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

Scientific Project

Group : 20

Ecole Centrale Casablanca

January 19, 2024

Outline

Research Context

Figure: Generated Images using Generative Adeversarial Networks

Research Context

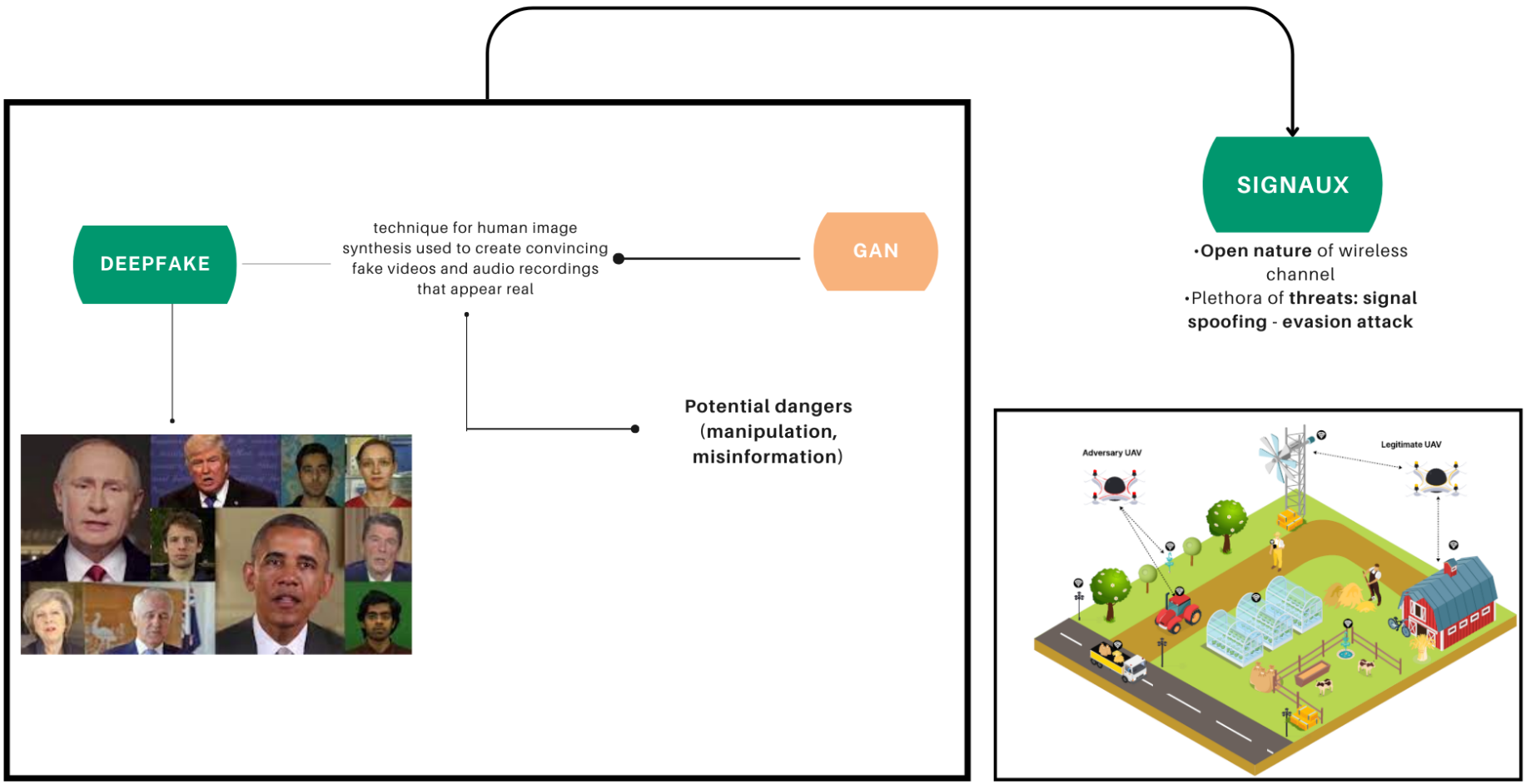


Figure: Drawing parallels between the deceptive capabilities of deepfakes in media and the misleading nature of signal spoofing in communications.

Problematic

- Limited Application to Single Carrier Signals and Complexity of Filtered Multicarrier Systems: In wireless communications, most GAN research has concentrated on single carrier signals.
- Our intention is to create an evasion attack using **Multicarrier Waveforms**, where an adversary attempts to fool a machine learning algorithm into making a wrong decision

Methodology

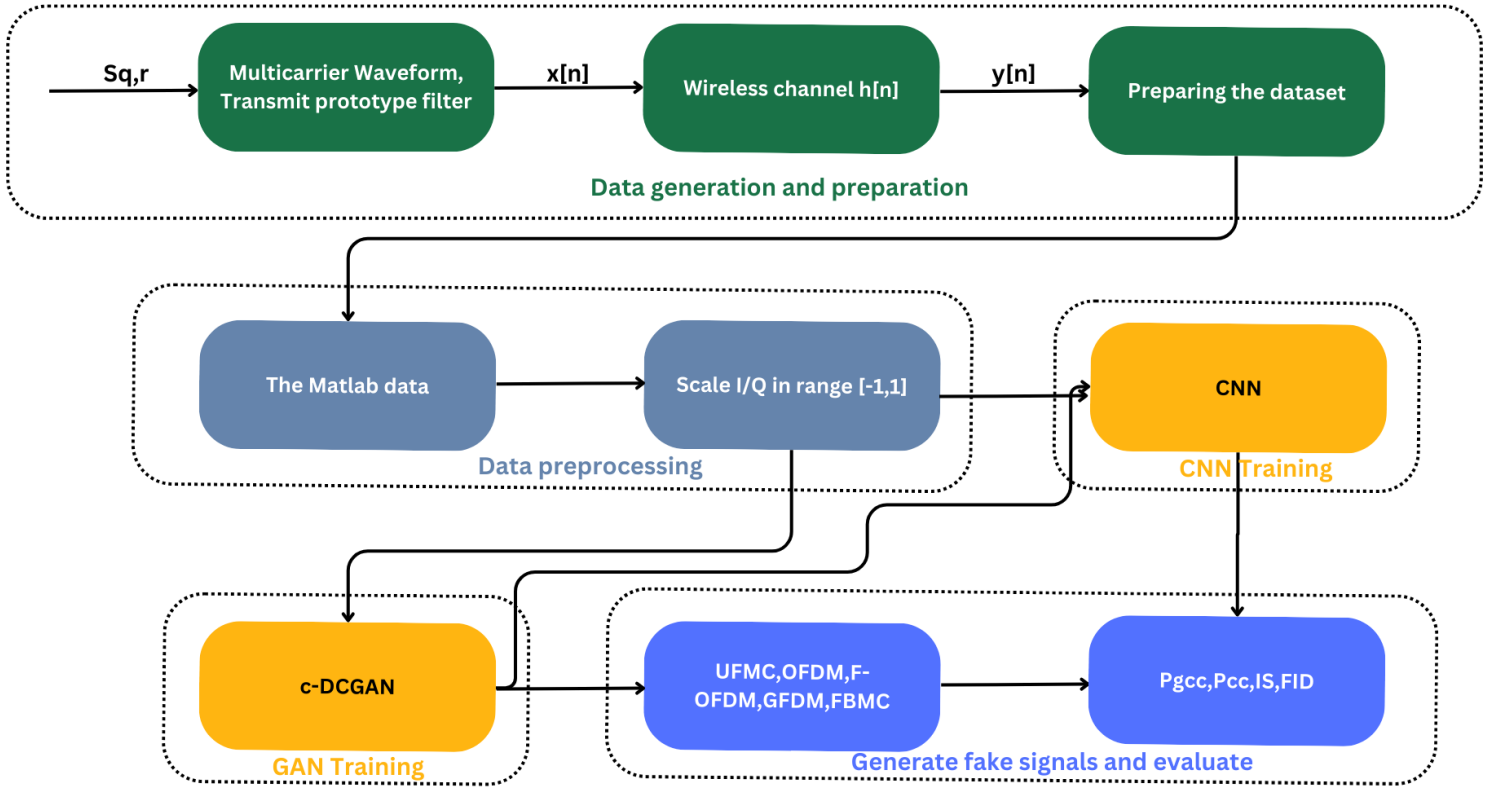
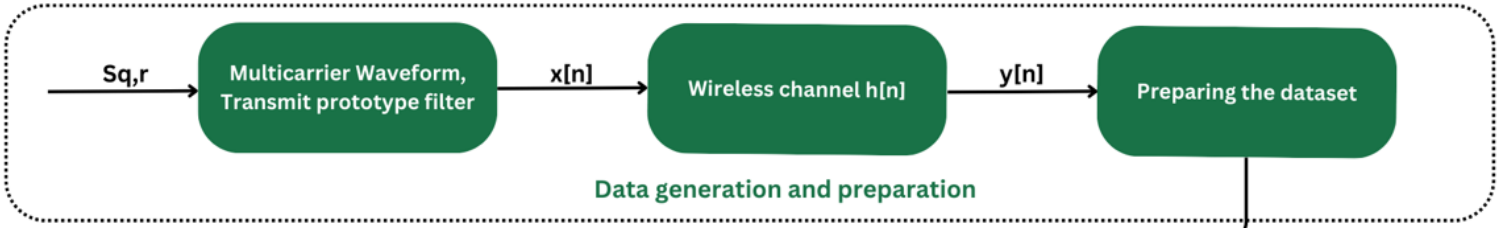
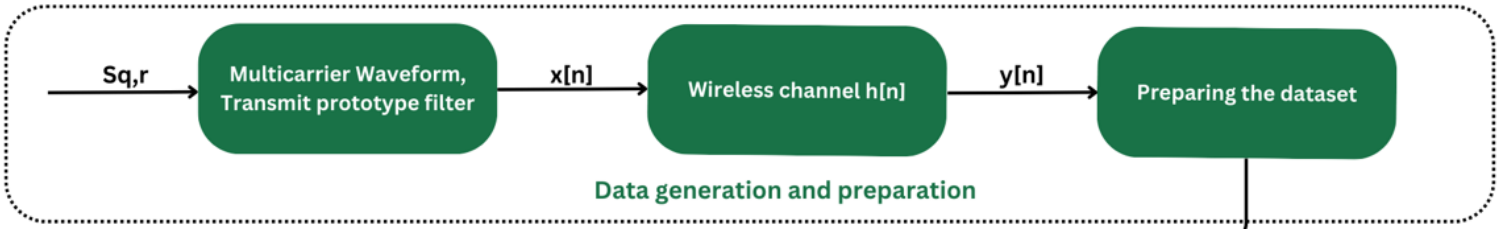


Figure: Process of our experience



Multicarrier Waveforms

- Orthogonal Frequency Division Multiplexing (OFDM) : renowned for its subcarrier orthogonality.
- Filter Bank Multicarrier (FBMC): FBMC uses sophisticated filtering for each subcarrier.
- Generalized Frequency Division Multiplexing (GFDM): introducing sub-symbols and circular filtering
- Universal Filtered Multicarrier (UFMC): uses individually tailored filters for each subcarrier
- Filtered-OFDM (F-OFDM): variant of OFDM that adresses some of its limitations



Mutlicarrier Modulation

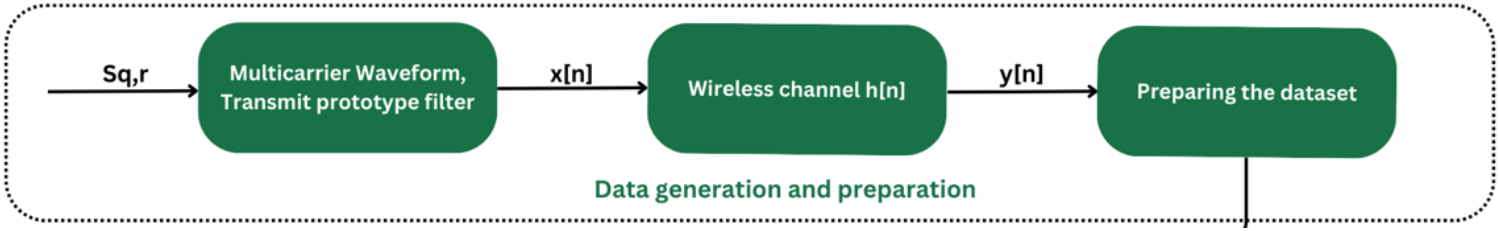
- The transmitted signal in a multicarrier setup can be mathematically formulated as follows:

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

- The synthesis function gq,r[n], a crucial component in a Gabor System, is articulated as:

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

Signal Theory

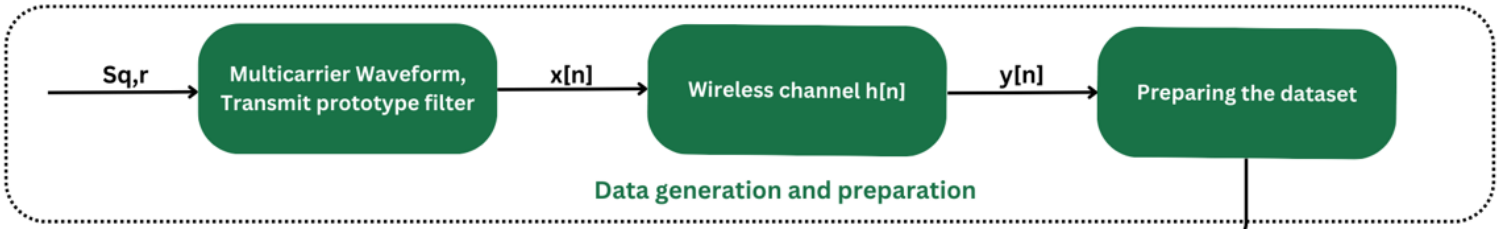


- Received signal:

$$\boxed{r_{\text{ch}}[n] = x[n] * h[n]} \tag{3}$$

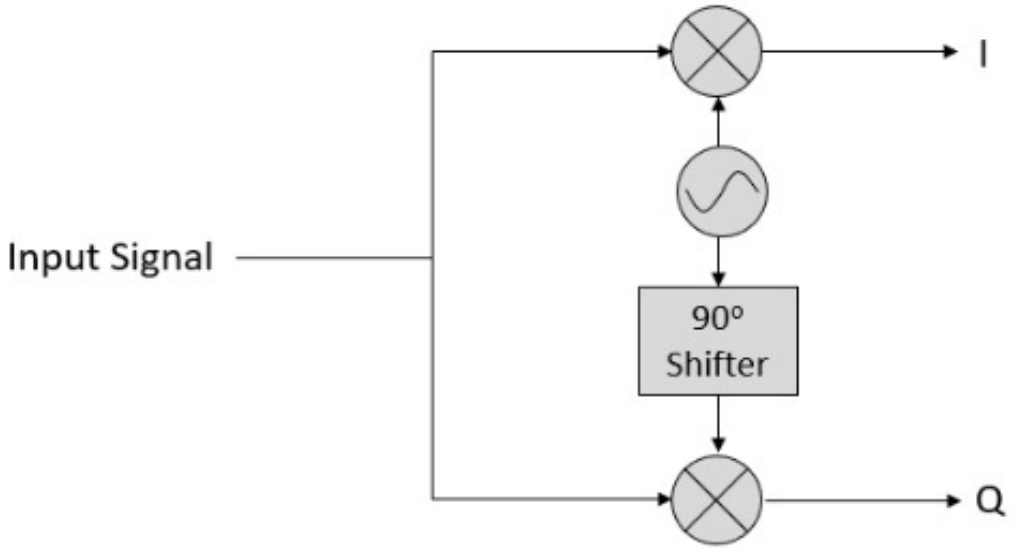
- Final received signal:

$$\boxed{y[n] = e^{j\frac{2\pi n\kappa}{N_c}} \cdot r_{\text{ch}}[n+\delta] + z[n]} \tag{4}$$

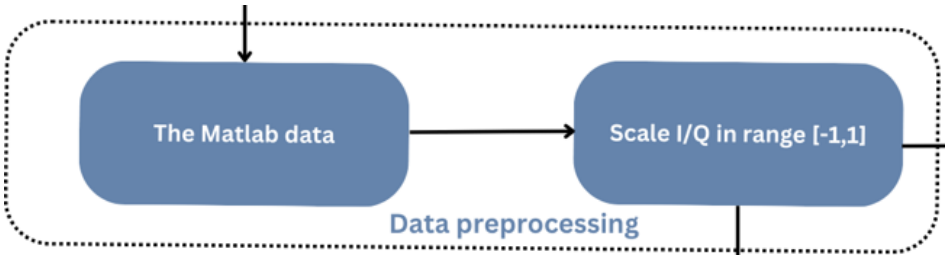


I/Q encoding

I/Q encoding is a method that separately represents a signal's in-phase (I) and quadrature (Q) components, helping to use the available frequency spectrum more efficiently.



Dataset



A 5 dimensional Dataset:

- 1st Dim : 2500 samples per class per SNR
- 2nd Dim : I components (128)
- 3rd Dim : Q components (128)
- 4th Dim : 5 Waveform types
- 5th Dim : 3 SNR(Signal-to-Noise Ratio) values

GAN and c-GAN architecture

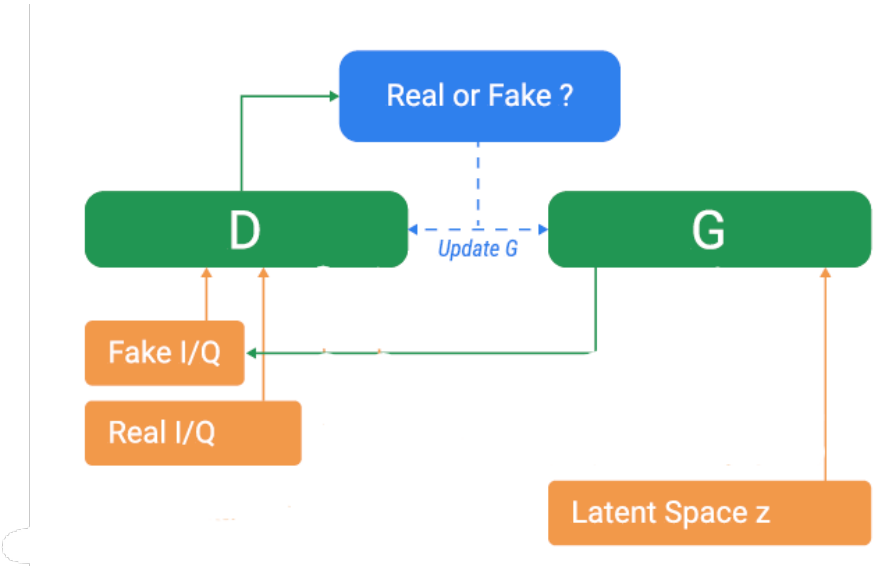


Figure: GAN architecture for multicarrier waveforms

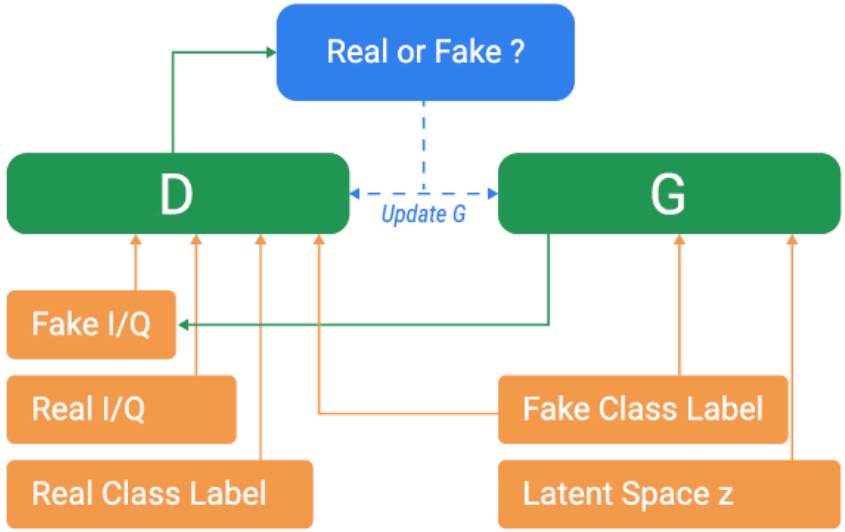


Figure: Class label conditional GAN architecture for multicarrier waveforms

Maths behind GAN

- The discriminator aims to learn the conditional probability $P(Y|X = x)$,where X is the input and Y is the output.
- The generator aims to learn the joint probability $P(X, Y)$.
- The discriminator is a binary classifier that distinguishes between real data (X) and generated data $(\hat{X})$.
- This is the Value function : *(proof 1 in annexe)*

$$\boxed{V(G, D) = \mathbb{E}_{x \sim P_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z)))]} \quad (5)$$

where x is sampled from the real data distribution P_{data}, and z is sampled from the input noise distribution P_z of the generator. The discriminator aims to maximize it while the generator aims to minimize it.

- The **minimax game** to achieve a balance.

- **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 θ_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 θ_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, $P_G = P_{\textbf{data}}$ *(proof in annexe).*

GAN and CNN baseline models

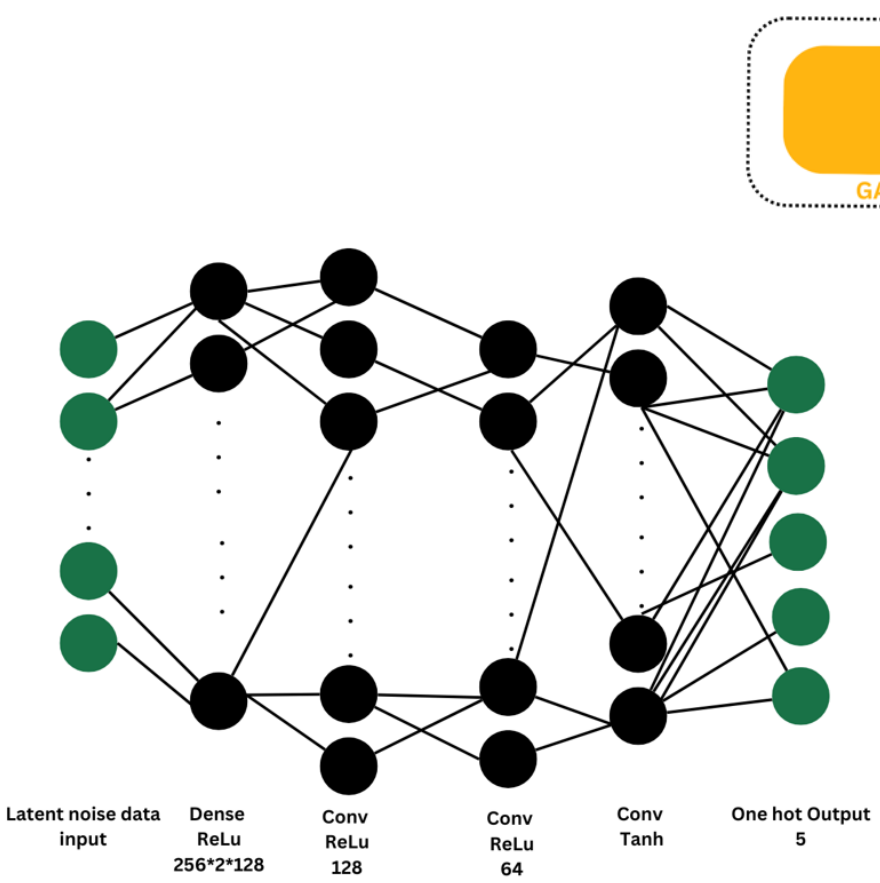


Figure: Generator's model

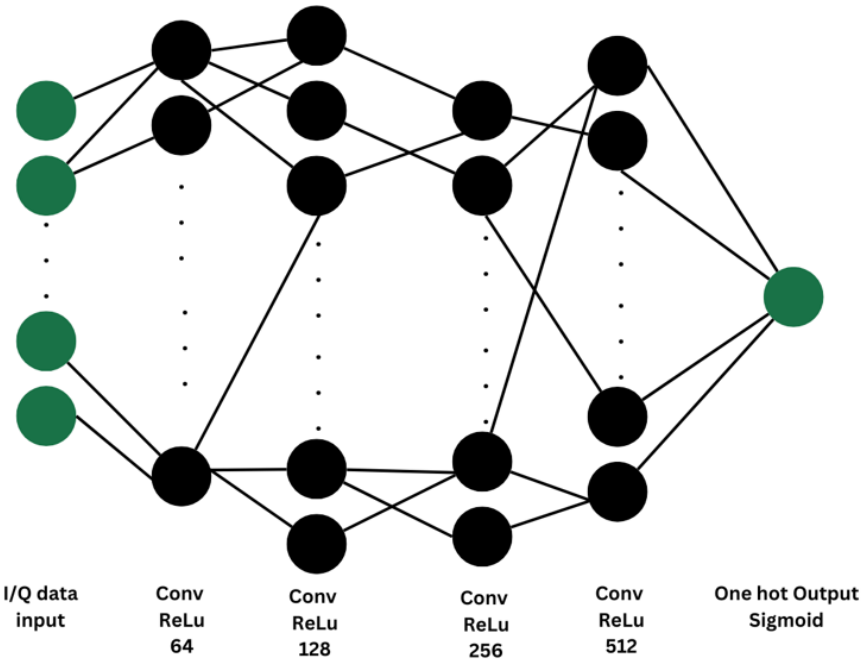


Figure: Discriminator's model

- Discriminator : DR 0.3 , Adam Optimizer.
- Generator : Embedding layer of size 50.

GAN and CNN baseline models

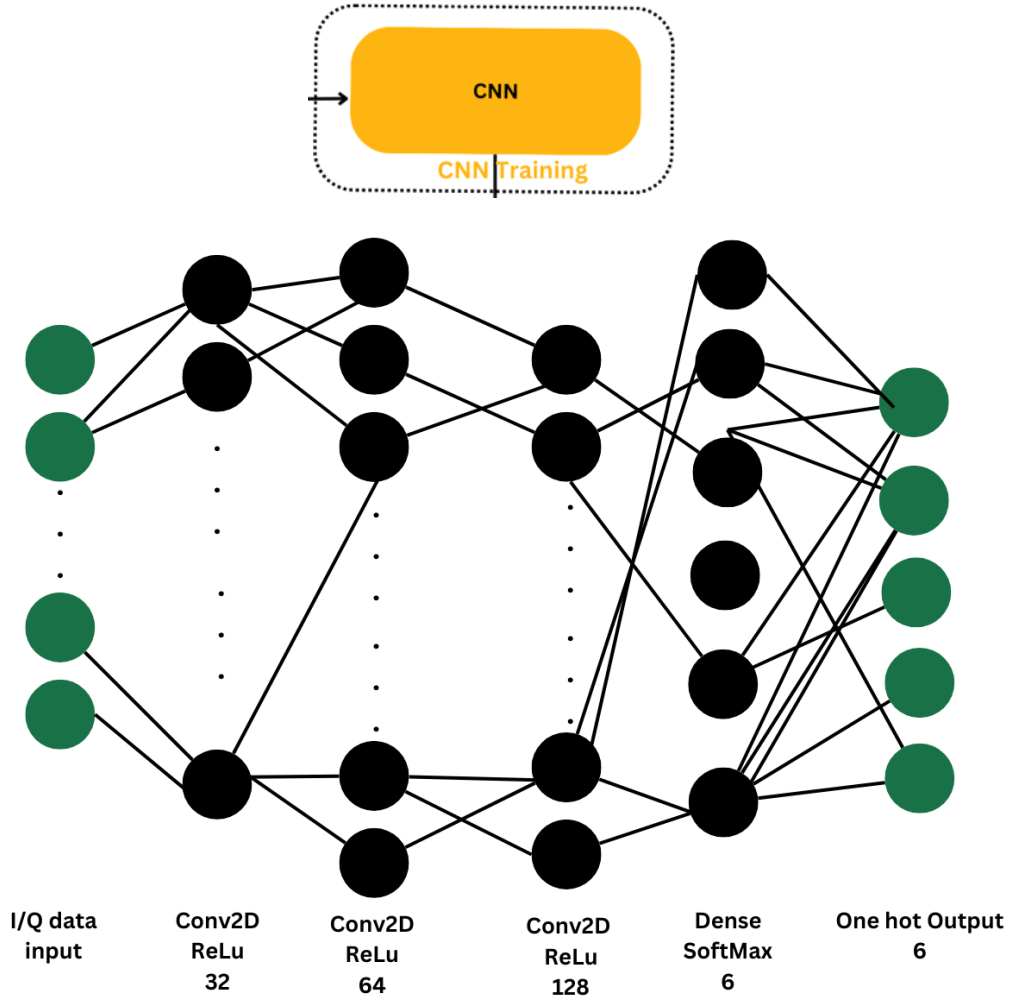
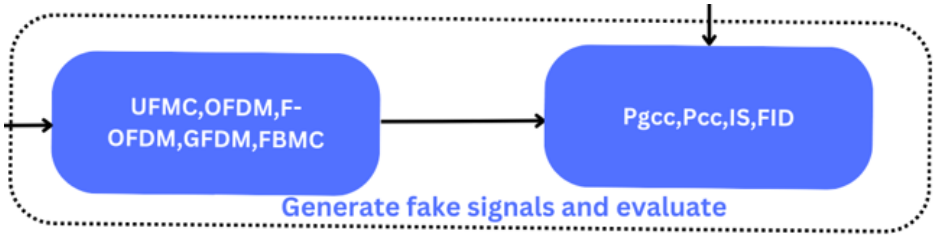


Figure: CNN Classifier's model

- CNN: DR 0.3 , Early Stopping, Adam Opt, 80% train and 20% test.

Results and Discussion



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

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

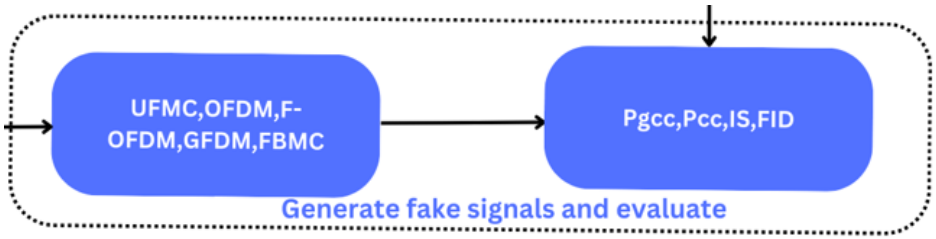
N_w : the total count of waveforms utilized for training the GAN. N_f : the number of fake signals generated per waveform. N_{cc} : the cumulative count of correct classifications across all waveforms.

- The Per-Class Correct Classification Probability (P_{ccc}), defined as:

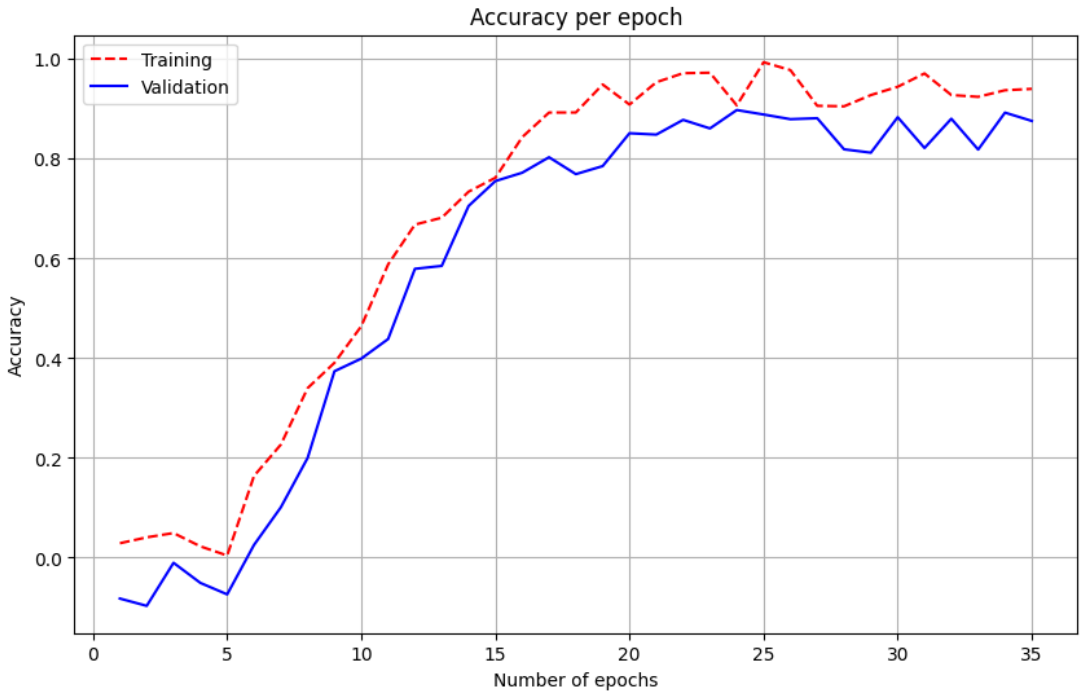
$$P_{ccc} = \frac{1}{N_f} \sum_{i=1}^{N_f} p_{c_i} \tag{7}$$

p_{c_i} denotes the per-class probabilities reported by the CNN classifier.

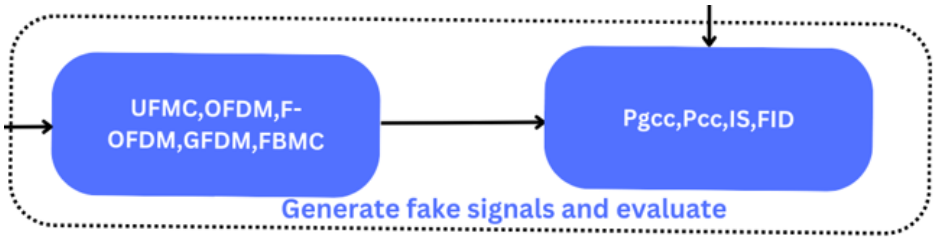
Results and Discussion



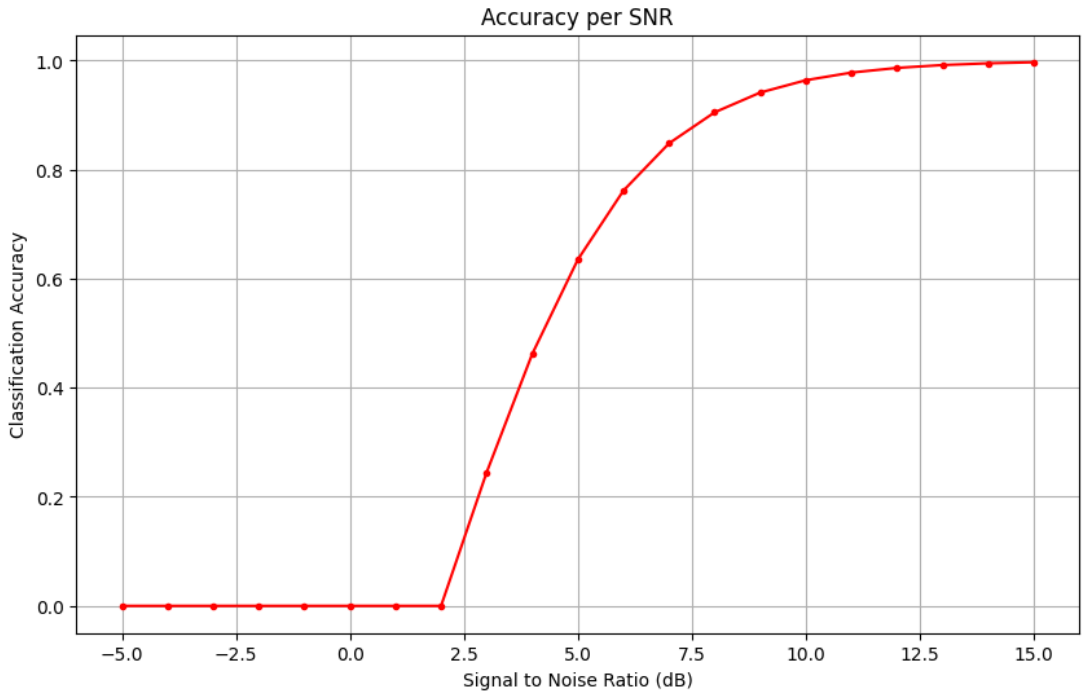
- Performance of the CNN Classifier



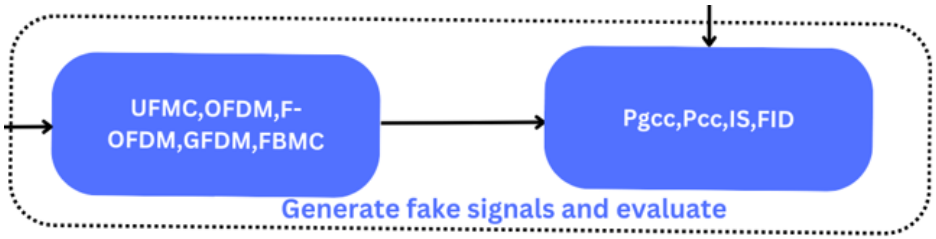
Results and Discussion



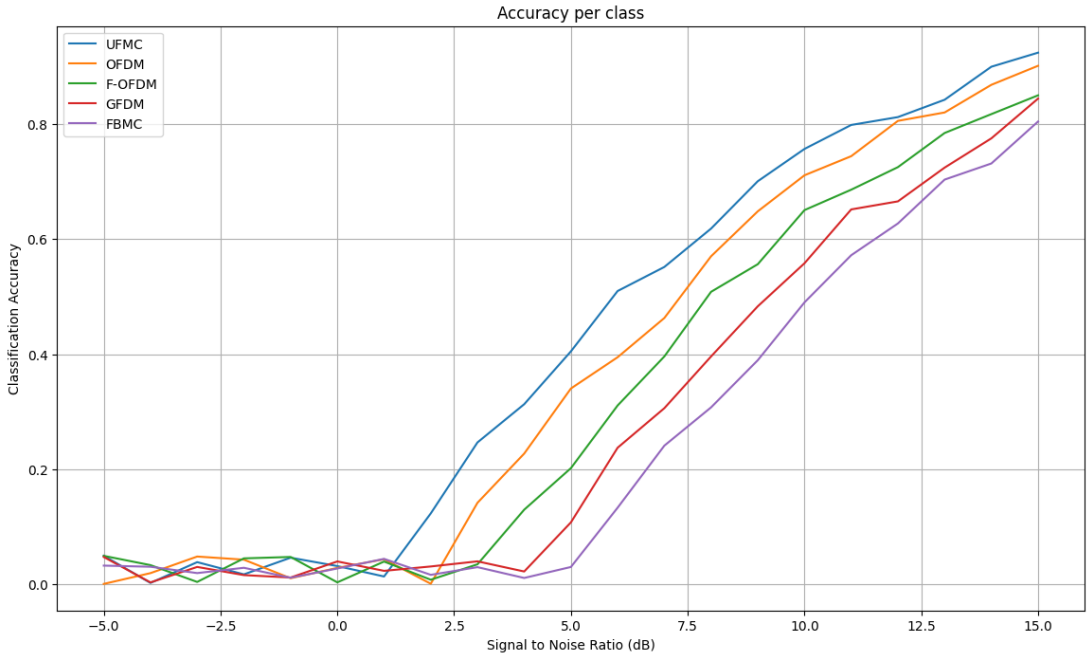
- Performance of the CNN Classifier



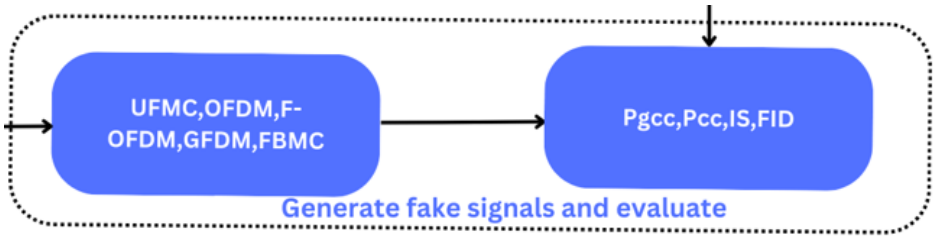
Results and Discussion



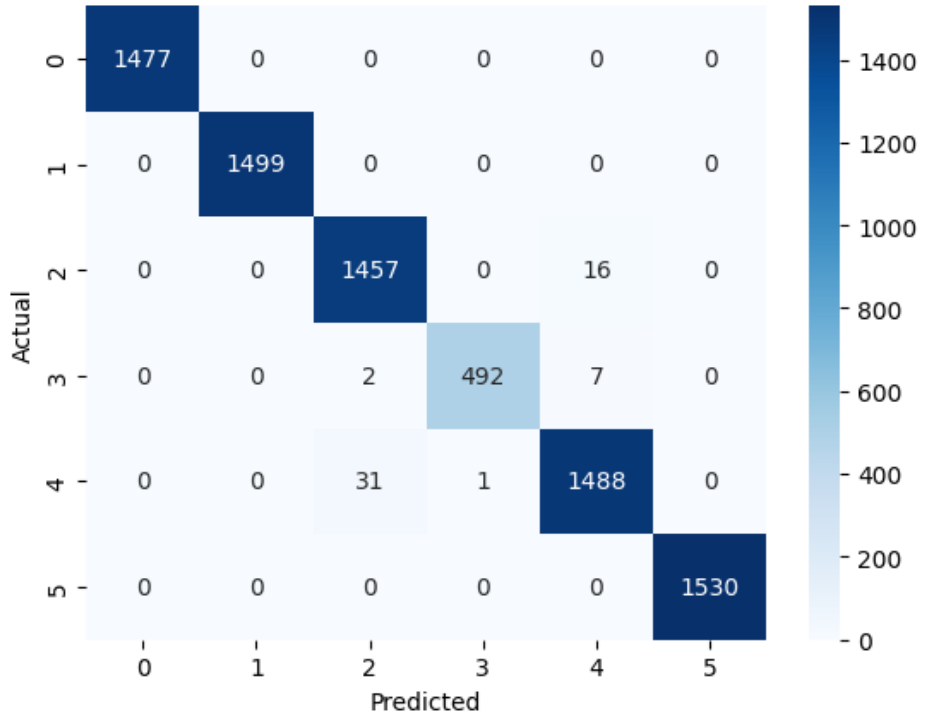
- Performance of the CNN Classifier



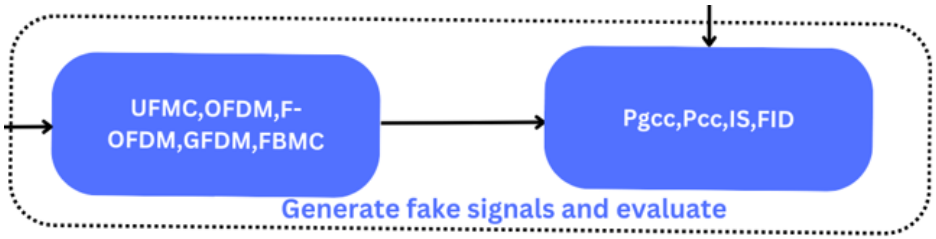
Results and Discussion



- Performance of the CNN Classifier



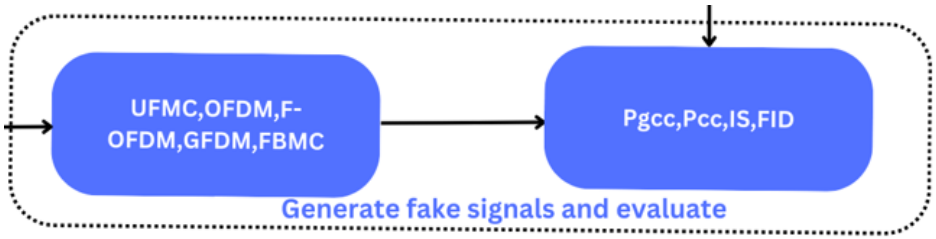
Results and Discussion



- Performance of the GAN , Batch size = 1280

	Epoch	Pgcc	Pccc_UFMC	Pccc_OFDM	Pccc_F-OFDM	Pccc_GFDM	Pccc_FBMC
0	20	0.61	0.67	0.023	0.65	0.50	0.75
1	40	0.62	0.73	0.035	0.77	0.55	0.78
2	60	0.66	0.74	0.021	0.89	0.57	0.80
3	80	0.67	0.60	0.037	0.92	0.70	0.82
4	100	0.71	0.74	0.070	0.95	0.73	0.85

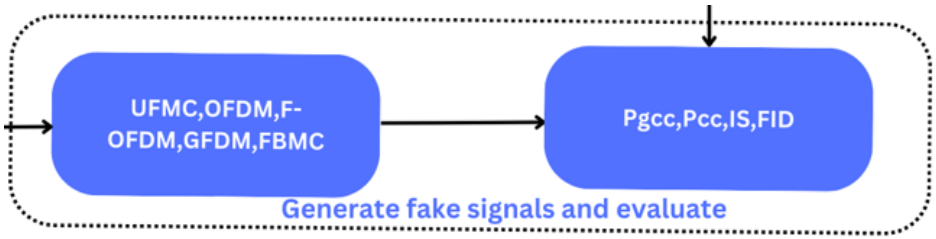
Results and Discussion



- Performance of the GAN , Batch size = 896

	Epoch	Pgcc	Pccc_UFMC	Pccc_OFDM	Pccc_F-OFDM	Pccc_GFDM	Pccc_FBMC
0	20	0.66	0.67	0.013	0.65	0.61	0.65
1	40	0.72	0.73	0.025	0.77	0.65	0.74
2	60	0.71	0.84	0.031	0.79	0.57	0.82
3	80	0.73	0.90	0.027	0.72	0.60	0.87
4	100	0.71	0.94	0.032	0.75	0.73	0.89

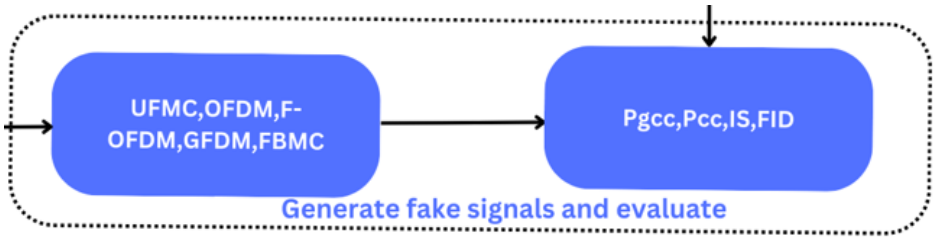
Results and Discussion



- Performance of the GAN , Batch size = 512

	Epoch	Pgcc	Pccc_UFMC	Pccc_OFDM	Pccc_F-OFDM	Pccc_GFDM	Pccc_FBMC
0	20	0.61	0.77	0.023	0.61	0.51	0.620
1	40	0.69	0.71	0.125	0.71	0.55	0.740
2	60	0.71	0.82	0.240	0.79	0.57	0.760
3	80	0.73	0.87	0.270	0.82	0.50	0.840
4	100	0.78	0.91	0.320	0.95	0.53	0.892

Results and Discussion



- Performance of the GAN , Batch size = 256

	Epoch	Pgcc	Pccc_UFMC	Pccc_OFDM	Pccc_F-OFDM	Pccc_GFDM	Pccc_FBMC
0	20	0.51	0.67	0.213	0.62	0.010	0.58
1	40	0.59	0.71	0.215	0.67	0.012	0.64
2	60	0.59	0.74	0.340	0.77	0.030	0.66
3	80	0.63	0.81	0.370	0.86	0.010	0.71
4	100	0.68	0.89	0.420	0.93	0.010	0.72

Results and Discussion

- 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 generator is not able to maximize both GFDM and OFDM at the same time.
- The OFDM and GFDM waveforms are the most complex among the investigated ones, indeed, in addition to the per subcarrier filtering, GFDM introduces also filtering per subsymbols.

Attack type	**Success probability**
Random signals	5%
GAN-based (Batch size 512)	78%

Table: Success Probability of evasion attacks

Limitations and perspectives

- Time constraint: which restricted the number of epochs to 100.
- The inability to run the Inception Score (IS) and Frechet Inception Distance (FID) metrics due to their difficult adaptability to signals as opposed to images.
- Parameters variation: number of layers, number of training samples, ...

Thank You

Thank you for your attention.

Bibiliograpghy

[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 Applications 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.

Bibliography

[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/.

Annexe

Proof 1 : 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.

Annexe

Proof 2 : Convergence at Optimality

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.

Annexe

The minimum value of V is related to JS:

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

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