Generative Adversarial Network in the Air: Deep Adversarial Learning for Wireless Signal Spoofing

Kawtar ZERHOUNI[1], Abderrahmane SALEHI[1], Badreddine SAADIOUI[1],
Ghita NAHLI[1], Jaafar SELLAKH[1], and Salma BENSLIMANE[1]

[1]Ecole Centrale Casablanca, Casablanca, Morocco

Abstract—**This study introduces innovative advancements in the application of Generative Adversarial Networks (GANs) in the realm of wireless communications, expanding upon the foundational concepts outlined in existing literature. Our research deepens the understanding of GANs in adversarial settings and paves the way for more advanced signal spoofing methods using GANs, to generate and transmit synthetic signals that cannot be reliably distinguished from intended signals. Highlighting our distinctive approach, we broke away from the conventional method of using single-carrier waveforms and instead introduced a pioneering technique that involves modeling multicarrier waveforms (UFMC, OFDM, F-OFDM, GFDM, FBMC). This departure from the traditional approach underscores the innovative nature of our research, as we ventured into uncharted territory by embracing the complexities of multicarrier waveform, a departure from the standard practice in the field. The adversarial model used comprises a transmitter-receiver pair, collaboratively orchestrating a mini-max game to optimize signal spoofing. This research demonstrates increased efficiency and reduced latency in signal spoofing, highlighting the potential of GANs in complex network typologies and mobility patterns.**

Index Terms—**Conditional generative adversarial networks, deep convolutional neural networks,spoofing attack, multicarrier waveform.**

I. Introduction

Wireless communications, being characterized by its open and shared nature, is intrinsically susceptible to adversarial attacks. The very ubiquity and accessibility that make wireless communication so vital in today's world also expose it to potential threats. One of the common tactics employed by adversaries in wireless signal spoofing involves capturing a legitimate transmission and replaying it at a later time, often with adjustments to the transmission power. While this approach can represent various features in the signal at a high level, it may fall short of reliably mimicking combined waveform characteristics, channel conditions, and device-specific effects.

In this context, machine learning has gained prominence in the field of wireless communications, finding applications in various aspects such as spectrum sensing [1] and modulation recognition [2]. However, the majority of existing ML systems operate under the assumption that data originates from regular users and is independently generated from the same distribution [1]. While some ML algorithms can handle small dense noises and large sparse outliers, a limited number of them address adversarial noises intentionally crafted by individuals with knowledge of the ML system and its data. These adversaries introduce meticulously crafted noises or manipulate the dataset to undermine or deceive the learning system, posing a significant threat, especially in security and safety-critical domains [1].

Generative Adversarial Networks (GANs) have gained attention for their ability to generate perturbations and realistic examples in various problem domains since their introduction in 2014 [3]. In wireless communications, most GAN research has concentrated on single carrier signals, addressing applications such as data augmentation, wireless channel modeling, physical layer design, adversarial attacks, and anomaly detection [4].

However, the emergence of filtered multicarrier waveforms in wireless communication standards has introduced complexity and diversity into the field. Unlike single-carrier systems, filtered multicarrier systems involve multiple carriers with specific filtering requirements and inter-carrier relationships. This complexity demands more sophisticated ML models, increased computational resources, and the need for models to generalize well to variations in 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.

Additionally, in the evolving landscape of machine learning, there has been a growing emphasis on the challenge of training resilient models under constraints of limited datasets. Traditional remedies, such as data duplication, often lead to over-fitting, compromising the model's ability to generalize effectively. To address this ongoing dilemma and push the boundaries of what's possible, we propose a pioneering solution: the utilization of Generative Adversarial Networks (GANs) to generate synthetic signals. These synthetic signals are not mere replicas; they are authentically distinct from the original signals, tailored explicitly for multi-signal carrier waveforms rather than single-carrier waveforms. This marks a significant departure from established practices and represents the first application of GANs in this domain, opening new avenues for improved model training in signal processing applications.

In this paper, we embark on a groundbreaking mission that extends beyond the traditional boundaries of machine learning in wireless security. Our objective is nothing short of revolutionary—to harness the power of GANs from an adversarial standpoint. We aim to train a GAN to generate wireless signals that are practically indistinguishable from genuine transmissions.

This study aims to explore the potential of GAN-based modeling for base-band filtered multicarrier waveforms directly from raw I/Q components of quadrature phase shift keying (QPSK) data. The focus of the investigation includes per-sub-carrier filtered multicarriers (e.g., filter bank multicarrier ”FBMC” and generalized frequency division multiplexing ”GFDM”) and per-sub-band filtered multicarriers (e.g., universal filtered multicarrier ”UFMC” and filtered orthogonal frequency division multiplexing ”F-OFDM”). The primary objective is to create an evasion attack where an adversary attempts to fool a machine learning algorithm, in our case a CNN classifier (Convolutional Neural Network), by generating synthetic waveforms that are statistically analogous to legitimate ones. The CNN must navigate increasingly sophisticated adversarial tactics to discern between real and fake signals. This research demonstrates the practicality of this proposal and provides a foundation for comprehensive and complete datasets.

The forthcoming sections of this paper will delve into the intricate technical details of our approach, the experimental framework we employed, and the results we achieved. Through this exploration, we aim to shed light on the transformative potential of GANs in redefining the landscape of wireless security.The paper’s structure includes a brief description of the investigated waveforms, details on the proposed GAN model, training steps, simulation parameters, discussions on findings, and concluding remarks with future perspectives in subsequent sections.

II. SIGNAL THEORY

Wireless communication systems are inherently challenged by phenomena like frequency selectivity, interference, and fading within propagation channels. Frequency selectivity refers to the differential treatment of various frequencies in a signal due to path loss, shadowing, and multi-path, leading to certain frequencies suffering more attenuation and phase shifts than others. Interference, both from within the network (like co-channel and adjacent channel interference) and from external sources, further complicates the signal transmission and reception process. Fading, caused by the constructive and destructive superposition of the multipath signals, leads to rapid fluctuations in the amplitude and phase of the received signal.

A. Multicarrier Modulation (MCM)

Multicarrier Modulation (MCM) is a technique that has been extensively adopted to counter these challenges. It involves transmitting data over multiple carrier frequencies, thus dividing the overall available bandwidth into numerous smaller, orthogonal subchannels. This division is pivotal in managing the intricacies of wireless channels and enhancing spectral efficiency and communication reliability. In MCM, the total available bandwidth is strategically subdivided into multiple smaller subchannels, with each subchannel functioning as an individual carrier for data transmission. This subdivision plays a crucial role in mitigating the impact of frequency-selective fading, a phenomenon where different frequencies in a signal are affected differently during transmission. By dispersing data transmission across various frequencies, MCM ensures that even if some subchannels are compromised due to fading, others continue to transmit effectively, thus preserving the overall integrity of the data communication. Furthermore, data symbols, which represent the basic units of digital information, are modulated onto these subcarriers. Each symbol is meticulously mapped onto a specific subcarrier and allocated specific time intervals, enhancing the modulation strategy’s adaptability to suit diverse communication scenarios. The temporal and spectral distribution of these symbols across the subchannels facilitates a flexible and efficient approach to data transmission. An essential aspect of MCM is the encoding of data using In-phase (I) and Quadrature (Q) components, wherein each subcarrier is represented by a complex waveform that includes both I and Q parts. This encoding mechanism optimizes the information transmission process, making efficient use of the available bandwidth. In some implementations of MCM, a binary encoding method that utilizes just 2 bits is adopted, striking a crucial balance between achieving high spectral efficiency and maintaining manageable signal complexity. This balanced approach underscores the innovative nature of MCM in handling the challenges of modern wireless communication.

In the realm of MCM, several key techniques have been developed, each addressing specific aspects of wireless communication challenges. Orthogonal Frequency Division Multiplexing (OFDM) is a widely embraced technique, renowned for its subcarrier orthogonality. OFDM excels in high data rate applications, demonstrating resilience to multipath fading and spectral leakage, making it a mainstay in various communication systems. Addressing some of OFDM’s limitations, particularly concerning out-of-band radiation, is the Filter Bank Multicarrier (FBMC) technique. FBMC utilizes sophisticated filtering for each subcarrier, significantly improving spectral containment and minimizing inter-carrier interference, thus enhancing the overall spectral efficiency of the system. Another innovative approach is Generalized Frequency Division Multiplexing (GFDM), which introduces subsymbols and circular filtering. This method strikes a balance between spectral efficiency and receiver complexity, marking it as a contemporary advancement in MCM. Expanding the spectrum of MCM techniques are Universal Filtered Multicarrier (UFMC) and Filtered-OFDM (F-OFDM). UFMC, inspired by resource block allocation in 4G systems, proposes filtering groups of subcarriers, known as subbands, offering moderate out-of-band emissions and circumventing some drawbacks of FBMC. On the other hand, F-OFDM uses longer filters compared to UFMC, thus presenting better spectral containment, albeit with a tendency towards increased inter-symbol interference.

These diverse techniques under the MCM umbrella each contribute uniquely to tackling the intricate challenges in wireless communication, enhancing the efficacy and reliability of data transmission in modern communication networks [5]-[6].

B. Analysis of Multicarrier Transmitted Signal

In the framework of multicarrier communication systems, the transmitted signal in a multicarrier setup can be mathematically formulated as follows:

$$x[n] = \sum_{r=0}^{N_{\text{symbols}}-1} \sum_{q=0}^{N_c-1} s_{q,r} g_{q,r}[n] \tag{1}$$

In this expression, $s_{q,r}$ are the data symbols that are zero-mean, independent, and identically distributed (i.i.d) random variables. These symbols are typically derived from a Quadrature Amplitude Modulation (QAM) constellation and are carried on the q-th subcarrier during the r-th symbol period. The term $g_{q,r}[n]$ represents the synthesis function, which is responsible for mapping the data symbols onto the corresponding signal dimension in the time-frequency grid.

The synthesis function $g_{q,r}[n]$, a crucial component in a Gabor System, is articulated as:

$$g_{q,r}[n] = g_{\text{tx}}[n - rN] e^{\frac{-j2\pi qn}{N_c}} \tag{2}$$

Here, $g_{\text{tx}}[n]$ is identified as the transmit prototype filter, which forms the basis for generating various pulse shapes through time shifting and frequency modulation. This transmit prototype filter plays a pivotal role in determining the characteristics of the transmitted signal, such as its bandwidth and time-frequency localization properties.

The received signal, which is the counterpart of the transmitted signal as per Eq. (1), undergoes various transformations due to the propagation environment. It can be represented as:

$$r_{\text{ch}}[n] = x[n] * h[n] = \sum_{j=0}^{L_{\text{ch}}-1} h[\eta_j] e^{-j2\pi f_d^j n} x[n - \eta_j] \tag{3}$$

In this formula, L_{ch} signifies the number of multipath components in the channel, η_j denotes the discrete propagation delay for each path, and $h[\eta_j]$ is the time-varying channel gain associated with each path. The term $f_d^j = \frac{c f_c v_d}{c}$ represents the Doppler shift, with f_c being the carrier frequency, v_d the relative speed between the transmitter and receiver, and c the speed of light.

The final received signal, considering various practical aspects such as symbol time offset δ, carrier frequency offset κ, and the presence of Additive White Gaussian Noise (AWGN) $z[n]$ with variance σ_z^2, is given by:

$$y[n] = e^{\frac{j2\pi n\kappa}{N_c}} r_{\text{ch}}[n + \delta] + z[n] \tag{4}$$

This equation succinctly encapsulates the composite effects of carrier frequency offset, symbol time offset, and the influence of noise on the received signal. It highlights the complexities involved in the signal reception in a multicarrier communication system, where factors like channel multipath effects, Doppler shifts due to mobility, and noise play significant roles in determining the quality and reliability of the received signal [7].

III. GAN Architecture for Multicarrier Signals

A. Introdction to GAN

Generative Adversarial Networks (GANs) have emerged as a revolutionary concept in machine learning, particularly in the realm of wireless communications. Originally conceptualized by Ian Goodfellow et al. in 2014, GANs consist of two neural networks—the generator and the discriminator each being trained to outperform the other. This architecture enables GANs to generate highly realistic data, which can be used in diverse applications ranging from image synthesis to complex signal processing tasks in wireless systems.In wireless communications, GANs have shown promising applications, particularly in creating complex, realistic wireless signal environments for testing and evaluation.

The Gan's Generator G is a neural network that generates synthetic samples given a random noise, sampled from a fixed length vector z, also known as a latent space, from a Gaussian distribution. G learns to map z to the dataset distribution. On the other hand, the Discriminator D, also a neural network, is a binary classifier that discriminates between whether the input sample is real [output a scalar value 1] or fake [output a scalar value 0]. 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. 1.

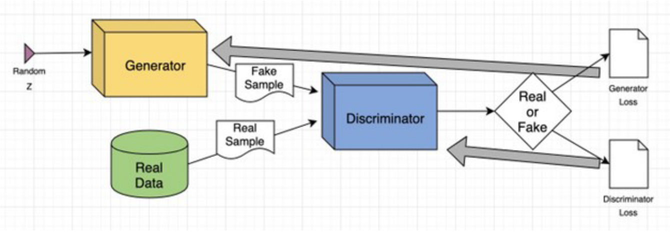


Fig. 1. The GAN's architecture

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. 1. 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:

$$\Theta = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{5}$$

In GANs for multi-carrier signals, the discriminator $D(x)$ estimates the likelihood of authentic data x being real. Conversely, the generator $G(z)$ creates outputs from noise input z, and $D(G(z))$ is the discriminator's estimate of the probability that these outputs are real. The generator, not having access to real data, focuses on minimizing $\log(1-D(G(z)))$. However, this often leads to saturation issues; thus, the generator aims to maximize $\log(D(G(z)))$.

GANs can generate credible samples but lack direct control over the types of fake instances produced. This is particularly challenging for multi-carrier waveforms which are not easily interpretable by humans.Hence, we explore a class conditional variant, cGAN, where additional information (like class labels) is incorporated into the input data. This is achieved by combining an embedding layer with a fully connected one to integrate this data as an added feature map. The GAN architecture for 2×128 I/Q signals includes convolutional layers for both discriminator and generator, with specific kernel sizes and strides, and utilizes techniques like Leaky ReLU, dropout, and Adam optimizer for training. The generator transforms latent space points into plausible I/Q vectors, using dense and upsampling layers, culminating in a final output layer with a Tanh activation function. An embedding layer is used to map class labels into distinct vectors.

B. Maths behind GAN

Generative Adversarial Networks (GANs) consist of two main components: the **Discriminator** and the **Generator**. The discriminator aims to learn the conditional probability $P(Y|X=x)$, where X is the input and Y is the output. Meanwhile, the generator aims to learn the joint probability $P(X,Y)$.The discriminator is essentially a binary classifier that distinguishes between real data (X,Y) and generated data $(\hat{X},\hat{Y})$. Its objective is to maximize the probability of assigning the correct label to real data and the correct label to generated data. Mathematically, the discriminator aims to maximize:

$$E_D = \mathbb{E}_{(X,Y)\sim P_{\text{data}}}[\log D(X,Y)] + \mathbb{E}_{(\hat{X},\hat{Y})\sim P_{\text{gen}}}[\log(1-D(\hat{X},\hat{Y}))]$$

Here, $D(X,Y)$ is the output of the discriminator for real data, and $D(\hat{X},\hat{Y})$ is the output for generated data. Conversely, the generator aims to fool the discriminator by generating samples that are indistinguishable from real data. Its objective is to minimize the probability of the discriminator correctly labeling generated data as fake. Mathematically, the generator aims to maximize:

$$E_G = \mathbb{E}_{(\hat{X},\hat{Y})\sim P_{\text{gen}}}[\log D(\hat{X},\hat{Y})]$$

In Generative Adversarial Networks (GANs), the value function $V(G,D)$ is defined as:

$$V(G,D) = \mathbb{E}_{x\sim P_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim P_z}[\log(1-D(G(z)))]$$

Here, x is sampled from the real data distribution P_{data}, and z is sampled from the input noise distribution P_z of the generator.

Proof : Derivation using Binary Cross-Entropy

The binary cross-entropy loss function is commonly used in GANs. It is given by:

$$E(y,\hat{y}) = -\left(y\log(\hat{y}) + (1-y)\log(1-\hat{y})\right)$$

Now, considering the two scenarios in the GAN value function:

- When $y=1$, $\hat{y}=D(x)$, and $E=\log(D(x))$.
- When $y=0$, $\hat{y}=D(G(z))$, and $E=\log(1-D(G(z)))$.

Adding these expectations, we get the value function:

$$V(G,D) = \mathbb{E}_{x\sim P_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim P_z}[\log(1-D(G(z)))]$$

This formulation reflects the adversarial nature of GANs, where the discriminator (D) and the generator (G) play a minimax game to achieve a balance.

The training of Generative Adversarial Networks (GANs) involves an iterative process where the generator and discriminator are updated alternatively. The algorithm consists of two main loops: an inner loop for updating the discriminator (D) and an outer loop for updating the generator (G).

- **Training Loop:**

```
Fix the learning rate
For each iteration:
  Inner loop for $D$ :
  - Take $m$ real data samples and $m$ fake data samples
  - Update parameters $\theta_D$ by gradient descent:
    $\frac{\partial}{\partial\theta_D}\frac{1}{m}\left(\ln(D(x)) + \ln(1-D(G(z)))\right)$
  Fix the learning of $D$ (out of the inner loop for $D$ )
  - Take $m$ fake data samples
  - Update parameters $\theta_G$ by gradient descent:
    $\frac{\partial}{\partial\theta_G}\frac{1}{m}\ln(1-D(G(z)))$
```

For every k updates of the discriminator, update the generator once.

At optimality, for a fixed G, the value function $V(G,D)$ is maximized when $D(x) = \frac{P_{\text{data}}(x)}{P_{\text{data}}(x)+P_G(x)}$. The value function is given by:

$$V(G,D) = \mathbb{E}_{x\sim P_{\text{data}}}\left[\frac{P_{\text{data}}(x)}{P_{\text{data}}(x)+P_G(x)}\right] + \mathbb{E}_{x\sim P_G}\left[\frac{\ln P_G(x)}{P_{\text{data}}(x)+P_G(x)}\right]$$

Minimizing V with respect to G leads to:

$$\min_G V = \frac{1}{2}(\mathbb{E}_{x\sim P_{\text{data}}}[1] + \mathbb{E}_{x\sim P_G}[1]) = 1$$

Now, consider the Jensen-Shannon Divergence (JS) between P_{data} and P_G, given by:

$$\text{JS}(P_{\text{data}}, P_G) = \frac{1}{2}\left(\text{KL}\left(P_{\text{data}}\middle\|\frac{P_{\text{data}}+P_G}{2}\right) + \text{KL}\left(P_G\middle\|\frac{P_{\text{data}}+P_G}{2}\right)\right)$$

where KL is the Kullback-Leibler divergence. The minimum value of V is related to JS:

$$\min_G V = 2 \cdot \text{JS}(P_{\text{data}}, P_G) - 2\ln 2$$

This is minimized when $P_{\text{data}} = P_G$, as the JS divergence is always non-negative.

Therefore, at optimality, $P_G = P_{\text{data}}$.

C. Dataset

The dataset is structured with the following dimensions: first, it encompasses a multitude of samples, with each class and signal-to-noise ratio (SNR) value represented by 2500 examples. The dataset captures the in-phase (I) and quadrature (Q) components of signals, resulting in two I/Q components. The sequence length of 128 signifies the temporal nature of the data, and it's worth noting that the dataset encompasses 5 waveform types, including UFMC, OFDM, F-OFDM, GFDM, and FBMC. Additionally, the dataset exhibits variations in SNR values, making it a rich resource for the research. This 5-dimensional dataset serves as the foundation for the work, enabling us to explore and analyze signals with diverse characteristics and applications.

D. GAN and CNN Architecture

The discriminator in our GAN architecture serves the purpose of binary classification, determining whether a given input instance is real or fake. It is constructed with four 2D convolutional layers, each featuring a kernel size of 4x4 and a stride of 2, enabling it to capture hierarchical features. Unlike traditional CNNs, we do not employ pooling layers in our design. The discriminator's output layer consists of a single node, incorporating a Sigmoid activation function to provide binary classification results. During training, the discriminator aims to minimize the binary cross-entropy loss, aligning its predictions with the ground truth labels. To enhance training stability and prevent sparse gradients, we opt for Leaky Rectified Linear Units (ReLU) with a slope of 0.2. Additionally, a dropout layer with a rate of 0.3 is introduced to facilitate improved model generalization. For optimization, we employ the Adam variant of stochastic gradient descent, configuring a learning rate of 0.0002 and a momentum value of 0.5.

The generator within our GAN architecture is tasked with transforming a latent space point into a credible 2x128 In-Phase/Quadrature (I/Q) vector. To achieve this, we employ a structured approach, initiating with a dense layer responsible for generating 256 parallel versions of a smaller I/Q vector, each comprising a 1x8 shape and encompassing distinct learned features. Subsequently, these vectors are subjected to transpose convolution layers, known as deconvolution layers, which encompass both upsampling and convolutional aspects. The kernel size of these layers is set at 4x4, with a stride of 2x2, effectively doubling the size of the I/Q vectors at each layer. The Leaky Rectified Linear Unit (ReLU) activation function is applied to the output of each layer to introduce non-linearity into the model. The concluding layer of the generator model consists of a 2D convolution layer with a solitary filter. For the purpose of stabilizing the GAN training process, we incorporate a Tanh activation function to ensure that the output values fall within the desired range of [-1, 1]. In parallel, we employ an embedding layer of size 50 to inject class label information, with each of the five waveform classes being mapped to a distinct 50-element vector interpretation.

In our study, we adopted a Convolutional Neural Network (CNN) model architecture for the classification of filtered multicarrier waveforms. The CNN architecture comprises four layers, organized into two convolutional layers followed by two fully connected dense layers. The first three layers utilize Rectified Linear Unit (ReLU) activation functions, while the final layer employs a SoftMax transfer function for classification. To tailor the network's hyperparameters to our specific dataset, which includes six classes, including a novel class of random data in addition to the investigated waveforms, we configured the following settings: In the initial convolutional layer, we employed 128 filters with a size of 1 × 3. Subsequently, in the second convolutional layer, we reduced the number of filters to 64, each with a size of 1 × 3. The dense layer was configured with a length of 128 neurons, corresponding to the size of our dataset's features. The output layer was structured with one neuron dedicated to each class label, totaling six neurons to accommodate the multi-class classification task.

To prevent overfitting and enhance the model's generalization capabilities, we applied dropout regularization after each of the first three layers. The dropout rate (d_r) was set to 0.3. Additionally, we incorporated an early stopping mechanism into our training process. This technique helps halt the training process when the model's performance on a validation dataset no longer improves, further mitigating the risk of overfitting. To optimize the model's learning process, we employed the Adam optimization algorithm, an advanced variant of traditional stochastic gradient descent. These strategies collectively ensured the robustness and generalization of our CNN model for accurate classification of filtered multicarrier waveforms, including the novel class of random data in our extended dataset.

IV. Methodology

In our study, we focused on assessing the performance of Generative Adversarial Networks (GANs) when applied to filtered multicarrier signals. To gain a better understanding of GAN performance in this context, we utilized various simulations and worked with a comprehensive dataset. This dataset consisted of signals with a fixed symbol length of $N_{\text{symbols}} = 10$. We varied the Signal-to-Noise Ratio (SNR) in the range from -8 dB to 20 dB in steps of 2 dB, and we ensured that each SNR level had 5000 examples per class.

To maintain consistency in symbol lengths across different multicarrier waveforms, we kept the number of subcarriers constant at $N_c = 16$ for most waveforms, with the exception of GFDM, where $N_c = 8$ was used. The prototype filters, which are well-established in the literature and previously studied for waveform classification, were employed for signal generation. Specifically, we used the Root Raised Cosine (RRC) filter with a roll-off factor $\beta = 0.35$ for GFDM, the Phydyas filter with an overlapping factor of $K = 2$ for FBMC, a Chebyshev filter for UFMC, and a truncated sinc filter for F-OFDM. The data symbols were drawn from a Quadrature Phase Shift Keying (QPSK) constellation.

In terms of additional simulation details, we set the cyclic prefix length to $N_c/4$ to ensure it was greater than the maximum channel delay, thereby avoiding Intersymbol Interference (ISI). We introduced multipath channel effects using the extended typical urban (ETU) model, specifically opting for the extended typical urban (ETU) model from the Long-Term Evolution (LTE) channel model. **To summarize, our role primarily involved receiving and working with the provided dataset, without directly intervening in the signal design or reception process.**

The dataset was received as a Matlab file , whereas we built the deep learning models in Python, relying on Keras and TensorFlow libraries, and trained them over the T4 GPU of Google Colab Pro.

A. Implementing the algorithms and training

We initiate our process with the Convolutional Neural Network (CNN) classifier. Initially, we load and rescale the 2x128 In-Phase/Quadrature (I/Q) data to a range of [-1, 1]. Subsequently, we label the data into the five distinct signal types(UFMC,OFDM,F-OFDM,GFDM,FBMC), forming a vector Y. The architecture of our CNN model includes multiple layers for feature extraction and classification. In our training procedure, 80% of the data is allocated for training, while the remaining 20% is reserved for validation and testing. Prior to training, the input training data is normalized to achieve a zero mean and a standard deviation of unity. The CNN model is trained using 2500 examples per class per Signal-to-Noise Ratio (SNR) ranging from 1 to 2500, for 100 epochs.

In parallel to our CNN classifier, we implemented the Generative Adversarial Network (GAN) using the same scaled data and label information. The GAN architecture includes the generator, the discriminator, and GAN model. Drawing inspiration from our previously trained CNN classifier, we initiated the process by loading and rescaling the 2x128 In-Phase/Quadrature (I/Q) data and arranging the data into labeled categories. Subsequently, we constructed the generator and discriminator networks. In the context of this study, we focused on refining the GAN's hyperparameters through a series of simulations. Specifically, we trained the GAN using the received waveforms, comprising 2500 examples per waveform for three Signal-to-Noise Ratio (SNR) values: 20 dB, 18 dB, and 16 dB. The training regimen spanned 100 epochs, with periodic saving of the generator model every 10 epochs.

B. Evaluation Metrics

Evaluating the performance of Generative Adversarial Networks (GANs) remains an ongoing research challenge within the computer vision community. When assessing the effectiveness of a GAN architecture, various quantitative and qualitative metrics have been introduced to scrutinize not only the quality but also the diversity of the generated data. In our specific context, where our objective is to replicate signals transmitted by legitimate users, we consider the GAN to have reached convergence when it can successfully deceive the CNN classifier employed at the receiving end. Leveraging the capabilities of a conditional GAN, we introduce the following key metrics:

- Probability of Global Correct Classification (P_{gcc}):

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

Here, N_w represents the total count of waveforms utilized for training the GAN, while N_f denotes the number of fake signals generated per waveform for GAN evaluation. The N_{cc} term corresponds to the cumulative count of correct classifications across all waveforms. It increments each time the labels predicted by the CNN classifier align with the conditional labels used by the GAN's generator to produce fake signals.

Additionally, we introduce the Per-Class Correct Classification Probability (P_{ccc}), defined as:

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

In this equation, N_f represents the total number of synthetic signals used for evaluation, and p_{c_i} denotes the per-class probabilities reported by the CNN classifier (denoted as p_c in Equation (7)). These probabilities are calculated individually for each class across the N_f synthetic signals and are then averaged to obtain the overall Per-Class Correct Classification Probability (P_{ccc}).

In our evaluation process, we incorporate two widely recognized metrics for assessing the quality and diversity of generated data: the Inception Score (IS) and the Frechet Inception Distance (FID) score. These metrics are traditionally applied to image evaluation, making them valuable benchmarks for our task. IS measures image diversity and quality, utilizing the high-performing Inception v3 image classification model. It calculates conditional probabilities $p(y|x)$ for each image, and marginal probabilities $p(y)$ by averaging the conditional probabilities across synthetic images within a group. The final IS score results from a sum of KL divergences over all images, with an average across all classes followed by exponentiation. The IS ranges from 1 to the number of classes in the classification model. We intend to adapt this concept by replacing Inception v3 with a CNN model tailored for filtered multicarrier waveforms classification. Similarly, for FID, which evaluates real and fake data based on statistical features, we aim to replace Inception v3 with our CNN model. We will omit the output layer and extract features from the last dense layer, calculating their mean and covariance to form a multivariate Gaussian. Real signals, used during GAN training,

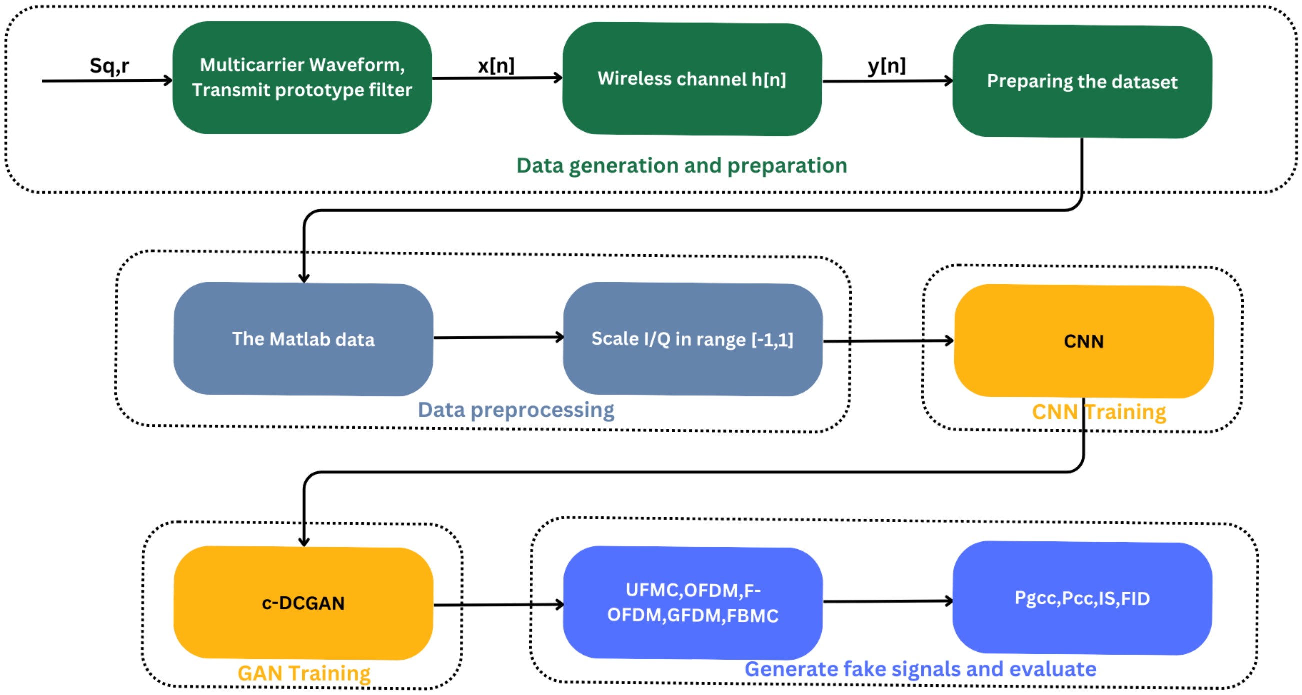

Sq,r
Multicarrier Waveform, Transmit prototype filter
x[n]
Wireless channel h[n]
y[n]
Preparing the dataset
Data generation and preparation
The Matlab data
Scale I/Q in range [-1,1]
Data preprocessing
CNN
CNN Training
c-DCGAN
GAN Training
UFMC,OFDM,F-OFDM,GFDM,FBMC
Pgcc,Pcc,IS,FID
Generate fake signals and evaluate

Fig. 2. Steps and methodology for generating fake signals using a c-DCGAN

will be compared to fake signals generated by the GAN. We acknowledge the challenges in adapting these image-centric metrics to signal data, but we are committed to exploring their potential applicability.

Fig. 2. summarizes our methodology.

V. Results and Discussion

A. Performance of the CNN classifier

In our study, we trained a Convolutional Neural Network (CNN) model using 2500 examples per class for each Signal-to-Noise Ratio (SNR) value ranging from 1 to 2500. Our objective was to achieve high classification accuracy across all SNR values, and we successfully obtained an overall accuracy of 0.992. Figure 3 provides a comprehensive overview of the CNN's performance:

a. Fig. 3a displays the CNN's learning curve, showing the training and validation accuracy. It illustrates that our model learned effectively, with a good fit between the training and validation data.

b. Fig. 3b shows how the CNN's accuracy changes as we vary the SNR values. This graph indicates that the CNN's performance varies with the SNR, and certain modulation schemes perform better than others.

c. Fig. 3c presents per-class accuracy as a function of SNR. Notably, F-OFDM classification consistently outperformed UFMC, GFDM, OFDM, and FBMC. This suggests that the CNN effectively captures unique statistical attributes created by applying filtering to these waveforms, enabling precise classification of these novel transmission schemes.

d. Fig. 3d illustrates the confusion matrix for all SNR values. It reveals that there is significant confusion between GFDM and OFDM, and vice versa. This can be explained by the fact that OFDM can be considered a special case of GFDM when the number of subsymbols is set to 1.

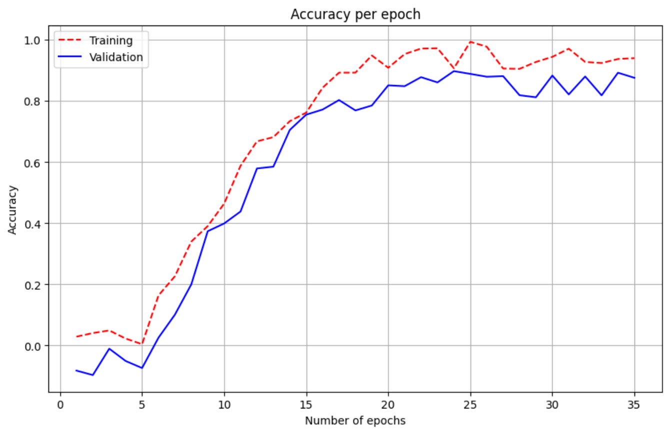


Fig. 3.a. CNN's accuracy per epoch

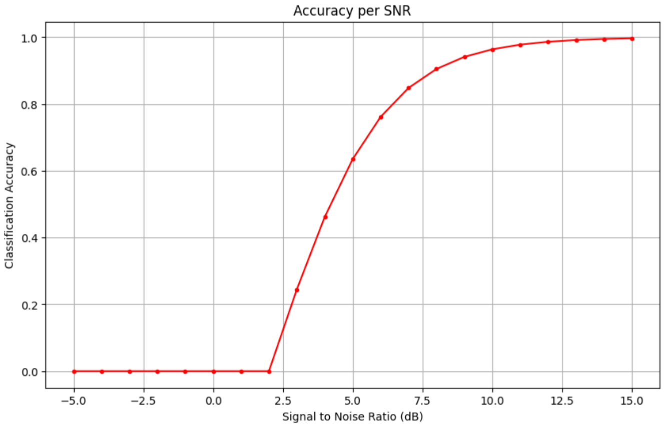


Fig. 3.b. CNN's accuracy per SNR

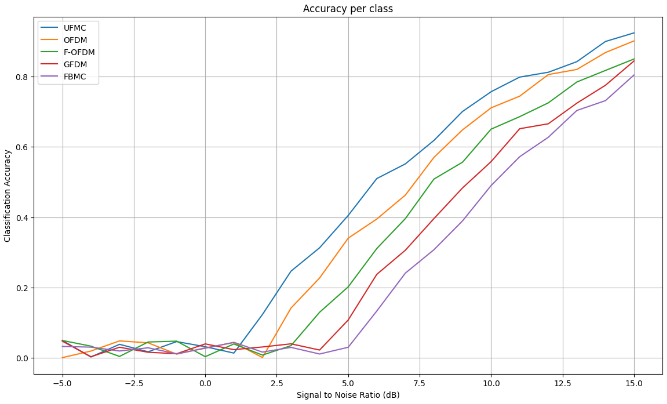


Fig. 3.c. CNN's accuracy per class

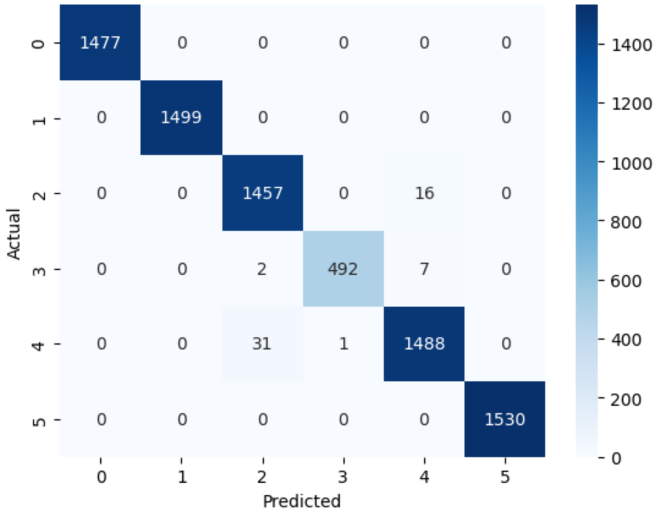
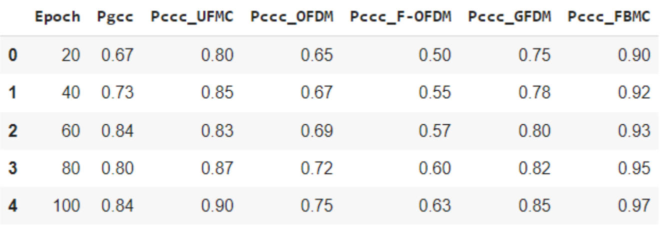


Fig. 3.a. CNN's confusion matrix

In summary, our CNN model demonstrated strong performance in classifying various modulation schemes based on distinctive statistical features induced by waveform filtering. The confusion between GFDM and OFDM can be attributed to the similarity between these two schemes, particularly when OFDM is treated as a special case of GFDM with one subsymbol.

B. B. Performance of the GAN

We trained the GAN for 100 epochs, and calculated the P_{gcc} and the P_{ccc} every epoch, then, for every 20 epochs, we stored the metrics in a table (Fig. 4.).



	Epoch	Pgcc	Pccc_UFMC	Pccc_OFDM	Pccc_F-OFDM	Pccc_GFDM	Pccc_FBMC
0	20	0.67	0.80	0.65	0.50	0.75	0.90
1	40	0.73	0.85	0.67	0.55	0.78	0.92
2	60	0.84	0.83	0.69	0.57	0.80	0.93
3	80	0.80	0.87	0.72	0.60	0.82	0.95
4	100	0.84	0.90	0.75	0.63	0.85	0.97

Fig. 4. The GAN's performance

Based on the table which shows the performance metrics Pgcc (Probability of Correct Class) and Pccc (Per-Class Correct Class) for a Generative Adversarial Network (GAN) over a series of epochs, we can infer the following:

1. Overall Trend: The Pgcc shows an upward trend as the epochs increase, which indicates that the GAN's ability to generate signals that are correctly classified by the CNN classifier is improving over time. Starting at a Pgcc of 0.67 at epoch 20, there is a consistent improvement, peaking at 0.84 by epoch 100.

2. Per-Class Analysis:

- UFMC: The performance for UFMC signal classification started strong at 0.80 and saw an incremental increase, reaching 0.90 by epoch 100. This suggests that the model is quite effective in learning to generate this type of signal correctly.

- OFDM: There was a modest improvement in the classification of OFDM signals, with Pccc starting at 0.65 and increasing to 0.75. There's room for improvement, but the trend is positive.

- F-OFDM: This signal type had the lowest initial performance at 0.50 Pccc, but it shows a steady increase in classification accuracy, suggesting that given more epochs, the performance could continue to improve.

- GFDM: Starting at 0.75, the performance saw some fluctuations but generally improved, indicating that the model is fairly consistent in generating this signal type.

- FBMC: The Pccc for FBMC is high throughout, starting at 0.90 and ending at 0.97, which is an excellent classification rate and suggests that the GAN has a strong ability to generate FBMC signals that are close to the real signals.

3. Potential for Improvement: The results suggest that while the GAN is learning and improving its generation of signals as judged by the CNN classifier, there is still potential for further enhancement. Training the GAN for more than 100 epochs could likely result in higher Pgcc and Pccc metrics across all signal types, as there is a clear upward trend in the results. Training on multiple layers might also allow the GAN to capture more complex features of the data, leading to improved performance. However, it's important to balance this against the risk of overfitting, where the GAN might generate signals that are too tailored to the training dataset and may not generalize well.

4. Constraints and Limitations: A limitation in the training was the time constraint, which restricted the number of epochs to 100. Given more time, the model could have been trained for a larger number of epochs, potentially leading to a plateau of performance where further training does not significantly improve the results, which is a common occurrence in deep learning models. The inability to run the Inception Score (IS) and Fréchet Inception Distance (FID) metrics due to their difficult adaptability to signals as opposed to images is also a notable limitation. These metrics are standard for evaluating the quality of images generated by GANs, and their adaptation to signal data is not straightforward. The absence of these metrics means that the analysis relies solely on the CNN classifier's ability to recognize the generated signals, which may not fully capture the fidelity and diversity of the generated data.

In conclusion, the GAN shows promising results in generating synthetic signals that are recognized by a CNN classifier. While performance is good and improving, the full potential of the model has not been realized due to time constraints and the challenge of adapting certain evaluation metrics to signal data. With additional training time, exploration of more complex model architectures, and development of more suitable evaluation metrics for signals, there is potential to achieve even higher performance.

VI. Conclusion and Perspectives

This study has successfully demonstrated the innovative application of Generative Adversarial Networks (GANs) in wireless signal spoofing, specifically focusing on an evasion attack where the adversary transmitter generates forged signals using a conditional GAN. These signals, based on advanced filtered multi-carrier waveforms like UFMC, F-OFDM, GFDM, FBMC, and the traditional OFDM, are indistinguishable from intended ones. The effectiveness of this approach is underscored by its ability to deceive a CNN classifier designed to differentiate between various waveforms and random signals. While the classifier readily recognizes random signal attacks, it often misclassifies those generated by the GAN, highlighting the model's sophistication.

Future research should focus on refining the GAN model to enhance the quality and authenticity of generated signals further. Additionally, the adaptation of this model to different types of wireless communication systems and the exploration of its scalability and robustness in varying network conditions will be crucial. Future studies should also delve into developing more nuanced evaluation metrics for wireless signals, like assessing the quality of power spectral density (PSD) and characteristics specific to each waveform. Moreover, considering the practicability of GAN-based evasion attacks against CNN-based detection methods, subsequent studies must explore defense mechanisms to identify and counteract these attacks. This exploration will not only refine the technology but also address potential security concerns and ethical implications, ensuring responsible use in secure wireless communication systems.

The continuous evolution of machine learning models and the increasing complexity of wireless communication systems present a fertile ground for expanding the applications of GANs in this field.

References

[1] K. Davaslioglu and Y. E. Sagduyu, “Generative adversarial learning for spectrum sensing,” IEEE International Conference on Communications (ICC), Kansas City, MO, May 20–24, 2018.

[2] T. O’Shea, J. Corgan, and C. Clancy, “Convolutional radio modulation recognition networks,” International Conference on Engineering Appli- cations of Neural Networks, Aberdeen, United Kingdom, Sept. 2–5, 2016.

[3] 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.

[4] 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.

[5] Tewelgn Kebede, Yihenew Wondie, Johannes Steinbrunn, Hailu Belay Kassa, Kevin T. Kornegay, ” Multi-Carrier Waveforms and Multiple Access Strategies in Wireless Networks: Performance, Applications, and Challenges ”, https://ieeexplore.ieee.org/document/9713895.

[6] ” What is Multicarrier Modulation, https://www.electronics-notes.com/articles/radio/multicarrier-modulation/basics-techniques.php.

[7] ” Understanding I/Q Signals and Quadrature Modulation, https://www.allaboutcircuits.com/textbook/radio-frequency-analysis-design/radio-frequency-demodulation/understanding-i-q-signals-and-quadrature-modulation/.