Adversarial Machine Learning Security Problems for 6G: mmWave Beam Prediction Use-Case

Evren Catak
Norwegian University of Science and Technology
Gjøvik, Norway
evren.catak@ntnu.no

Ferhat Ozgur Catak
Simula Research Lab.
Fornebu, Norway
ozgur@simula.no

Arild Moldsvor
Norwegian University of Science and Technology
Gjøvik, Norway
arild.moldsvor@ntnu.no

***Abstract*—6G is the next generation for the communication systems. In recent years, machine learning algorithms have been applied widely in various fields such as health, transportation, and the autonomous car. The predictive algorithms will be used in 6G problems. With the rapid developments of deep learning techniques, it is critical to take the security concern into account when applying the algorithms. While machine learning offers significant advantages for 6G, AI models' security is normally ignored. Due to the many applications in the real world, security is a vital part of the algorithms. This paper proposes a mitigation method for adversarial attacks against proposed 6G machine learning models for the millimeter-wave (mmWave) beam prediction using adversarial learning. The main idea behind adversarial attacks against machine learning models is to produce faulty results by manipulating trained deep learning models for 6G applications for mmWave beam prediction. We also present the adversarial learning mitigation method's performance for 6G security in millimeter-wave beam prediction application with fast gradient sign method attack. The mean square errors of the defended model under attack are very close to the undefended model without attack.**

***Index Terms*—machine learning, AI, millimeter-wave, beamforming, adversarial machine learning**

I. Introduction

In the past 20 years, most of the physical layer technologies, i.e., modulations, multiple access waveforms, coding techniques and time/frequency multiplexing, have flourished over the evolution of cellular systems. However, up to 4G, time-frequency domain technologies have been explored to increase overall system capacity [1]. The recent developments in 5G and beyond technologies support emerging applications such as smart homes, vehicular networks, augmented reality (AR), virtual reality (VR) with unprecedented rates enabled by recent advances in massive multiple-input multiple-output (MIMO), millimeter-wave (mmWave) communications, network slicing, small cells, and Internet of things (IoT). These complex structures of 5G and beyond technologies can be captured by using data-driven approach machine learning (ML) algorithms [2]. The strong learning, reasoning and intelligent recognition abilities of ML allow the network structure to train and adapt itself to support the diverse demands of the systems without human intervention [3].

The extraordinary growth of data traffic on wireless communication has driven the need to examine the highest frequency spectrum to meet the requirements by using mmWave communications [4]. The frequency range of mmWaves is between 30 and 300 GHz, i.e. an available bandwidth of about 250 GHz. Enabling mmWave communication faces mainly three critical challenges [5] i) the sensitivity for atmospheric attenuation obligates it to propagate solely by line-of-sight paths, ii) hand over problem between base stations (BSs), iii) adjustment of the large numbers of beamforming arrays. In addition, due to the use of large antenna arrays and low complexity, transceiver demands are captured by using ML algorithms for mmWave communication.

mmWave communication systems require the pointing of the narrow beams. The goal is to choose the best beams for the analogue beamforming with both receiver and transmitter having multi-antenna arrays. A beam codeword is a set of analogue phase-shifted values applied to the antenna elements forming an analogue beam. In [6], deep learning base beam selection is proposed for exploiting channel state information for the sub-6 GHz links. In addition to beam prediction, information about the locations and sizes of vehicles in the communication environment are used in [7] to predict the optimal beam pair. Locational based beamforming solutions are more suitable for line-of-sight (LOS) communication. The same locations for the non-line-of-sight (NLOS) transmission need different beamforming solutions.

The integration of the ML for the 6G and beyond technologies lead to potential security concerns. Especially, wireless communication systems have security vulnerabilities due to their nature. The studies of 6G and beyond technologies with ML methods should be evaluated in terms of security. Current research is mainly building the ML models for the 6G communication problems, and security concerns are mostly ignored in previous studies. Alkhateeb et al. [5] proposed a feed-forward deep learning model for RF beamforming codeword prediction with several base stations (BSs) with multiple users. The BSs beamforming vectors are predicted from the received signals using the omni and quasi-omni beam patterns to enable both LOS and NLOS transmissions. While the proposed method in [5] showed promising results for the beamforming problem, the security of the deep learning algorithm itself was not investigated. Based on the shortcomings of the literature's security concepts, we deal with the security problem of ML application for beamforming prediction. More specifically, in this study, we focus on adversarial attack strategies based on loss maximisation-based attacks against proposed ML models for 6G mmWave communication. We consider the adversarial ML attacks to poisoning the beamforming prediction model [5]. Thus our main contributions for this paper are as follows:

- We show that an undefended RF beamforming codeword deep learning model's prediction performance will decrease with the craftily designed adversarial noise.
- We demonstrated that the adversarial training based robustness approach is one of the mitigation methods for this domain.

The rest of the paper is organized as follows: Section II describes background information about beamforming and deep learning algorithms. Section III shows our adversarial model and Section IV evaluates the proposed model according to defined research questions. Finally, Section V concludes this paper.

Notations

In this paper, we employ the following notations:

- Vectors are denoted in lowercase bold font and matrices in uppercase bold font i.e., **a** and **A** respectively.
- For a given vector **a**, $\mathbf{a}_{i,j}$ and $\mathbf{a}_k$ denote the (i, j)-th component of **a** and k-th component of **a** respectively.

II. Background Information

A. Downlink Transmission

Consider a mmWave communication system as in Figure 1 where N is the number of the BSs with equipped M antennas, serving for one mobile user who has a single antenna. All the BSs are connected with a cloud processing unit. The transmitted signal $\mathbf{s} = [s_1, s_2, \dots, s_K]$ with K subcarriers is firstly precoded by using code vector $\mathbf{c}_k = [c_{k,1}, c_{k,2}, \dots, c_{k,N}]^T$ and then transformed into time domain using K-point IFFT operation. Thus, the baseband signal from the n-th BS and k-th subcarrier is

$$\mathbf{x}_{k,n} = \mathbf{f}_n c_{k,n} s_k \tag{1}$$

where $\mathbf{f}_n$ is the beam steering vector defined for each BS antennas as $[\mathbf{f}_n]_m = \frac{1}{\sqrt{M}} e^{j\theta_{n,m}}$ where $\theta_{n,m}$ is a quantized angle. RF precoding matrix $\mathbf{F}^R F = blkdiag(\mathbf{c}_1, \mathbf{c}_{2\dots,\mathbf{c}_N}) \in \mathbf{C}^{NM \times N}$ The received signal at the k-th subcarrier is expressed as

$$\mathbf{y}_k = \sum_{n=1}^{N} \mathbf{h}_{k,n}^T \mathbf{x}_{k,n} + v_k \tag{2}$$

where v_k is additive white Gaussian noise (AWGN) with variance σ^2, i.e., $N(0, \sigma^2)$.

B. Effective Achievable Rate

Perfect channel information satisfies optimum achievable rate, however, the channel state information requires large training overhead due to the large number of antennas. On the other hand, the channel information and beamforming vector need to be updated as the user moves. These issues can be captured with the channel coherence time T_C and channel beam coherence time T_B, respectively examined in detail in [8]. The multi-path channel and beams stay aligned on the T_C and T_B duration respectively. The channel training and beamforming design take place in the first T_{tr}, the rest of it is used to the data transmission. To develop a model with efficient channel training and beamforming design, the effective achievable rate needs to be maximized. The final problem formulations [5] are

$$\begin{aligned}
&\prod \left(T_{tr}, \{\mathbf{c}_k\}_{k=1}^{K}, \mathbf{F}^R, \mathcal{F}\right) = \\
&\text{argmax} \left(1 - \frac{T_{TR}}{T_B}\right) \sum_{k=1}^{K} \log_2 \left(1 + SNR|\sum_{n=1}^{N} \mathbf{h}_{k,n}^T \mathbf{f}_n c_{k,n}|^2\right) \\
&\quad s.t.\ \mathbf{f}_n \in \mathcal{F},\ \forall n \\
&\quad \|c_k\|^2 = 1 \ \ \forall k
\end{aligned} \tag{3}$$

where $\mathcal{F}$ is the quantized codebook for the BSs RF beamforming vectors. Solving these equations determine a solution for a low channel training ahead and realize the beamforming vector to satisfy the maximum achievable rate, R.

C. Using Deep Learning Algorithms to estimate RF beamforming vectors

Using the benefits of ML algorithms gives a novel solution for a massive amount of MIMO channel training and scanning a large number of narrow beams. The beams depend on the environmental conditions like user and BSs locations, furniture, trees, buildings e.t.c. It is too difficult to define all these environment conditions as a closed-form equation. A good alternative is to use omni and quasi-omni beam patterns to predict the best RF beamforming vectors. Using these beam patterns benefits to take into account the reflection and diffraction of the pilot signal. The deep learning solution consists of two states: training and prediction. Firstly, the deep learning model learns the beams according to the omni-received pilots. Secondly, the model uses the trained data to predict the RF beamforming vector for the current condition.

1) Training Steps: The user sends uplink training pilot sequences for each beam coherence time T_B. BSs combine received pilot sequences on RF beamforming vector and feed them to the cloud. The cloud uses the received sequences from all the BSs as the input of the deep learning algorithm to find the achievable rate in (4) for every RF beamforming vector to represent the desired outputs, where $\mathbf{g}_p$ is the channel coefficient for omni beams.

$$R_n^{(p)} = \frac{1}{K} \sum_{n=1}^{N} \log_2 \left(1 + SNR|\mathbf{h}_{k,n}^T \mathbf{g}_p|^2\right) \tag{4}$$

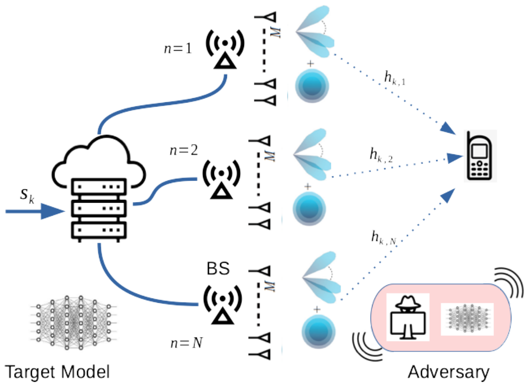

Fig. 1: Block diagram of the mmWave beamforming system.

2) Learning Steps: In this stage, the trained deep learning model is used to predict the RF beamforming vectors. Firstly, the user sends an uplink pilot sequence. The BSs combine these sequences and send them to the cloud. Then, the cloud uses the trained deep learning model to predict the best RF beamforming vectors to maximize the achievable rate for each BS. Finally, BSs use the predicted RF beamforming vectors to estimate the effective channel $\mathbf{h}_{k,n}$.

To sum up, ML algorithms find diverse applications in a wireless communication system where we consider the RF beamforming vector prediction [5]. On the other hand, security concerns in wireless communication are also a problem for the ML algorithm. In the following subsection, we will briefly describe adversarial ML, attack environments, and adversarial training that we have used in this study.

D. Adversarial Machine Learning

Adversarial machine learning is an attack technique that attempts to fool neural network models by supplying craftily manipulated input with a small difference [9]. Attackers apply model evasion attacks for phishing attacks, spams, and executing malware code in an analysis environment [10]. There are also some advantages to attackers in misclassification and misdirection of models. In such attacks, the attacker does not change training instances. Instead, he tries to make some small perturbations in input instances in the model's inference time to make this new input instance seem safe (i.e. normal behaviour) [11]. We mainly concentrate on this kind of adversarial attacks in this study. There are many attacking methods for deep learning models, and the Fast-Gradient Sign Method (FGSM) is the most straightforward and powerful attack type. We only focus on the FGSM attack, but our solution to prevent this attack can be applied to other adversarial machine learning attacks. FGSM works by utilizing the gradients of the neural network to create an adversarial example to evade the model. For an input instance $\mathbf{x}$, the FGSM utilizes the gradients ∇_x of the loss value ℓ for the input instance to build a new instance $\mathbf{x}^{adv}$ that maximizes the loss value of the classifier hypothesis h. This new instance is named the adversarial instance. We can summarize the FGSM using the following explanation:

$$\mathbf{x}^{adv} = \mathbf{x} + \epsilon \cdot sign(\nabla_x \ell(\theta, \mathbf{x}, y)) \tag{5}$$

By adding a slowly modest noise vector $\eta \in \mathbb{R}^n$ whose elements are equal to the sign of the features of the gradient of the cost function ℓ for the input $\mathbf{x} \in \mathbb{R}^n$, the attacker can easily manipulate the output of a deep learning model. Figure 2 shows the details of the FGSM attack.

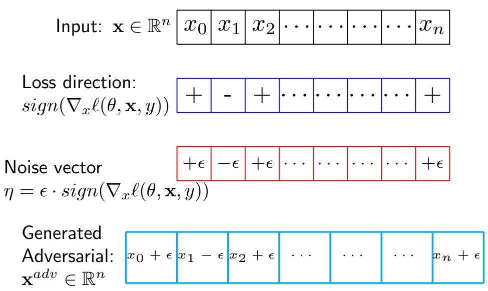
Input: $\mathbf{x} \in \mathbb{R}^n$
x_0 x_1 x_2 $\cdots$ $\cdots$ $\cdots$ $\cdots$ x_n
Loss direction:
$sign(\nabla_x \ell(\theta, \mathbf{x}, y))$
+ - + $\cdots$ $\cdots$ $\cdots$ $\cdots$ +
Noise vector
$\eta = \epsilon \cdot sign(\nabla_x \ell(\theta, \mathbf{x}, y))$
$+\epsilon$ $-\epsilon$ $+\epsilon$ $\cdots$ $\cdots$ $\cdots$ $\cdots$ $+\epsilon$
Generated
Adversarial:
$\mathbf{x}^{adv} \in \mathbb{R}^n$
$x_0 + \epsilon$ $x_1 - \epsilon$ $x_2 + \epsilon$ $\cdots$ $\cdots$ $\cdots$ $x_n + \epsilon$

Fig. 2: FGSM attack steps. The input vector $\mathbf{x} \in \mathbb{R}^n$ is poisoned with loss maximization direction.

E. Adversarial Training

Adversarial training is a widely recommended defense technique that implies generating adversarial instances using the gradient of the victim classifier, and then re-training the model with the adversarial instances and their respective labels. This technique has demonstrated to be efficient in defending models from adversarial attacks.

Let us first think of a common classification problem with training instances $X \in \mathbb{R}^{m \times n}$ of dimension d, and a label space Y. We assume the classifier h_θ has been trained to minimize a loss function ℓ as follows:

$$\underset{\theta}{min} \frac{1}{m} \sum_{i=1}^{m} \ell(h_\theta(\mathbf{x}_i, y_i)) \tag{6}$$

Given a classifier model $h_\theta(\cdot)$ and an input instance x with a responding output y, then an adversarial instance x^{adv} is an input such that:

$$h_\theta(x^{adv}) \neq y \quad \wedge \; d(x, x^{adv}) < \epsilon \tag{7}$$

where $d(\cdot,\cdot)$ is the distance metric between two input instances, the original input x and the adversarial version x^{adv}. Most actual adversarial model attacks transform Equation (7) into the following optimization problem:

$$\underset{x}{arg\,max}\, \ell\left(h_\theta(x^{adv}), y\right) \tag{8}$$

$$s.t. \quad d(x, x^{adv}) < \epsilon \tag{9}$$

where ℓ is the loss function between predicted output $h(\cdot)$ and correct label y. In order to mitigate such attacks, at per training step, the conventional training procedure from Equation 6 is replaced with a `min-max` objective function to minimize the expected value of the maximum loss, as follows:

$$\underset{\theta}{min} \underset{(x,y)}{\mathbb{E}} \left(\underset{d(x,x^{adv})<\epsilon}{max} \ell(h(x^{adv}), y) \right) \tag{10}$$

III. System Model

A. Adversarial Training

Figure 3 shows the adversarial training process. After the model is trained, adversarial inputs are created using the model itself, combined with legitimate users information and added to the training. When the model reaches the steady-state state, the training process is completed. In this way, the model will both predict RF beamforming codeword for legitimate users while at the same time being immune to the craftily designed noise attack that will be added as input.

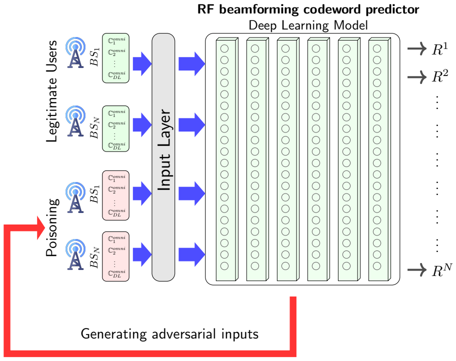

Fig. 3: The diagram of RF beamforming codeword adversarial training.

B. Capability of the Attacker

We assumed that the attacker's primary purpose is to manipulate the RF model by applying carefully crafted noise to the input data. In a real-world scenario, this white-box setting is the most desired choice for an attacker that does not take the risks of being caught in a trap. The problem is that it requires the attacker to access the model from outside to generate adversarial examples. After manipulating the input data, the attacker can exploit the RF beamforming codeword prediction model's vulnerabilities in the same manner as in an adversary's sandbox environment. The prediction model predicts the adversarial instances when the attacker can convert some model's outputs to other outputs (i.e. wrong prediction).

However, to prevent this noise addition from being easily noticed, the attacker must answer an optimization problem to determine which regions in the input data (i.e. beamforming) that must be modified. By solving this optimization problem using one of the available attack methods [10], the attacker aims to reduce the prediction performance on the manipulated data as much as possible. In this study, to limit the maximum allowed perturbation allowed for the attacker, we used l_∞ norm, which is the maximum difference limit between original and adversarial instances. Figure 4 shows the attack scenario. The attacker gets an legitimate input, $\mathbf{x}$, creates a noise vector with an ϵ budget $\eta = \epsilon \cdot sign(\nabla_x \ell(\theta, \mathbf{x}, y))$, sums the input instance and the craftily designed noise to create adversarial input $\mathbf{x}^{adv} = \mathbf{x} + \eta$.

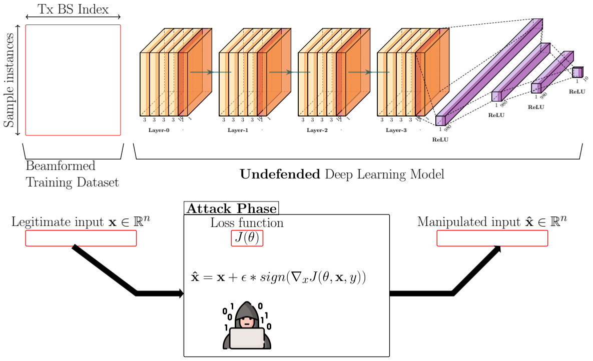
Tx BS Index
Sample instances
Beamformed
Training Dataset
Layer-0
Layer-1
Layer-2
Layer-3
3
1
10
980
ReLU
Undefended Deep Learning Model
Attack Phase
Loss function
$J(\theta)$
Legitimate input $\mathbf{x} \in \mathbb{R}^n$
Manipulated input $\hat{\mathbf{x}} \in \mathbb{R}^n$
$\hat{\mathbf{x}} = \mathbf{x} + \epsilon * sign(\nabla_x J(\theta, \mathbf{x}, y))$
0 1 0 0 1 1 1 0

Fig. 4: RF Beamforming manipulation process.

IV. Experiments

In the experiments, we tested three different scenarios

- **SC1:** Undefended beamforming codework prediction model without any adversary
- **SC2:** Undefended beamforming codework prediction model with FGSM attack
- **SC3:** Adversarial trained beamforming codework prediction model with FGSM attack

The experiments were performed using the Python scripts and ML libraries: Keras, Tensorflow, and Scikit-learn, on the following machine: 2.8 GHz Quad-Core Intel Core i7 with 16GB of RAM. For all scenarios, two models, undefended and adversarial trained, were built to obtain prediction results. In the first model, the model is trained without any input poisoning. The first model (i.e. undefended model) was used with legitimate users (for SC1) and adversaries (for SC2). The second model (i.e. the adversarially trained model) was used under the FGSM attack. The hyper-parameters such as the number of hidden layers and the number of neurons in the hidden layers, the activation function, the loss function, and the optimization method are the same for both models.

The model architectures are given in Table I and the hyper-parameters selected in Table II.

TABLE I: Model architecture

Layer type	Layer information
Fully Connected + ReLU	100
Fully Connected + ReLU	100
Fully Connected + ReLU	100
Fully Connected + TanH	1

TABLE II: Milimater-wave beam prediction model parameters

Parameter	Value
Optimizer	Adam
Learning rate	0.01
Batch Size	100
Dropout Ratio	0.25
Epochs	20

A. Research Questions

We consider the following two research questions (RQs):

- **RQ1**: Is the deep learning based RF beamforming codeword predictor vulnerable for adversarial machine learning attacks?
- **RQ2**: Is the iterative adversarial training approach a mitigation method for the adversarial attacks in beamforming prediction?

B. RF Beamforming Data Generator

We employed the generic deep learning dataset for millimeter-wave and massive MIMO applications (DeepMIMO) data generator in our experiments [12]. Figure 5 shows the bird's-eye view of a section of the *O1' ray-tracing scenario*, showing the two streets' intersection.

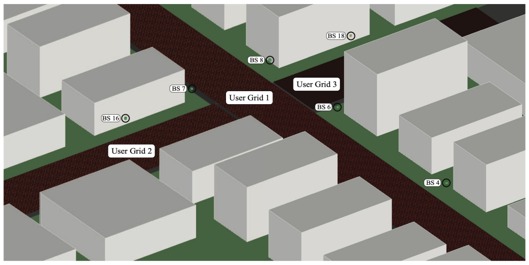

Fig. 5: Original scenario [12].

In this section, we conduct experiments on the mmWave communication and massive MIMO applications dataset from the publicly available data set repository. We implemented the proposed mitigation method using Keras and TensorFlow libraries in the Python environment.

C. Results for RQ1

Figure 6 shows the original undefended deep learning model results without any attack. According to the figure, the deep learning model's predictions are very close the original value. Figure 7 shows the training history of the beamforming prediction model with 35.000 training instances. The model is trained with clean (non-perturbated) instances. Figure 8 shows the performance results of the beamforming prediction model's evaluation results under the FGSM attack. We have used l_∞ norm as the distance metric, which shows the maximum allowable perturbation amount for each item in the input vector $\mathbf{x}$. According to the figures, the undefended RF beamforming codeword prediction model is vulnerable for the FGSM attack. The MSE performance result of the model under attack is approximately 40 (i.e. $\frac{0.00843(Normal)}{0.00021(Attacked)} \approx 40.14$) times higher.

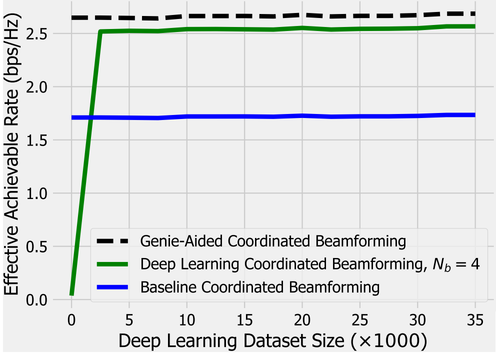

Fig. 6: Original (Undefended) RF beamforming codeword deep learning model results.

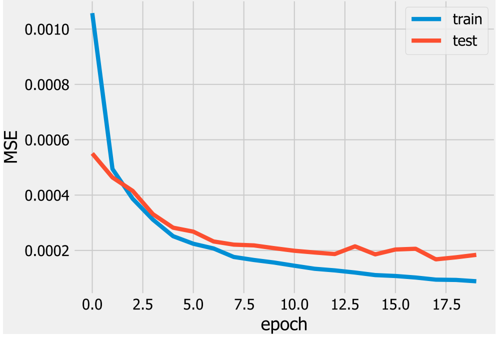

Fig. 7: The beamforming prediction model history.

D. Results for RQ2

Adversarial training is a popularly advised defense mechanism that proposes generating adversarial instances using the victim model's loss function, and then re-training the model with the newly generated adversarial instances and their respective outputs. This approach has proved to be effective in protecting deep learning models from adversarial machine learning attacks. Figure 9 shows the adversarial trained deep learning model results with FGSM attack. According to the figure, the deep learning model's predictions are very close to the original (i.e. undefended and non-attacked) value in Figure 6. Figure 10 shows the MSE of the performance results for all scenarios.

E. Threats to Validity

A key *external validity* threat is related to the generalization of results [13]. We used only the RF beamforming dataset in our experiments, and we need more case studies to generalize the results. Moreover, the dataset reflects different types of milimeter-wave beams.

Our key *construct validity* threat is related to the selection of attack type FGSM. Nevertheless, note that this attack is from the literature [13] and applied to several deep learning usage domains. In the future, we will conduct dedicated

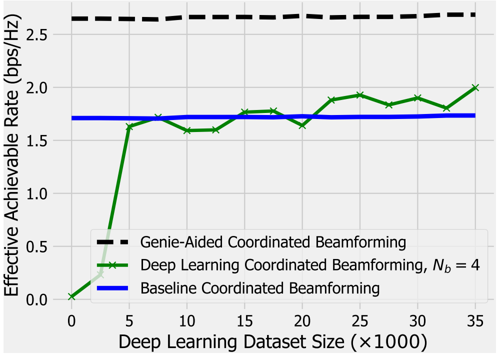

Fig. 8: Attacked (Undefended) RF beamforming codeword deep learning model results.

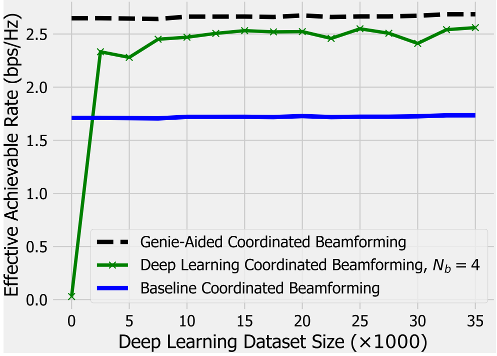

Fig. 9: Attacked (Undefended) RF beamforming codeword deep learning model results with adversarial training.

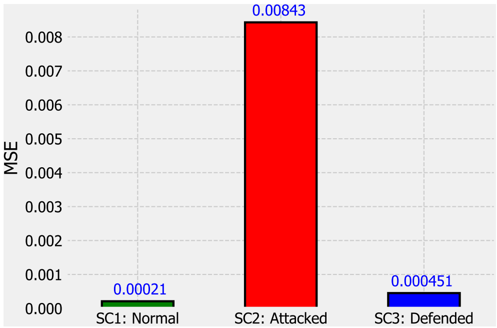

Fig. 10: The performance results for all scenarios.

empirical studies to investigate more adversarial machine learning attacks systematically.

Our main *conclusion validity* threat is due to finding the best attack budget ϵ that is responsible for manipulating the legitimate user's signal for poisoning the beamforming prediction model. To mitigate this threat, we repeated each experiment 20 times to reduce the probability that the results were obtained by chance. In a standard neural network training, all weights are initialized uniformly at random. In the second stage, using optimization, these weights are updated to fit the classification problem. Since the training started with a probabilistic approach, there is a possibility of facing optimization's local minimum problem. To eliminate the local minimum problem, we repeat the training 20 times to find the ϵ value that gives the best attack result. In each repetition, the weights were initialized uniformly at random but with different values. If the optimization function failed to find the global minimum in the next experiment, it is likely to see it as the weights have been initialized with different values.

V. Conclusions and Future Works

This research discussed one of the security issues related to RF beamforming codeword prediction models' vulnerabilities and solutions: (1) Is the deep learning-based RF beamforming codeword predictor vulnerable for adversarial machine learning attacks? (2) Is the iterative adversarial training approach a mitigation method for the adversarial attacks in beamforming prediction? We conducted experiments with the DeepMIMO's *O1' ray-tracing scenario* scenario to answer these questions. Our results confirm that the original model is vulnerable to a modified FGSM attack. One of the mitigation methods is the iterative adversarial training approach. Our empirical results also show that iterative adversarial training successfully increases the RF beamforming prediction performance and creates a more accurate predictor, suggesting that the strategy can improve the predictor's performance.

References

[1] H. Viswanathan and P. E. Mogensen, "Communications in the 6G era," *IEEE Access*, vol. 8, pp. 57063–57074, 2020.

[2] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.

[3] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks," *IEEE Access*, vol. 6, pp. 32328–32338, 2018.

[4] W. Roh, J. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 106–113, 2014.

[5] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37328–37348, 2018.

[6] M. S. Sim, Y. Lim, S. H. Park, L. Dai, and C. Chae, "Deep learning-based mmwave beam selection for 5G NR/6G with sub-6 GHz channel information: Algorithms and prototype validation," *IEEE Access*, vol. 8, pp. 51634–51646, 2020.

[7] Y. Wang, A. Klautau, M. Ribero, M. Narasimha, and R. W. Heath, "Mmwave vehicular beam training with situational awareness by machine learning," in *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2018.

[8] V. Va, J. Choi, and R. W. Heath, "The impact of beamwidth on temporal channel variation in vehicular channels and its implications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5014–5029, 2017.

[9] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," *arXiv e-prints*, p. arXiv:1611.01236, Nov. 2016.

[10] M. Aladag, F. O. Catak, and E. Gul, "Preventing data poisoning attacks by using generative models," in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pp. 1–5, 2019.

[11] O. Faruk Tuna, F. Ozgur Catak, and M. Taner Eskil, "Exploiting epistemic uncertainty of the deep learning models to generate adversarial samples," *arXiv e-prints*, p. arXiv:2102.04150, Feb. 2021.

[12] A. Alkhateeb, “DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications,” 2019.

[13] P. Runeson, M. Höst, R. Austen, and B. Regnell, *Case Study Research in Software Engineering – Guidelines and Examples*. United States: John Wiley and Sons Inc., 2012.