which we define as the average of the per class probabilities reported by the CNN, denoted p_c in Eq. (7), over the N_f synthetic signals, for each class separately.

Furthermore, we borrow two largely employed measures for evaluating fabricated images: The inception score (IS) and Frechet inception distance (FID) score [62]–[64]. Both our proposed scores and the borrowed ones are computed utilizing a previously trained classification model. IS score is calculated using the top-performing image classification model Inception v3 model [65]. This score attempts to acquire two properties of a compilation of forged images: Image diversity and image quality. To calculate the IS score on a selection of fabricated images, first, the conditional probability for each image $p(y|x)$, is calculated employing the Inception v3 model, afterwards the marginal probability is computed as the mean of the conditional probabilities for the synthetic images in the group $p(y)$. The Kullback-Leibler (KL) divergence (kld) is measured for the individual images as follows:

$$kld = p(y|x) \times (\log(p(y|x)) - \log(p(y))). \tag{8}$$

To compute the final score, first a sum of KL divergence over all images is performed, then an average over all classes is calculated, finally the exponent of the result is determined. The minimum value of the IS is 1 and the maximum value is the number of classes of the classification model. We use the same idea, replace the Inception v3 with a CNN model for filtered multicarrier waveforms classification, and fake images with fake multicarrier signals. FID on the other hand, assesses the quality of a compilation of fabricated images compared to a collection of real ones, based also on the Inception v3 model. More precisely, the score evaluates real vs forged data based on their statistical features, comparing both image quality, through conditional class predictions for each fabricated instance, and image diversity through the marginal probability of the predicted classes [57]. Again, we explore the same idea, replace the Inception v3 with

TABLE I
SIMULATION PARAMETERS [49]

Parameter	Symbol	Value
Overall parameters		
QAM	-	QPSK
IDFT size	N_c	16
Subcarrier spacing	Δf	15 kHz
Sampling time	T_s	$1/(\Delta f * N_c)$
CP length	N_{cp}	$N_c/4$
GFDM		
Filter name	-	RRC
Roll off	β	0.35
Subsymbols number	M	2
Subcarriers number	N_c	8
UFMC		
Filter length	L_u	N_{cp}+1
Filter name	-	Dolph-Chebyshev
Filter attenuation(dB)	-	40
Subband size	Q	8
Subbands number	S	2
FBMC		
Filter name	-	Phydyas
Overlapping factor	K	2
F-OFDM		
Filter name	-	Truncated Sinc
Filter length	L_{fo}	$N_c/2$

a CNN model for filtered multicarrier waveforms classification, remove the output layer, then use the features extracted by the last dense layer. These activations are computed for a compilation of real and forged signals, then summed up as a multivariate Gaussian by calculating the mean and covariance of the extracted features. The real signals are those used for the GAN training, while the fake signals are those spawned from the generator. The distance between the two distributions is then measured using the Frechet distance as proposed in [66]:

$$d = ||\mu_1 - \mu_2||^2 + Tr(C_1 + C_2 - 2 \times \sqrt{C_1 \times C_2}). \quad (9)$$

$Tr(.)$ indicates the trace linear algebra operation. μ_1 and μ_2 refer to the feature-wise mean of the actual and fabricated signals. C_1 and C_2 are the covariance matrices for the actual and fabricated feature vectors. Smaller FID values indicate that the two compared groups are more alike. Python implementation details of the IS and FID scores can be found in Chapter 12 and Chapter 13 of [57] respectively, where the Inception v3 model was replaced by the CNN model presented afterwards.

We use a CNN model similar to the architecture proposed in [49] for filtered multicarrier waveforms classification. A 4-layer network is subdivided into two convolutional layers and two dense fully connected ones. The three first layers utilize ReLU activation function, whereas the last one adopts a SoftMax transfer function. While tuning the hyper-parameters of this network to our new database- which includes a 6^{th} class of random data in addition to the investigated waveforms- we employ 128 filters of length 1×3 in the first layer, reduce the second layer size to 64 filters of 1×3, and set the dense layer's length to 128 neurons. The output layer uses one node per class label for a total size of 6-class neurons.

Following the recommendations in [45], we add a dropout after each layer of the three first ones, the dropout rate is set to $dr = 0.3$, in addition an early stopping method is employed, both techniques are used to avoid over-fitting. We resort to the Adam optimization version of the traditional stochastic gradient descent [59]. In all our experiments, 80% of the data is used for training, and the remaining 20% of examples are utilized for validation and testing. Input training data are normalized to achieve a zero mean of the observed values, and a unity standard deviation. We train the CNN model using 2500 examples per class per SNR (from 1 to 2500), and achieve an overall classification accuracy across all SNR values of 0.895.

Fig. 4 depicts the CNN performance. Fig. 4a displays a good fit of the CNN learning curve, monitoring the accuracy on the training and validation data correspondingly. Fig. 4b shows the evolution of the CNN accuracy with increasing values of SNR. Fig. 4c represents the per class accuracy as a function of SNR. From the figure, we notice that F-OFDM classification outperforms the others, ahead of UFMC, GFDM then OFDM and FBMC. The application of filtering in the analyzed waveforms creates distinctive statistical attributes that the CNN model adeptly captures and utilizes to precisely classify these novel transmission schemes. Finally, Fig. 4d illustrates the confusion matrix for all SNR values, It is evident that there is a significant confusion between GFDM and OFDM, and vice versa. This outcome can be rationalized by the fact that OFDM can be viewed as a special case of GFDM, particularly when the number of subsymbols is set to 1.

B. GAN based physical layer attack generation

In this set of simulations, we train the GAN using the received waveforms- 2500 examples per waveform per SNR, for 3 SNR values, 20, 18, and 16 dB- for different cases in order to tune the GAN's hyper-parameters. We perform the GAN training for 500 epochs, save the generator model periodically (each 10 training epochs), then calculate the IS, FID, P_{gcc}, and P_{ccc} for each saved model using $N_f = 2500$ examples of fake signals per waveform. We reiterate each experiment 3 times, then disclose the best results.

We start by evaluating the impact of the batch size on the proposed C-DCGAN performance. In [67], different batch sizes were tested and evaluated, and the authors recommend using larger batch sizes, up to 2048, to achieve better quality synthetic images. Such result was not found to be true for our database, for batch sizes multiple of 128. Although larger batch size reduces the training time as shown in Table II, the highest IS score, highest P_{gcc}, as well as lowest FID score were achieved for the 130^{th} epoch of the 512 batch size. Based on the reported scores, we

TABLE II
IMPACT OF BATCH SIZE ON GAN'S TRAINING TIME AND PERFORMANCE.

Exp.	Batch	Training	Epoch	P_{gcc}	IS	FID	P_{ccc}				
	Size	(hh:mm)					UFMC	OFDM	F-OFDM	GFDM	FBMC
1	1280	01:26	310	0.71	2.95	160.21	0.571	0.019	0.955	0.748	0.876
2	896	01:34	480	0.73	3.12	134.86	0.999	0.032	0.792	0.627	0.928
3	512	02:12	130	0.78	4.06	103	0.939	0.362	0.964	0.528	0.991
4	256	02:37	280	0.67	3.12	165.68	0.927	0.53	0.977	0.014	0.763

TABLE III
IMPACT OF VARIETY AND SIZE OF TRAINING EXAMPLES ON GAN'S TRAINING TIME AND PERFORMANCE.

Exp.	SNR	Examples	Training	Epoch	P_{gcc}	IS	FID	P_{ccc}			
	(dB)	(indices)	(hh:mm)					UFMC	F-OFDM	GFDM	FBMC
5	16 to 20	1 to 2500	01:44	450	0.997	3.95	81.38	0.999	0.991	0.996	0.985
6	16 to 20	2500 to 5000	01:30	500	0.921	3.95	130.11	0.998	0.997	0.592	0.984
7	10 to 20	1 to 2500	03:04	270	0.887	3.68	116.80	0.955	0.970	0.540	0.791
8	16 to 20	1 to 5000	02:58	480	0.866	3.20	119.35	0.978	0.986	0.474	0.974

can also conjecture that there is no direct correlation between the batch size and GAN performance for the chosen architecture and database. Table II reports also the per-class correct classification probability P_{ccc} for the same experiments.

We notice that the highest IS scores correlate with high P_{ccc}. In fact, the best IS score, 4.06 corresponds to 3 classes with P_{ccc} higher than 0.93 each, UFMC, F-OFDM and FBMC, plus GFDM with a lower P_{ccc}. We noticed an interesting behavior for this particular GAN architecture, the generator is not able to maximize both GFDM and OFDM at the same time. It should be mentioned that we stop at 500 epochs because further training did not initially lead to improvement. We trained our GAN for 1000 epochs, with a batch size of 512, 3 times, where each trial took around 4 hours of training, and achieved the best results prior to 500 epochs. More precisely the P_{ccc} of GFDM and OFDM didn't improve, while that of the other waveforms hovers around 0.95 after the 300^{th} epoch. We attribute the improved behavior of our C-DCGAN for these filtered OFDM variants, to the richer statistical features they exhibit as a result of filtering at multiple levels (subband, per subcarrier, per subsymbol). The filtering employed in the studied waveforms gives rise to unique statistical characteristics, which the GAN model effectively captures and employs to accurately generate synthetic versions of these novel transmission schemes.

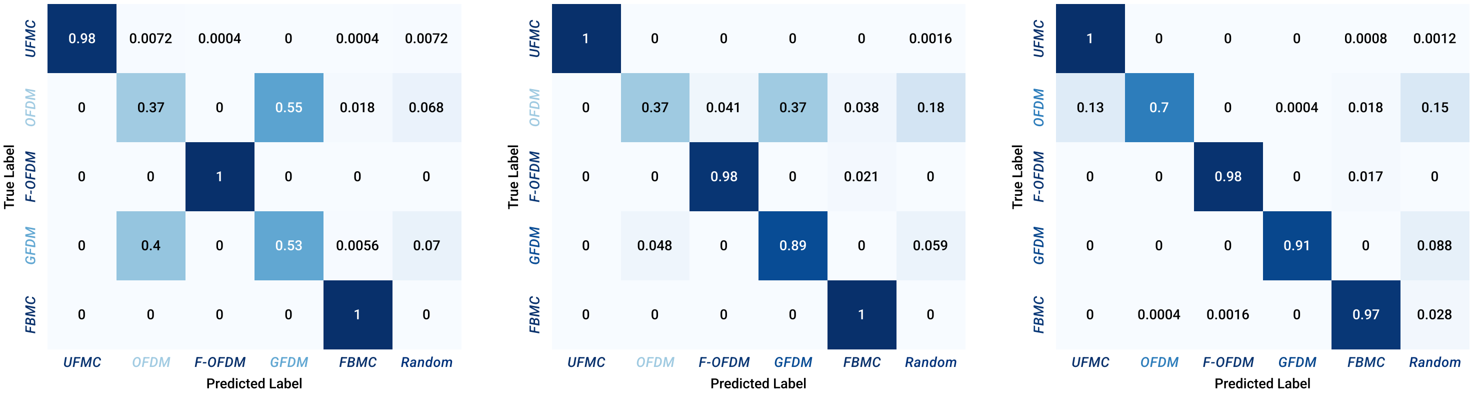
Since we are more interested in the filtered multicarrier families, we want to have a single generator model which can forge the four investigated waveforms. We run a second set of experiments, where we train the GAN omitting OFDM from the training vector. Table III reports the training time and scores for these different experiments. In Exp 5, we use the 4 investigated OFDM variants, and train the GAN with the same set of examples used to train the CNN (1 to 2500 per waveform per SNR), while in Exp 6 we use a different set of examples (2500 to 5000 per waveform per SNR). We were able to achieve a lower FID and higher P_{gcc} in both experiments compared to Exp 3, as detailed in Table III. The P_{ccc} of the targeted waveforms is higher than 0.98 for Exp 5, while it drops to 0.6 in Exp 6 for GFDM, which means that our model did not generalize well for this particular waveform across different datasets.

Furthermore, we examine the impact of the number and variety of examples on our C-DCGAN performance. We train the GAN with 5000 examples total per waveform, with two different configurations. In Exp 7, we use 2500 examples per waveform for 6 SNR values (10 up to 20 dB), while in Exp 8 we use 5000 examples per waveform for 3 SNR values (16 up to 20 dB). The examples used in Exp 7 have all been seen by the CNN, but are more varied in their noise level, while half of those in Exp 8 have never been used for the CNN training. Increasing the number of examples did not lead to improvement for this architecture. From Table. III, we can notice that the P_{gcc} dropped to 0.887 and 0.86 for Exp 7 and 8 respectively. The waveform that the GAN failed to maximize in both experiments is GFDM, while FBMC P_{ccc} decreased with a dataset of mixed SNR values. We can conclude that our model did not generalize well for GFDM. This waveform is the most complex among the investigated ones, indeed, in addition to the per subcarrier filtering, GFDM introduces also filtering per subsymbols.

Finally, we inspect the effect of the network depth on the capabilities of our C-DCGAN, for all waveforms. We increase either the number of filters, as in Exp 9 and 10, or we add a convolutional layer (at both G and D models) as in Exp 11 and 12. As reported in Table. IV, we were able to improve the global correct classification probability for Exp 9 and 10 compared to our best model of Exp 3, at the expense of a larger network and longer training time. The fake GFDM P_{ccc} improved from 0.528 in Exp 3, to 0.84 and higher for Exp 9 and 10, while that of OFDM increased to 0.59. From the scores of both Exp 11 and 12, we can deduce that adding a convolutional layer did not lead

TABLE IV
IMPACT OF NETWORK DEPTH ON GAN'S TRAINING TIME AND PERFORMANCE.

Exp.	Filters	Training	Epoch	P_{gcc}	IS	FID	P_{ccc}				
	per layer	(hh:mm)					UFMC	OFDM	F-OFDM	GFDM	FBMC
3	256/128/64/32	02:12	130	0.78	4.06	103	0.939	0.362	0.964	0.528	0.991
9	256/256/128/64	03:10	290	0.852	3.92	113.589	0.991	0.389	0.951	0.879	0.998
10	256/256/128/128	03:53	460	0.906	3.964	135.11	0.996	0.591	0.967	0.848	0.874
11	256/128/128/64/32	01:54	180	0.77	3.56	125.72	0.989	0.017	0.999	0.832	0.938
12	256/256/128/64/32	03:15	450	0.712	3.31	126.291	0.872	0.188	0.959	0.359	0.971



(a) Exp. 3, Epoch 130

(b) Exp. 9, Epoch 290

(c) Exp. 10, Epoch 460

Fig. 5. Confusion matrices of the CNN classifier with fake signals, for different generator models.

to any improvement for this specific dataset.

Lastly, Fig. 5 depicts confusion matrices drawn using the best generator models. From Fig. 5a, 5b and 5c, we can clearly see that the fake GFDM and OFDM signals are confused, but the GAN generates more discernable instances of these two waveforms with increased number of filters. In these figures, true labels are the conditional labels used in the generator, while the predicted ones are declared by the CNN. All our trained models can be used to create a successful evasion attack. Since the intention of the adversary transmitter is to emit waveforms which are statistically almost indistinguishable from the intended transmitter's ones, and fool the classifier at the target receiver, we can consider a random signal as a basic evasion attack. The success probability of such an attack is 4.56% reported by the CNN for a vector of random data. The classifier trained at intended receivers, can discriminate real signals from random ones, but cannot effectively identify fabricated signals generated by the GAN. The success probability of the GAN based evasion attack is much higher around 99.7% using the generator model of Exp 5 for instance as reported in Table V.

TABLE V
SUCCESS PROBABILITY OF EVASION ATTACKS.

Attack type	Success probability
Random signals	4.56%
GAN-based (Exp. 5)	99.7%

V. CONCLUSION AND FUTURE SCOPE

In this work, we introduced a wireless evasion attack, where an adversary transmitter generates forged signals that cannot be accurately differentiated from intended ones using a conditional GAN. We considered the advanced filtered multicarrier waveforms, namely UFMC, F-OFDM, GFDM and FBMC and the legacy OFDM. Since the goal of this paper is to fool a CNN classifier, trained at the intended receiver to discriminate between the investigated waveforms and random signals, we defined the global and per class correct classification probabilities, P_{gcc} and P_{ccc} respectively, then used them with the IS and FID scores to evaluate the forged signals. We showed that the classifier recognizes attacks based on random signals, while it misclassifies those generated by the GAN. We also noticed that the proposed model was not able to learn all the waveforms equiprobably, for instance the generator failed to maximize both GFDM and OFDM at the same time. In all our experiments, we trained the proposed GAN using the raw I/Q data, and spawned fake signals which were 99% misclassified as real ones, with correct labels. In a future work, we plan to develop other evaluation metrics specific to wireless signals, such as the quality of the power spectral density (PSD) and others more specific to each waveform such as the shape of the filters. Furthermore, the GAN based evasion attack presents a practicable menace against CNN based detection methods, in a future work we will investigate defense mechanisms to identify and circumvent these type of attacks.

REFERENCES

[1] F. Li, L. Lai, and S. Cui, *Machine Learning Algorithms: Adversarial Robustness in Signal Processing*. Springer Nature, 2022.

[2] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial machine learning in wireless communications using rf data: A review," *IEEE Communications Surveys & Tutorials*, 2022.

[3] K. W. McClintick, J. Harer, B. Flowers, W. C. Headley, and A. M. Wyglinski, "Countering physical eavesdropper evasion with adversarial training," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1820–1833, 2022.

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

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[6] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2018.

[7] ——, "Physical adversarial attacks against end-to-end autoencoder communication systems," *IEEE Communications Letters*, vol. 23, no. 5, pp. 847–850, 2019.

[8] Q. Liu, J. Guo, C.-K. Wen, and S. Jin, "Adversarial attack on dl-based massive mimo csi feedback," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 230–235, 2020.

[9] T. Hou, T. Wang, Z. Lu, Y. Liu, and Y. Sagduyu, "Undermining deep learning based channel estimation via adversarial wireless signal fabrication," in *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, 2022, pp. 63–68.

[10] T. Hou, S. Bi, T. Wang, Z. Lu, Y. Liu, S. Misra, and Y. Sagduyu, "Muster: Subverting user selection in mu-mimo networks," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 140–149.

[11] B. Kim, Y. Sagduyu, T. Erpek, and S. Ulukus, "Adversarial attacks on deep learning based mmwave beam prediction in 5g and beyond," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2021, pp. 590–594.

[12] Y. E. Sagduyu, R. A. Berry, and A. Ephremides, "Jamming games in wireless networks with incomplete information," *IEEE Communications Magazine*, vol. 49, no. 8, pp. 112–118, 2011.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[14] A. Smith and J. Downey, "A communication channel density estimating generative adversarial network," in *IEEE Cognitive Communications for Aerospace Applications Workshop*, 2019, pp. 1–7.

[15] H. Ye, L. Liang, G. Y. Li, and B.-H. Juang, "Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3133–3143, 2020.

[16] S. Zhao, Y. Fang, and L. Qiu, "Deep Learning-Based channel estimation with SRGAN in OFDM Systems," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.

[17] H. Jiang and L. Dai, "End-to-End Learning of Communication System without Known Channel," in *IEEE International Conference on Communications*, 2021, pp. 1–5.

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

[19] 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, 2020.

[20] D. Roy, T. Mukherjee, M. Chatterjee, and E. Pasiliao, "Defense against PUE attacks in DSA networks using GAN based learning," in *IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[21] X. Zhou, J. Xiong, X. Zhang, X. Liu, and J. Wei, "A Radio Anomaly Detection Algorithm Based on Modified Generative Adversarial Network," *IEEE Wireless Communications Letters*, 2021.

[22] A. Toma, A. Krayani, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "Deep Learning for Spectrum Anomaly Detection in Cognitive mmWave Radios," in *IEEE* 31^{st} *Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–7.

[23] B. Ozpoyraz, A. T. Dogukan, Y. Gevez, U. Altun, and E. Basar, "Deep learning-aided 6g wireless networks: A comprehensive survey of revolutionary phy architectures," *arXiv preprint arXiv:2201.03866*, 2022.

[24] I. Ahmad, M. Liyanage, S. Shahabuddin, M. Ylianttila, and A. Gurtov, "Design principles for 5g security," *A Comprehensive Guide to 5G Security*, pp. 75–98, 2018.

[25] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, "Security for 5g and beyond," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3682–3722, 2019.

[26] K. S. Germain and F. Kragh, "Channel prediction and transmitter authentication with adversarially-trained recurrent neural networks," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 964–974, 2021.

[27] 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.

[28] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *International conference on machine learning*. PMLR, 2017, pp. 2642–2651.

[29] M. Patel, X. Wang, and S. Mao, "Data augmentation with Conditional GAN for automatic modulation classification," in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, 2020, pp. 31–36.

[30] T. J. O'shea and N. West, "Radio machine learning dataset generation with gnu radio," in *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.

[31] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*. Springer, 2016, pp. 213–226.

[32] I. Lee and W. Lee, "UniQGAN: Unified Generative Adversarial Networks for Augmented Modulation Classification," *IEEE Communications Letters*, 2021.

[33] A. Sahin, I. Güvenç, and H. Arslan, "A Survey on Multicarrier Communications: Prototype Filters, Lattice Structures, and Implementation Aspects." *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, pp. 1312–1338, 2014.

[34] N. Michailow, M. Matthé, I. S. Gaspar, A. N. Caldevilla, L. L. Mendes, A. Festag, and G. Fettweis, "Generalized frequency division multiplexing for 5^{th} generation cellular networks," *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3045–3061, 2014.

[35] "Phydyas: physical layer for dynamic spectrum access and cognitive radio," FP7 european project, Tech. Rep., 2010.

[36] M. Bellanger, "FS-FBMC: An alternative scheme for filter bank based multicarrier transmission," 5^{th} *International Symposium on Communications, Control and Signal Processing*, pp. 1–4, 2012.

[37] V. Vakilian, T. Wild, F. Schaich, S. ten Brink, and J.-F. Frigon, "Universal-filtered multi-carrier technique for wireless systems beyond LTE," *Globecom Workshops*, pp. 223–228, 2013.

[38] L. Zhang, A. Ijaz, P. Xiao, A. ul Quddus, and R. Tafazolli, "Subband Filtered Multi-Carrier Systems for Multi-Service Wireless Communications," *Transactions on Wireless Communications*, vol. 16, pp. 1893–1907, 2017.

[39] M. J. Abdoli, M. Jia, and J. Ma, "Filtered OFDM: A new waveform for future wireless systems," 16^{th} *International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 66–70, 2015.

[40] "F-OFDM scheme and filter design," Huawei, HiSilicon, 3GPP TSG RAN WG1 Meeting 85, Tech. Rep., 2016.

[41] M. Jouhari, E. M. Amhoud, N. Saeed, and M.-S. Alouini, "A Survey on Scalable LoRaWAN for Massive IoT: Recent Advances, Potentials, and Challenges," *arXiv preprint arXiv:2202.11082*, 2022.

[42] A. Azougaghe, O. A. Oualhaj, M. Hedabou, M. Belkasmi, and A. Kobbane, "Many-to-one matching game towards secure virtual machines migration in cloud computing," in *IEEE International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, 2016, pp. 1–7.

[43] A. Bentajer, M. Hedabou, K. Abouelmehdi, and S. Elfezazi, "CS-IBE: a data confidentiality system in public cloud storage system," *Procedia Computer Science*, vol. 141, pp. 559–564, 2018.

[44] A. Rajakaruna, A. Manzoor, P. Porambage, M. Liyanage, M. Ylianttila, and A. Gurtov, "Enabling end-to-end secure connectivity for low-power iot devices with uavs," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*. IEEE, 2019, pp. 1–6.

[45] K. Zerhouni, E. M. Amhoud, and G. Noubir, "Deep Neural Networks for Multicarrier Waveforms Classification in UAV Networks," in *IEEE* 17^{th} *International Symposium on Wireless Communication Systems (ISWCS)*, 2021, pp. 1–6.

[46] J. Yang, H. Gu, C. Hu, X. Zhang, G. Gui, and H. Gacanin, "Deep complex-valued convolutional neural network for drone recognition based on rf fingerprinting," *Drones*, vol. 6, no. 12, p. 374, 2022.

[47] R. Akter, V.-S. Doan, J.-M. Lee, and D.-S. Kim, "Cnn-ssdi: Convolution neural network inspired surveillance system for uavs detection and identification," *Computer Networks*, vol. 201, p. 108519, 2021.

[48] K. Zerhouni, F. Elbahhar, R. Elassali, K. Elbaamrani, and N. Idboufker, "Influence of pulse shaping filters on cyclostationary features of 5G waveforms candidates," *Signal Processing*, vol. 159, pp. 204–215, 2019.

[49] K. Zerhouni, E. M. Amhoud, and M. Chafii, "Filtered Multicarrier Waveforms Classification: A Deep Learning Based Approach," *IEEE Access*, 2021.

[50] C. W. Korevaar, A. B. Kokkeler, P.-T. de Boer, and G. J. Smit, "Spectrum efficient, localized, orthogonal waveforms: closing the gap with the Balian-low theorem," *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 2155–2165, 2016.

[51] C. Heil and A. M. Powell, "Gabor Schauder bases and the Balian-Low theorem," *Journal of mathematical physics*, vol. 47, no. 11, p. 113506, 2006.

[52] E. M. Amhoud, M. Chafii, A. Nimr, and G. Fettweis, "OFDM with Index Modulation in Orbital Angular Momentum Multiplexed Free Space Optical Links," in *IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–5.

[53] J.-B. Doré, V. Berg, N. Cassiau, and D. Kténas, "FBMC receiver for multi-user asynchronous transmission on fragmented spectrum," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 41, 2014.

[54] M. V. Eeckhaute, A. Bourdoux, P. D. Doncker, and F. Horlin, "Performance of emerging multi-carrier waveforms for 5G asynchronous communications," *EURASIP J. Wireless Comm. and Networking*, vol. 2017, p. 29, 2017.

[55] R. Gerzaguet, N. Bartzoudis, L. G. Baltar, V. Berg, J.-B. Doré, D. Kténas, O. Font-Bach, X. Mestre, M. Payaró, M. Färber *et al.*, "The 5G candidate waveform race: a comparison of complexity and performance," *EURASIP Journal on Wireless Communications and Networking*, no. 1, pp. 1–14, 2017.

[56] M. Hedabou, "A frobenius map approach for an efficient and secure multiplication on Koblitz curves," *International Journal of Network Security*, vol. 3, no. 3, pp. 239–243, 2006.

[57] J. Brownlee, *Generative adversarial networks with Python: deep learning generative models for image synthesis and image translation*. Machine Learning Mastery, 2019.

[58] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[59] D. Kingma and J. Ba, "Adam: A method for stochastic optimization Ba J," *International Conference on Learning Representations*, 2014.

[60] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.

[61] "User Equipment (UE) Radio Transmission and Reception," 3GPP, TS 36.101, 2016.

[62] A. Borji, "Pros and cons of GAN evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[63] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *Advances in neural information processing systems*, vol. 29, 2016.

[64] 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," *Advances in neural information processing systems*, vol. 30, 2017.

[65] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[66] 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," *Advances in neural information processing systems*, vol. 30, 2017.

[67] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," in 7^{th} *International Conference on Learning Representations, ICLR*, 2019, pp. 2818–2826.