Exploring Practical Vulnerabilities of Machine Learning-based Wireless Systems

Zikun Liu, Changming Xu, and Emerson Sie, *University of Illinois Urbana-Champaign;* Gagandeep Singh, *University of Illinois Urbana-Champaign and VMware Research;* Deepak Vasisht, *University of Illinois Urbana-Champaign*

https://www.usenix.org/conference/nsdi23/presentation/liu-zikun

This paper is included in the
Proceedings of the 20th USENIX Symposium on
Networked Systems Design and Implementation.

April 17–19, 2023 • Boston, MA, USA

978-1-939133-33-5

Exploring Practical Vulnerabilities of Machine Learning-based Wireless Systems

Zikun Liu[†], Changming Xu[†], Emerson Sie[†], Gagandeep Singh[†‡], Deepak Vasisht[†]
[†] *University of Illinois Urbana-Champaign,* [‡] *VMware Research*

Abstract

Machine Learning (ML) is an increasingly popular tool for designing wireless systems, both for communication and sensing applications. We design and evaluate the impact of practically feasible adversarial attacks against such ML-based wireless systems. In doing so, we solve challenges that are unique to the wireless domain: lack of synchronization between a benign device and the adversarial device, and the effects of the wireless channel on adversarial noise. We build, RAFA (RAdio Frequency Attack), the first hardware-implemented adversarial attack platform against ML-based wireless systems and evaluate it against two state-of-the-art communication and sensing approaches at the physical layer. Our results show that both these systems experience a significant performance drop in response to the adversarial attack.

1 Introduction

Next-generation networks, 5G and beyond, promise to be unprecedented in their scale and the diversity of applications, ranging from virtual reality to low power Internet of Things applications. Machine Learning (ML) has emerged as a key component of such future networks to deliver application-specific performance goals by optimally managing the diverse capabilities of these networks – multiple antennas, different spectrum bands, and smart surfaces. In academia, researchers have efficaciously applied ML for both communication [45, 55, 60, 79, 85] and sensing [6, 51, 59, 74, 94] applications. ML-based techniques are increasingly making their way to the industry, in both RAN (radio access network) and the network core. This trend has been accelerated by the recent shift of telcos to cloud-based execution models.

Our goal: We investigate the vulnerabilities of using ML in wireless systems. Our investigation is motivated by two reasons. First, wireless networks play a crucial role in many human-critical applications like autonomous driving, smart healthcare, factory control, etc. Any failure to meet network performance goals can have severe consequences in such settings. Second, in popular domains such as computer vision and natural language processing, past work has shown that an adversary can add small imperceptible noise to the inputs of a neural network making it predict completely different results [27, 76] (e.g., a turtle is classified as a gun). Several of these attacks have been reproduced in the real-world on state-of-the-art ML models in these domains [5, 48, 49, 71], showing that despite their impressive performance, the ML models are not robust. These practical attacks have promoted the development of new techniques for formal verification [40, 72, 73] and robust training [11, 34, 81, 84, 87, 90] in the vision and NLP domains.

Our goal is to explore the *practical* vulnerabilities of state-of-the-art ML-based wireless systems using adversarial attacks. To mount practical real-world adversarial attacks, an adversary must meet three requirements. First, it must not need access to the infrastructure in real-time, i.e., it cannot coordinate its transmissions with a benign sender, or access the signal sensed by a benign receiver. Second, it must be low complexity, i.e., it must not require large antenna arrays. Finally, it must be low power. It is relatively straightforward to jam the spectrum with blind high-power transmissions. However, jamming causes large-scale disruption to the spectrum and causes spectrum owners (e.g., telecom operators) to react. We are interested in small changes of the signal that specifically target the ML models in wireless systems, and expose their vulnerabilities.

Past work [2, 7, 9, 17, 18, 23, 43, 68] has studied adversarial attacks against ML-based wireless systems in simulation. These attacks do not meet the requirements above. Specifically, these attacks make unrealistic assumptions about the attacker capabilities. For example, they assume that an adversary can perfectly transmit adversarial signal or the attacker can directly manipulate the input matrix to the neural network. In practice, these assumptions do not hold. Adversarial signal undergoes wireless transformations described below before it arrives at a receiver. Similarly, directly altering the input matrix to the neural network requires access to the receiver.

Challenges: Consider the scenario shown in Fig. 1, where a (multi-antenna) base station communicates with a client device and uses ML-based models to deliver communication or sensing services. The adversary introduces small amounts of noise in the environment. Generating real-world adversarial attacks in such scenarios is challenging because of the underlying physics of wireless signal propagation. A typical adversarial attack takes an input to the ML model and crafts a noise vector specific to this input. This structured noise, when added to the input, causes the model to predict an incorrect output. In the wireless systems context, an attacker does not know the wireless channel between the client and the base station, and therefore does not know the signal being fed to the ML model. Secondly, the noise vector transmitted by the attacker is vastly different from what gets observed at the base station because: (a) Propagation effects: As the noise travels from the attacker to the end device, the noise vector undergoes the wireless channel experiencing reflection, attenuation, and phase shifts in the environment. (b) Clock offsets: the clock of

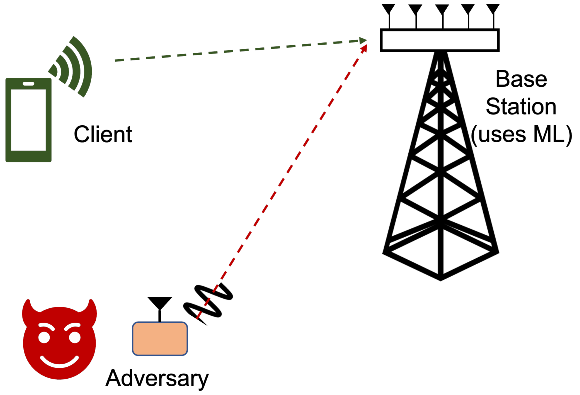


Figure 1: *A multi-antenna base station uses ML-based methods to deliver communication or sensing services to the client. The adversary transmits small amounts of noise and disrupts these services.*

the adversary is not synchronized with the end device, leading to random time and frequency offsets.

RAFA: We build the first real-world hardware-implemented adversarial attack platform, RAFA, that solves these challenges and targets ML-based wireless systems. Our system operates using a single antenna software defined radio and does not need real time access to the client or the base station in the attack setup. RAFA senses an ongoing communication on the wireless medium and introduces small amounts of noise (perturbation) to the medium that disrupts state-of-the-art ML-models. We demonstrate this attack on two state-of-the-art systems at the physical layer: one communication system (FIRE [55]) and one localization system (DLoc [6]). We show the effectiveness of RAFA in both white-box and black-box settings. The design of RAFA solves the following challenges:

Unknown Inputs: Adversarial algorithms [16, 27, 57] typically identify a perturbation that changes the output of a neural network *for a given input*. However, in practice, a wireless adversary, like RAFA, does not know the input to the ML model because: (a) the wireless channel from the client to the base station is unknown to the adversary, and (b) once the signal is received at the base station, the signal undergoes transformations (e.g., correcting for carrier frequency offsets) before it is fed to the ML model. Therefore, to model practical attacks, we require generating input-agnostic adversarial perturbations. We, first, design a universal adversarial perturbation (UAP), that focuses on changing the output on a distribution of inputs, rather than a single input. This allows us to be robust to the distribution of wireless signals, implying that we do not need to know the channel from the client to the base station. Furthermore, we develop differential versions of the pre-processing steps so that the adversary can generate perturbation vectors that remain adversarial even on pre-processed data points.

Lack of Synchronization: The adversary is not synchronized with the client or the base station. Therefore, its transmission is *not* aligned with the client in time or frequency. The lack of time synchronization creates temporal misalignment between the benign signal and the adversarial perturbation. Similarly, the lack of frequency synchronization creates a time-varying phase shift between them. Furthermore, such clock offsets are random and hard to predict beforehand. To counter such offsets, we create a robustness mechanism in our UAP design, that tests the perturbation vector for arbitrary phase offsets, and picks perturbation vectors that are robust to such offsets. By doing so, we shift the burden of dealing with the clock offsets from hardware to software, therefore simplifying our hardware design for the attack.

Channel-induced Transformation: Finally, the perturbation vector crafted by RAFA undergoes a channel transformation as it travels to the base station. The channel transformation changes both the amplitude and phase of the perturbation vector. Therefore, there are no guarantees on the value of the perturbation vector after the channel transformation. This means we cannot design a perturbation vector that is robust to these transforms. RAFA leverages reciprocity to counter this challenge. Specifically, the base station occasionally transmits beacons or responses to its legitimate clients. Our adversary overhears these transmissions and uses it to estimate the channel from the base station to itself. Due to the reciprocity principle, this channel is equal to the channel from the adversary to the base station. Once we know this channel, we use our robust UAP method to construct a perturbation vector that is effective even after the channel transform.

While these factors serve as natural protectors for wireless systems against adversarial attacks, RAFA demonstrates the ability to mount effective adversarial attacks despite these challenges. We have implemented RAFA using the USRP software defined radio against two state-of-the-art ML-based wireless systems: FIRE [55] (for MIMO communication), and DLoc [6] (for ML-based localization). For FIRE, RAFA's adversarial attack can reduce the median SNR (from original SNR of 17.8 dB) of the predicted channel by 4.1 dB on average compared to just 2.1 dB drop for Gaussian baseline. Similarly, for DLoc, RAFA increases the median localization error (from original error of 1.04 m) by 71 cm on average compared to just 2 cm increase for Gaussian baseline. Our results also present a preliminary version of potential defense strategies.

Contributions: Our main contributions are:

- We design a new robust adversarial attack against ML-based wireless systems that is input-agnostic and models real-world effects such as lack of synchronization.
- We leverage channel reciprocity to model the effect of wireless channel on adversarial perturbations.
- We demonstrate the first hardware-implemented adversarial attacks against ML-based wireless systems.

ML-based wireless systems are increasingly being proposed in academia [6, 10, 35, 55, 94], and actively being explored in the industry [39, 64, 65] . Therefore, it is timely and important to explore the challenges posed by adversarial attacks in this context. To the best of our knowledge, our work is the first to demonstrate realistic hardware-implemented attacks against ML-based wireless systems. We believe RAFA will allow researchers and practitioners to test the practical robustness of ML systems before they get deployed widely in the real-world and have severe consequences for any failures when exposed to such attacks. We also envision that adversarial examples exposed by RAFA will lead to development of robust ML models. An early attempt at developing such robust models is demonstrated in Sec. 6.6.

2 Adversary Model

Objective: Our goal is to promote the development of robust ML-models by identifying the attack surface of ML-based wireless systems in the real world. We focus on the existence and performance of *practically feasible* wireless attacks. We identify *practically feasible* as attacks that can be implemented using real hardware and without requiring coordination with the base station or client. Furthermore, we are interested in vulnerabilities specific to ML-models in the wireless setting. Therefore, we do not consider jamming, which is a brute-force solution that blocks all communication in the medium. We consider the scenario in Fig. 1. A client communicates with a base station on the wireless medium. The base station can have multiple antennas. The base station relies on a machine learning based approach to deliver communication or localization services to the client. Some examples for ML-based communication systems are shown in Tab. 1.

Application	Examples
Communication	FIRE [55], OptML [10], NeuMac [35]
Localization	DLoc [6], IPS [93] , LAFA [38]

Table 1: *Examples of ML-based Wireless Systems*

Adversary Goal: The adversary wants to degrade the quality of ML-based location or communication service offered by the base station. The adversary aims to target specific ML-based services, and not jam the entire spectrum. To achieve this objective, the adversary transmits a carefully designed perturbation signal over the wireless channel. This perturbation gets superimposed at the receiver (which could be a cellular base station, access point, etc) with the benign signal transmitted from the client such as a cell phone. The receiver will later feed this seemingly intact but actually compromised signal into the ML-pipeline, negatively affecting its output prediction. Consistent with recent trends, we focus on neural networks as target ML models for this paper.

We describe the attacker properties in our threat model:

Coordination-free: We do not assume any coordination between the base station and the adversary (or between the client and adversary). This implies that the adversary is unsynchronized, i.e. has time and frequency shifts with respect to the other (benign) devices. The adversary also does not know when the transmission from the client begins or ends.

Base Station Information: The adversary does not know the location of the client or the base station. The adversary knows only public information about base station hardware, such as information which can be gleaned from FCC filings or standards documents.

Low-complexity: The adversary uses low complexity hardware. Even though the base station and the client may have multiple antennas, the adversary uses a single antenna transmitter. This reduces the cost and complexity of the attack, making it more universal, and generalizable.

Knowledge about the ML model: We assume that the adversary can sample data from the training distribution of the ML model running on the base station and knows the model family (e.g.,variational autoencoder) but not necessarily the architecture (e.g., fully-connected, convolutional), and the operations involved in the pre-processing pipeline. We believe that these assumptions are feasible for the real-world ML-based wireless systems because: (1) details about the model family are disclosed and accessible, (2) the attacker can access sample data simply by overhearing the client transmission, and measuring the corresponding wireless signals, (3) pre-processing pipelines (e.g., correcting for channel frequency offsets) are fairly standardized. We consider both white-box (access to model architecture and parameters) and black-box adversaries (no access to model architecture and parameters). Our results show that black-box adversaries are almost as effective as white-box ones.

Noise Budget: To avoid large-scale disruption to the wireless spectrum, we require that the L_∞-norm of the noise vector crafted by the adversary is bounded by a small constant $\varepsilon \in \mathbb{R}$. This prevents the noise being concentrated in individual sub-frequencies. We show the effect of the noise crafted with different values of ε on the model performance in Sec. 6.

Test Time Attack: We do not consider attacks that interfere with model training, the model is trained and fixed for our attacks. The adversary transmits a noise signal at test time.

3 System Overview

3.1 Target Systems

We consider two state-of-the-art ML-based wireless systems – one each for communication and sensing. In this paper, we focus solely on physical layer systems, while delegating investigation of attacks on higher layers to future work.

A. FIRE: Reciprocity for FDD MIMO systems – In order to achieve MIMO capabilities in 5G, base stations need to know the downlink wireless channel from their antennas to every client device. In FDD (Frequency Domain Duplexing)

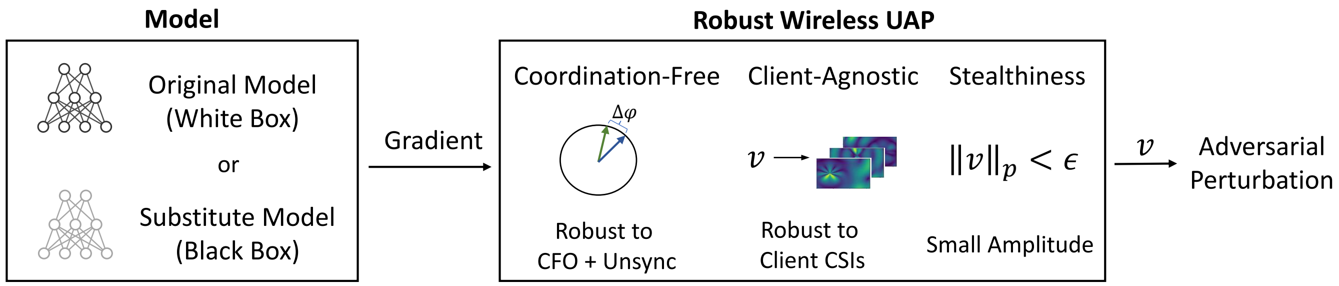


Figure 2: *RAFA Pipeline: RAFA works in both modes – white-box and black-box (with a substitute model). RAFA is robust to multiple wireless transformations such as clock offsets and channel transformations.*

systems, dominant in the United States, the client devices measure the wireless channel using extra preamble symbols transmitted by the base station and send it as feedback to the base station. However, this feedback is unsustainable and causes huge spectrum waste. Leveraging the intuition that both uplink and downlink channels are generated by the same underlying physical environment, recent work [55] proposed FIRE which uses an end-to-end ML based approach to predict the downlink channels without any feedback from the client. FIRE's ML model uses a variant of the variational autoencoders (VAE). FIRE takes uplink channels measured at the base station as input and predicts the downlink channel. The accuracy of the downlink channel prediction determines the performance of multi-antenna techniques like MIMO, MU-MIMO. Our goal is to induce errors in this model and make it predict erroneous downlink channels. Errors in downlink channel estimates reduce the communication efficiency of multi-antenna systems (e.g., MIMO).

B. DLoc: Deep Learning based Wireless Localization – DLoc [6] is a deep learning based wireless localization algorithm that overcomes traditional limitations of RF-based localization approaches such as mutlipath and occlusions. DLoc acquires wireless channels from four fixed access points to the user device that it wants to localize. The wireless channels are then sent into an autoencoder neural network as input. The network predicts the user location in 2D-Cartesian coordinates. DLoc achieves state-of-the-art localization accuracy in indoor localization. Therefore, we choose DLoc as the representative sensing application. Our goal is to increase the localization error of DLoc, and hence, increase failures in location-based applications (such as robotic navigation).

3.2 Operation Overview

Both the systems defined above rely on the wireless channel estimated at the base station. Our goal is to alter this wireless channel by transmitting a perturbation vector during the channel estimation process. During the channel estimation process, the client transmits a preamble. The preamble is by the standard and is, therefore, public knowledge. The base station receives the preamble and uses it to identify the wireless channel. Due to the broadcast characteristic of the wireless channel, we can pollute the channel estimation process by transmitting a noise signal. RAFA transmits the adversarial perturbation using our custom hardware platform. Thus, the signal received at the base station is a function of both the preamble transmitted by the client and the noise transmitted by our platform.

Fig. 2 shown an overview of RAFA's operation. RAFA's algorithm (discussed in Sec. 4) computes an adversarial perturbation which will be transmitted by the RAFA hardware, described in Sec. 5. The base station receives a sum of the signal transmitted by the client and the adversary, with channel and clock distortions. This combined signal then passes through the pre-processing steps and the ML-models defined above. We evaluate the performance of the ML-models with and without the adversarial perturbation in Sec. 6.

Notation: For the rest of this paper, we use capital-bold to denote matrices (**M,N,...**), small-bold to denote vectors (**u,v,...**), and mathematical font to denote functions or ML-models ($\mathcal{A}, \mathcal{L}, \ldots$).

4 RAdio Frequency Attack (RAFA)

In this section, we present our formulation of RAFA. We start with a brief background on adversarial attacks then discuss our attack formulation and modeling of wireless properties. Finally, we discuss our algorithmic implementation of RAFA.

4.1 Background on Adversarial Perturbations

For a given trained ML-model $\mathcal{M}$ mapping inputs $\mathbf{x_i}$ to outputs $\mathbf{y_i}$, an adversarial attack algorithm aims to find a perturbation vector, $\mathbf{v_i}$, whose magnitude is bounded by a small constant $\varepsilon \in \mathbb{R}$ in some norm, such that the loss function $\mathcal{L}(\mathcal{M}(\mathbf{x_i}+\mathbf{v_i}), \mathbf{y_i})$ is maximized, i.e., the output of the model for the perturbed input $\mathbf{x_i}+\mathbf{v_i}$ is far away from the target $\mathbf{y_i}$. Formally, the attack algorithm solves the following optimization problem:

$$\arg\max_{v_i} \mathcal{L}(\mathcal{M}(\mathbf{x_i}+\mathbf{v_i})) \text{ s.t. } ||\mathbf{v_i}|| < \varepsilon \tag{1}$$

This attack problem formulation is well studied and many approaches [16, 27, 57] have been proposed to approximate this optimization problem. We utilize the state-of-the-art Projected Gradient Descent (PGD) [57] method. PGD iteratively takes steps in the direction of the gradient while restricting the total perturbation to be within ε. The constraint on the norm of the noise vector ensures that the perturbed vector is not significantly different from the original input $\mathbf{x_i}$. We note that PGD requires access to the model parameters for computing gradients. One way to handle the black-box setting where the attacker does not have this access is to train a surrogate model on the training distribution of the original model (to which the attacker has access) and transfer the attacks computed on the surrogate model to the original model.

As discussed before, our model does not know the input to the model, therefore we rely on universal adversarial perturbations (UAPs) [58]. Instead of computing a different additive noise for each input, UAPs compute a single additive noise that is effective for all inputs, $X = \{\mathbf{x_i}\}_{i=1}^{N}$ in the training distribution of the ML model. One way to encode this is by maximizing the expected value of the loss.

$$\arg\max_{\mathbf{v}} \mathbb{E}_{\mathbf{x_i} \in X} \mathcal{L}(\mathcal{M}(\mathbf{x_i} + \mathbf{v}), \mathbf{y_i}) \text{ s.t. } ||\mathbf{v}|| < \varepsilon \qquad (2)$$

Typically, this optimization problem is approximated by iteratively computing input-aware gradient updates (obtained from the original or a surrogate model) to a perturbation vector over the training set [58]. We use a variant of the UAP, called robust UAP [83], which designs a universal adversarial perturbation such that it is robust to transformations (such as image rotations, and translations etc for vision models).

4.2 Our Attack Formulation

For simplicity, we start with a single input case, i.e., we wish to design a perturbation vector that is specific to wireless channel matrix, $\mathbf{H}$, observed at the base station. $\mathbf{H}$ is a $N_{ant} \times N_{subc}$ matrix, where N_{ant} is the number of antennas on the base station and N_{subc} is the number of OFDM subcarriers. Our goal is to search for a perturbation vector, $\mathbf{v}$, of length N_{subc} that can disrupt a machine learning model $\mathcal{M}_\theta$, where θ is the set of weights for the model. Specifically, we aim to optimize:

$$\arg\max_{v} \; \mathbb{E}_{T_\tau \in \mathcal{T}_\tau} \mathcal{L}(\mathcal{M}_\theta(\mathcal{P}(\mathbf{H} + T_\tau(\mathbf{v}))), \mathbf{y}), \;\; \text{s.t. } ||\mathbf{v}||_p < \varepsilon \qquad (3)$$

where $\mathcal{L}(\cdot,\cdot)$ is a chosen loss function to measure the difference between the model's prediction and the ground truth, ε restricts the l_p norm of the perturbation vector. We also define two new abstractions in the equation above: $\mathcal{P}(\cdot)$ is the pre-processing pipeline used by the base station, before it is fed to the ML-model, $\mathcal{M}_\theta$. Similarly, $\mathcal{T}_\tau$ represents the transformations T_τ the perturbation vector goes through before it arrives at the base station. These transformations are parameterized by τ. Next, we discuss how these abstractions model the real-world effects for wireless systems.

4.3 Modeling Pre-Processing

Both our target systems perform pre-processing on the estimated channel and feed the processed channel to the neural network for prediction. We model these pre-processing steps as $\mathcal{P}$. FIRE involves two pre-processing steps: it standardizes the Carrier Frequency Offset (CFO) and hardware detection delay across different measurements of the same channel. DLoc transforms the channel into the 2D-cartesian heatmap representing the probability of a signal originating from a given location (using signal-processing approaches like Fourier Transforms).

We need to represent $\mathcal{P}$ as a differential operation, to enable optimization in Eqn. 3. In both pre-processing methods, there are non-differentiable functions such as `argmax()`, `ceil()`, `sign()` that hinder gradient propagation for our optimization problem. Therefore, we use a differentiable approximation of these functions which is supplied by common ML frameworks such as Pytorch [61].

4.4 Modeling Lack of Synchronization

Since the adversary is not synchronized with the client or the base station, the noise transmitted by it experiences two kinds of distortions that must be modelled by $\mathcal{T}_\tau(\cdot)$ defined above:

Carrier Frequency Offset: The oscillators at the adversary and the base station are not synchronized. This leads to a CFO between them. This frequency offset, denoted by Δf, will continuously add a phase shift in the received signal $\hat{s}(t)$ with respect to the true signal $s(t)$ over time: $\hat{s}(t) = s(t)e^{j2\pi\Delta f t}$.

Since the client and attacker have different transmission chains, their CFO with respect to the base station is also different. Assume the CFO between the client and the base station is Δf_1, the CFO between the attacker and the base station is Δf_2, thus the CFO discrepancy will add phase offset $e^{j2\pi(\Delta f_1 - \Delta f_2)t}$ to the transmitted adversarial signal with respect to the client signal. This is a time-varying effect, implying that the sum of the client signal and the adversary signal changes over time. Since t, Δf_1 and Δf_2 are random, we can simplify this effect as a multiplication of $e^{j\phi}, \phi \in [0, 2\pi]$ to the adversary signal.

Unsynchronized Transmissions: Ideally, the adversary should start transmitting the perturbation signal at the same time when the client starts transmitting the preamble so that the perturbation can be superimposed at the base station precisely. In the real-world setting, this is hard to achieve since the attacker cannot coordinate with the client preemptively or synchronize its clocks. There are two possible approaches to solve this problem: (a) A part of the signal preamble is used for packet detection before the channel estimation phase. One can design an attacker that detects the start of the packet, and starts transmitting a perturbation in response. This can achieve coarse-grained synchronization, but requires fast processing (e.g., FPGAs). (b) An alternative approach is to let the adversary transmit multiple copies of the perturbation signal, and deal with the resulting large mis-alignment.

For simplicity, we go with the latter approach, which requires much lower overhead. This causes a random time delay Δt between the benign client signal and the adversary signal in the time domain. Due to the properties of OFDM, this is equal to a phase offset $e^{-j2\pi\Delta t f_i}$, where f_i is the frequency of the ith OFDM subcarrier. Note that, this is similar to timing misalignment in typical OFDM receivers [24, 77], wherein a portion of the OFDM symbol is repeated (in the cyclic prefix). Any sample misalignment adds a phase that is linearly dependent on the amount of misalignment and the frequency of the subcarrier. We model this phase shift in $\mathcal{T}_\tau(\cdot)$.

4.5 Modeling Channel Transformations on the Perturbation Vector

Like any other wireless signal, the perturbation vector transmitted by the adversary goes through the wireless channel. Let us say that the wireless channel matrix for the adversary is, $\mathbf{H_a}$, with dimensions $N_{ant} \times N_{subc}$ (same dimensions as $\mathbf{H}$ for the client). Then, if the adversary transmits the perturbation vector, $\mathbf{v}$, it is received at the base station as $\mathbf{H_a v} + \mathbf{g}$, where $\mathbf{g}$ is additive white Gaussian noise.

Each element in $\mathbf{H_a}$ has an amplitude and a phase. Therefore, the final outcome of the added perturbation can be unbounded, if we do not know $\mathbf{H_a}$. However, $\mathbf{H_a}$ can only be measured at the base station, that too in the absence of the client signal. Clearly, measurement by this method is not possible for the attacker because it does not have any coordination with the client or the base station. Therefore, it is pertinent to find an alternative method to measure $\mathbf{H_a}$.

To estimate $\mathbf{H_a}$, we leverage channel reciprocity. Channel reciprocity is a fundamental physical principle that states that wireless signals take the same path in either direction between any two devices. Therefore, the wireless channel from the adversary to the base station is equal to the wireless channel from the base station to the adversary (modulo some hardware differences). RAFA listens to the wireless medium and captures signals transmitted by the base station (either periodic beacons or communication with legitimate clients). This allows the adversary to estimate an approximation to $\mathbf{H_a}$. We discuss in Sec. 5 how this step is implemented in practice.

4.6 Generating Practical Adversarial Attacks

The above factors will jointly modify the adversarial perturbation signal v^i transmitted in the ith subcarrier by the following transformation function:

$$\mathcal{T}_{\phi,\Delta t,\mathbf{h}_a^i,\mathbf{g}_i}(\mathbf{v}^i) = \mathbf{v}^i e^{j\phi} e^{-j2\pi f_i \Delta t}\mathbf{h}_a^i + \mathbf{g}_i \tag{4}$$

Note that the CFO term ϕ is invariant to the frequency, so it's the same across all the subcarriers.

Algorithm 1: Robust Wireless UAP (`RW-UAP`)

```
Input : Dataset $\mathbf{x_i}, \mathbf{y_i} \in \mathcal{H}$, network model $\mathcal{M}_\theta$, $l_\infty$
        norm threshold $\varepsilon$, desired network loss value
        $\delta$, attacker channel $\mathbf{H_a}$, number of epochs $ep$
Output: Robust universal perturbation $\mathbf{v_g}$ for dataset
        $X$
Initialize $\mathbf{v_g} \leftarrow Uniform(-\varepsilon, \varepsilon)$
for $n \leftarrow 0$ to $ep - 1$ do
    for each batch $\mathbf{B}_i \subset \mathcal{H}$ do
        $\Delta\mathbf{v_i} \leftarrow$ RW-PGD $(\mathbf{B}_i, \mathbf{v_g}, \mathcal{M}_\theta, \varepsilon, \delta, \mathbf{H_a})$
        $\mathbf{v_g} \leftarrow (\mathbf{v_g} + \Delta\mathbf{v_i})$.clamp$(-\varepsilon, \varepsilon)$
    end for
end for
return $\mathbf{v_g}$
```

Now that we have characterized both $\mathcal{P}$ and $\mathcal{T}_\tau$, we try to find a perturbation that is robust to these wireless factors. We build on the algorithm presented in [83] which works in the vision domain to work with $\mathcal{P}$ and $\mathcal{T}_\tau$, Algorithm 1 shows the pseudocode for generating Robust Universal Adversarial Perturbations in the wireless setting. It contains twp loops to iterate multiple times on the training dataset and on every batch of data points respectively. We first initialize the perturbation vector $\mathbf{v_g}$ randomly. Because of random initialization, we can generate several different UAPs by running the algorithm multiple times. The algorithm iteratively updates the initial perturbation with the goal of being adversarial for all elements in the training set. Given a set of training data points $\mathcal{H}$ sampled from the training distribution of the ML model, the algorithm iterates over batches $\mathbf{B} \subset \mathcal{H}$. During each epoch, it iterates over every batch, $\mathbf{B}_i$, finding an adversarial direction vector $\Delta\mathbf{v_i}$ that is robust to wireless factors for that batch using the Robust Wireless PGD (`RW-PGD`) algorithm shown in Algorithm 2. $\Delta\mathbf{v_i}$ is then added to $\mathbf{v_g}$ and the result is projected back so that the updated $\mathbf{v_g}$ does not violate the constraint on its norm.

Next, we describe the `RW-PGD` algorithm shown in Algorithm 2. It contains two loops to iterate multiple times on the input data points and on different transformations respectively. The algorithm takes a batch of inputs, $\mathbf{B}$. It then first randomly samples N tuples of wireless factors including CFO, resynchronization, and gaussian noise as described in the previous section. Increasing the value of N increases both the robustness of the output perturbation and the runtime of the `RW-PGD` algorithm. We chose a value of N that balances the tradeoff between cost and robustness. At each iteration, we transform our current perturbation by each of the N transformations. We then compute the mean loss over all data points in the batch added to each of our N transformed perturbations and conduct gradient ascent on it aiming to increase the mean loss. Unlike traditional PGD [57], we compute a single vector per batch of data points. We further found that using the raw gradient

Algorithm 2: Robust Wireless PGD (`RW-PGD`)

```
Input  : Batch of data points $\mathbf{x_i}, \mathbf{y_i} \in \mathbf{B}$, current
         perturbation $\mathbf{v_g}$, network model $\mathcal{M}_\theta$, $l_\infty$ norm
         threshold $\varepsilon$, desired network loss value $\delta$,
         attacker channel $\mathbf{H_a}$, maximum number of
         epochs $ep$, learning rate $\alpha$, number of
         transformations $N$
Output : Robust perturbation $\mathbf{v}$
Initialize $\mathbf{v} \leftarrow 0$
Sample $N$ sets of wireless factors,
  $\tau_j \leftarrow \{\phi_j, \Delta t_j, g_j, \mathbf{H_a}\}$, uniformly at random
for $n \leftarrow 0$ to $ep - 1$ do
    for $\mathbf{x_i}, \mathbf{y_i} \in \mathbf{B},\ j \in [N]$ do
        Get predictions for N transformations:
        $\mathbf{y^*_{i,j}} \leftarrow \mathcal{M}_\theta(\mathbf{x_i} + \mathcal{T}_{\tau_j}(v_g + v))$
    end for
    $\mathcal{L}_B \leftarrow \frac{1}{N \cdot |\mathbf{B}|} \sum_i \mathcal{L}(\mathbf{y_i}, \mathbf{y^*_{i,j}})$
    if $\mathcal{L}_B > \delta$ then
        break
    end if
    $\mathbf{v} \leftarrow (\mathbf{v} + \alpha \cdot \Delta \mathcal{L}_B).\text{clamp}(-\varepsilon, \varepsilon)$
end for
return $v$
```

to update was more effective than the $sign(\cdot)$ of the gradient in our case. Note that, $\mathbf{H_a}$, i.e., the wireless channel between the attacker and the base station, is fixed and obtained by the attacker without sending any preambles.

$\mathbf{H_a}$ is continuously sampled by the attacker, so we need to recompute the attack on the fly. This is inevitable as different $\mathbf{H_a}$ have essentially unbounded effect on the perturbation. Thus, we further speed up our algorithm by computing multiple UAPs at the same time. We have chosen algorithm parameters that maximize the speed for our required performance.

By using the above algorithm, we are able to find robust universal perturbations that work effectively against a variety of wireless factors. In order to truly expose the vulnerabilities of wireless system models, we show in a real-world hardware setting that our attack is effective.

5 Hardware Design

After identifying the adversarial noise to inject, we ask if this noise is feasible in practice, i.e., can we design hardware that can introduce such noise? In this section, we design a generic physical hardware platform as an attacker to inject such perturbation into wireless channels. We believe that this step is novel and unique to the wireless domain because no past work has demonstrated hardware-driven attacks on wireless systems. We describe the design principles of RAFA's hardware platform and its implementation.

5.1 Design Principle

We design our platform with following design principles:

No Synchronization Needed: We design RAFA's platform to have minimum assumptions, no requirement to access the client and the base station. Our attack belongs to the realm of pilot contamination but previous literatures [28, 36, 37] all assume that the attacker know the exact timing to synchronize with the client so that it can transmit the noise signal at the same time when client transmits the preamble. In our design, we get rid of this assumption and instead, transmit the attacker's perturbation in a unsynchronized manner. According to Sec. 6.2, our perturbation is robust to such unsynchronization on the sample level, as a result, we don't need the timing of the client to conduct effective attack.

Leveraging Reciprocity: Recall that we leverage reciprocity to estimate $\mathbf{H_a}$, the channel from the attacker to the base station. However, reciprocity requires correcting for the hardware effects caused by each respective transmitter. These effects comprise of phase and signal strength variations. The phase variations are captured in the transformations caused due to CFO and timing offsets for our perturbations. Therefore, in our attack, we just need to calibrate for the difference in transmit power between the attacker and the base station.

Specifically, the transmit power is different at the attacker and the base station (the base station transmits a much higher power). Although transmit powers don't affect the applications enabled by reciprocity such as beamforming, signal nulling, etc. [41, 47, 53], we need to know the true channel value including the power in order to control the received perturbation power at the base station, as shown in Algorithm 2. The adversary can obtain the transmit power of the base station using public documents like FCC filings. We can estimate the transmit power of the adversary hardware using specification sheets. Then, the adversary computes the difference between the two and uses it to correct for the transmit power difference.

Single Antenna: We design RAFA's hardware platform with a single antenna. This limits our hardware complexity and making it easy to implement (single transmit-receive chain). One class of adversarial attacks on traditional systems [42, 44] is only effective when the attacker has same or more number of antennas as the base station. In our case, we show that even with one antenna, it is possible to mount reasonable attacks.

However, this choice limits the capability of the attack. While the input channel matrix, $\mathbf{H}$, has dimension $N_{ant} \times N_{subc}$, the perturbation is a vector $\mathbf{v}$ with length N_{subc}. Inherently, this implies that the perturbation has less control over the final output. Mathematically, $\mathbf{v}$ operates in a 1-dimensional subspace of a N_{ant} dimensional antenna space. The larger the N_{ant}, the less powerful our adversary. We expect multi-antenna adversaries to be more effective, but we chose single antenna adversaries for their low complexity.

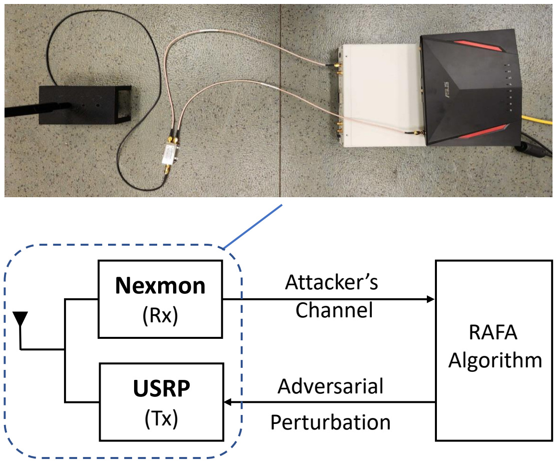


Figure 3: *(Top) The top-down view of the attacker hardware platform. (Bottom) The platform consists of a Nexmon-receiver and a USRP transmitter – both share a single antenna connected through a splitter.*

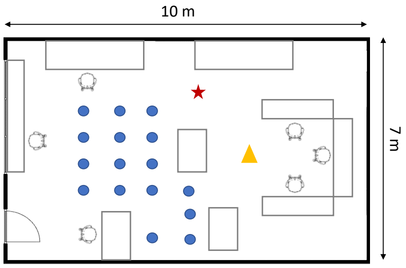


Figure 4: *Layout of the experimental environment: We conduct our experiments in a lab setup measuring 10m by 7m. (Blue: Client Locations, Yellow: Base station, Red: Attacker)*

5.2 System Implementation

We use the Nexmon [29] tool to measure wireless channels. We build a 4-antenna base station using a commercial Asus RT-AC86U router with bcm4366c0 WIFI Chip. This router reports channel state information (CSI) and signal strength (RSSI) for each received packet. We configure a client to connect to this router and send Wi-Fi packets. The router periodically sends broadcast beacons, as is standard for the Wi-Fi protocol and is set to work in the 5GHz frequency band.

For the attacker, we use a Nexmon receiver and a USRP (X310 [21], a software-defined radio) as the transmitter. The Nexmon receiver and the USRP transmitter share a single antenna through a splitter. The Nexmon receiver measures the wireless channel from the base station to itself and feeds it into the RAFA system. RAFA applies the reciprocity correction to estimate, $\mathbf{H_a}$, and computes an adversarial perturbation, $\mathbf{v}$, and transmits it using the USRP software-defined radio. The setup is shown in Fig. 3

We implement the client using a USRP software-defined radio (X310) equipped with one antenna and we configure it to transmit the 802.11ac packets generated from MATLAB wireless toolbox. This signal is received by the base station and used to measure the wireless channels as the input to the neural networks in the wireless applications mentioned in Section 3.1. Note that, the client and attacker do not have any time or frequency synchronization between them.

During our experiment, we deploy RAFA on a local machine with RTX3070 GPU, a low-end GPU. Our implementation is written in PyTorch. For attacking FIRE, it takes 13 seconds on average to generate a single perturbation. We believe that further speedup can be achieved using more advanced computing resources.

6 Results

In this section, we show the effectiveness of the RAFA attack against FIRE and DLoc. We further show that our attack can be performed in a black-box setting. Finally, we show that adversarial training can be used in order to harden our models and defend against these attacks.

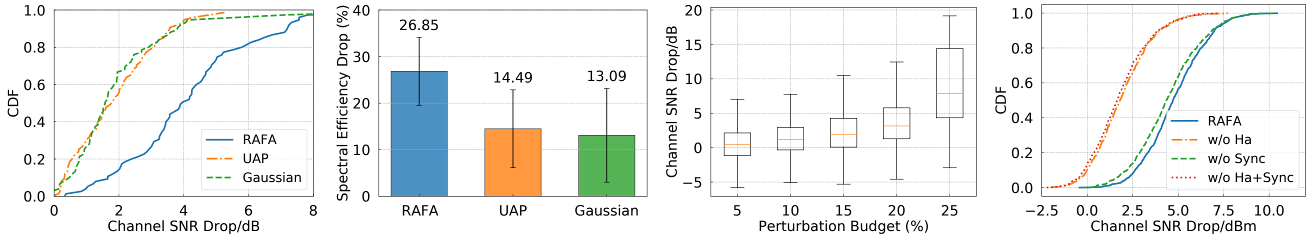
Baselines: We compare RAFA against two baselines: (a) Gaussian noise, and (b) vanilla UAP. Gaussian noise transmits randomly sampled noise into the air. Vanilla UAP designs and injects perturbations attacks that do not include robustness to wireless transformations implemented by RAFA. For fairness, we evaluate each method with the same budget ε on the magnitude of the perturbation vector measured in L_∞-norm.

6.1 Wireless Systems Re-implementation

We do a best-effort re-implementation of our target systems: FIRE [55] and DLoc [6] using details provided in the respective papers and by email exchanges with the authors.

We re-implement the Variational Autoencoder for FIRE. We adopt 7 linear layers in both the encoder and decoder networks, which is 3 layers more than the original design to optimize FIRE's performance. We train FIRE using dataset collected in our environment. We collect 10000 data points by moving the antenna randomly in a lab space shown in Fig. 4. The size of the lab is 10m by 7m, and is composed of many reflectors (like metal cupboards, white-boards, etc.) and obstacles. We split the dataset in the radio of 8:2 for training and testing , and the training takes roughly 20 minutes. Note that the training only needs to be done once before the attack is initiated. Our trained model achieves a channel SNR of 17.8dB on the test dataset and confirm the SNR of 15.8dB on validation dataset, which is consistent with the performance of FIRE reported in [55].

Training DLoc requires dataset collected using a robot (which performs joint mapping and localization). We do not have access to the robot, thus cannot recreate the original experiments. However, the datasets and code for DLoc are in the public domain. Therefore, we train DLoc model using the datasets collected by its authors with the name 'jacobs_Jul28' and keep the original neural network architecture and training setting. The dataset contains channel estimation matrices collected from 4 routers and each router has 4 antennas (so 16



(a) Attack Performance (b) Spectral Efficiency Drop (c) Perturbation Budget Analysis (d) Ablation Study

Figure 5: *Attacking FIRE: (a) RAFA causes a median drop of 4.06 dB on FIRE's channel prediction accuracy, over 2X better than baselines. (b) This corresponds to a 26.85% drop in spectral efficiency. (c) Increasing the perturbation budget increases the effectiveness of the RAFA adversary. (d) Modelling channel reciprocity is the dominant reason for RAFA's improved performance.*

antennas in total), and the training takes roughly 40 minutes. We achieve the localization accuracy of 1.04m on the test dataset which is randomly split with the ratio of 3:7 to the training dataset. This is consistent with the results reported by the authors in [6]. To create the adversarial perturbation for RAFA, we sample channels from the same distribution as the data used for training.

6.2 Adversarial Attacks against FIRE

To mimic a real-world setup, we deploy the base station at a fixed location, while the client moves across the 16 positions shown in Fig. 4. The attacker is also deployed at a fixed location shown in the figure. The attacker and the client can be in line-of-sight or non-line-of-sight.

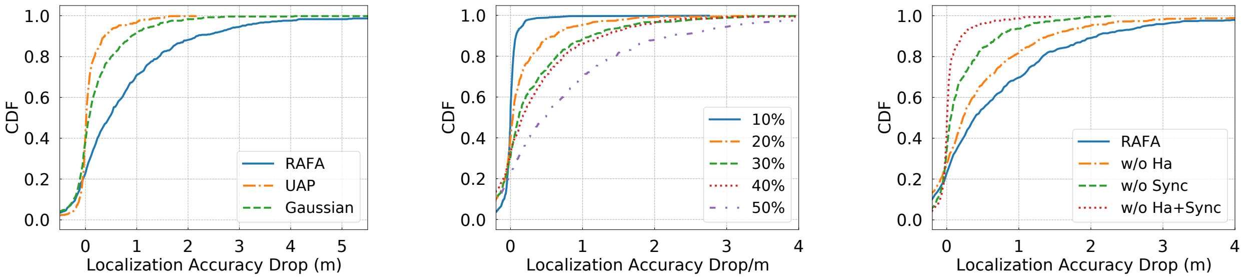
Attack Effectivenss: We first compare RAFA with other two baselines in the real world attack scenario. FIRE predicts downlink channels, given uplink channels. Our goal is to reduce the SNR of the predicted channels, so as to disrupt multi-antenna communication between the base station and the client. To perform this experiment, we perform five attack rounds for every method at each client location. In each round, the adversary transmits the RAFA adversarial perturbation for 10 seconds. Then, we transmit the UAP adversarial perturbation and Gaussian perturbation each for 10 seconds at the same perturbation budget as RAFA which is set to be maximum of 20%. The budget is the ratio of the maximum value that's allowed for any element of the perturbation vector compared to the average amplitude of the benign channel estimates and is ε in Algorithm 1. Overall, we get 8000 channel estimates at each client location for every baseline. We set the learning rate of RW-UAP to start from 10 and decay by 0.6 for every epoch. We set the number of total epochs in RW-UAP to be 3 and we only use a random 10% of the training dataset for each epochs to accelerate the training which leads to a running time of roughly 30 seconds with the setup in Sec. 5. We set the RW-PGD iterations to be 10, and 10 transformations in each iterations are used to get the robust perturbations. We use the same parameters (e.g., number of epochs, % of training dataset) for training Vanilla UAP.

We show the effect of different attack methods on FIRE prediction in Fig. 5a. Each of these methods causes the SNR of the predicted channel to drop. We plot the CDF of this drop in channel SNR across all of our attacks, e.g., a CDF value of 0.3 with a corresponding drop of X dB indicates that 30% of the inputs had a drop of X dB or less in performance. It shows that when being attacked by RAFA, the SNR of the channel predicted by FIRE drops by 4.06dB on average (7.15 dB drop on the 90-th percentile). Traditional UAP attacks and Gaussian baseline are not as effective. RAFA outperforms the baselines by 2.21× and 2.14× respectively on this metric. Our benefits over Gaussian noise stem from the directed nature of our attack, i.e., we specifically target the ML model and find its vulnerability. On the other hand, the UAP-based model finds perturbations that are directed at the ML model, but undergo transformations in-air which render it ineffective when implemented in the real world. This result highlights that: (a) FIRE is vulnerable to practical adversarial attacks, even when the adversary uses low-complexity hardware, and (b) modelling the wireless transformations on the adversarial noise (as done in RAFA) are essential for practical adversarial attacks.

We also show how these adversarial attacks affect the application quality in the real world. We plot the spectral efficiency (bits per Hz) results with the budget of 20% in Fig. 5b. Spectral efficiency is the data rate that can be transmitted over a given bandwidth and can be computed through channel SNR [62]. With small amount of budget, RAFA is able to shrink the user data rate by 26.85% which is two times as effective as the UAP attack. This will affect the user experience severely, especially in latency critical applications such as online meetings.

Finally, we note that, we focus on reducing SNR for FIRE, because FIRE is trained to optimize for SNR. SNR is also a key metric for any communication techniques. Independent mechanisms like coding and CRC checks do not prevent against attacks like RAFA. For example, if a coding scheme is chosen to optimize for SNR X dB, it won't be sufficient if the actual SNR is X-5 dB (when under attack).

Effect of Adversarial Budget: In the same setup as above, we study the effect of budget on the effectiveness of the attack. We experiment with 5 different budget parameters and plot the SNR drop for these parameters in Fig. 5c. As expected, as



(a) Attack Performance (b) Perturbation Budget Analysis (c) Ablation Study

Figure 6: *Attacking DLoc: (a) DLoc's original localization error (median) is 1.04 m. RAFA increases this by 0.71m. (b) The attack performance improves with increasing perturbation budget. (c) Synchronization robustness is the most important factor in attacks on DLoc.*

Budget (%)	10	20	30	40	50
Power Ratio-FIRE (%)	1.5	4.7	8.6	14.7	12.0
Power Ratio-DLoc (%)	1.3	8.6	20.2	34.6	51.8

Table 2: *Power Ratio vs Budget*

the budget parameter increases, RAFA's attack becomes more effective. With the budget parameter set to 25%, the SNR of the predicted channel drops by 7.4 dB on average, with lower drops at lower budget values (e.g., 2.14 dB at 15%).

Note that, the budget defines what's the maximum absolute value of any perturbation subcarrier, and not the average value. Therefore, the average power of the perturbation is expected to be lower than the budget. We compare the average power of the adversarial signal to the average power of the user signal at the base station. We show this ratio as a function of the budget parameter in Table. 2. As shown, for 20% budget, RAFA's perturbations utilize <5% power on average compared to the user signal. Even with a 50% budget parameter, this value is only 12%. This shows that RAFA's attacks are surreptitious. Note that, counter-intuitively, the power ratio at 50% is lower than 40% budget parameters. This is possible because bigger budget parameters allow some subcarriers to have larger peaks, and enable more flexibility for RAFA to choose "peakier" more effective perturbation vectors.

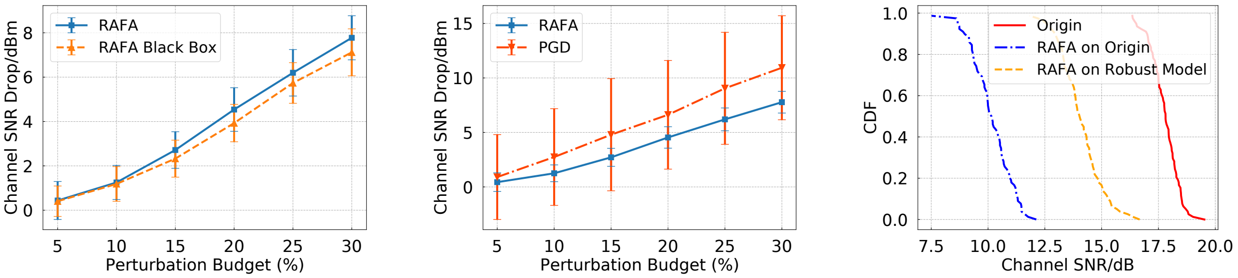
Ablation Study: Next, we analyze the contribution of different components of RAFA. We conduct an ablation study on attacking FIRE using the same dataset that we used for real-world attack experiment. We compare the original RAFA effectiveness with the following cases: removing the attacker channel ($\mathbf{H_a}$), removing the synchronization robustness term (Sync), and removing both of them. The results are shown in Fig. 5d. For FIRE application, removing the knowledge of the attacker's channel has the most significant impact on the effectiveness of the attack, the channel SNR drop decreases by 59.5% compared to the optimal performance achieved by RAFA. This is because our `RW-UAP` algorithm derives an adversarial perturbation specifically for a given attacker's channel. When removing the synchronization robustness term, the SNR drop decreases by 7.5%. We believe that this effect is milder because FIRE's design includes some pre-processing to normalize CFO, hardware detection delays, etc.

6.3 Adversarial Attacks against DLoc

Next, we study the effect of RAFA on DLoc. DLoc conducts user localization using the channels estimated from 4 routers with a total of 16 antennas as the input to a neural network, consisting of 12 Resnet blocks [32]. Compared to FIRE, which uses a single router, four antennas, and a simpler network structure, the attack scenario is harder.

Experiment Setup: As noted before, getting ground truth location estimates for DLoc in new environments requires a robot for data collection. We perform our attack in a trace-driven simulation using the data collected by the authors as we do not have access to their robot. We randomly sample, $\mathbf{H_a}$, the attacker's channel to each access point from the set of channels in the original DLoc dataset. Then, RAFA's perturbation undergoes the attacker channel in addition to random time and frequency offsets. We repeat this experiment with different values of $\mathbf{H_a}$ to remove any bias. Our attacker is still a single antenna attack. We limit to a set of 128 datapoints, out of 8008, randomly sampled from the original training set and use only 4 transformations during `RW-PGD`. We train the perturbation for 3 epochs. The learning rate is set to 0.006 and decays by 0.99 per iteration. We evaluate on randomly sampled $\sim$500 datapoints from the test set and average the performance of the attack over 8 randomly sampled transformations to generate different wireless transmissions. The same parameters are used for training Vanilla UAP.

Attack Effectiveness: The adversary aim is to reduce DLoc's localization accuracy. We plot the accuracy degradation caused by RAFA, traditional UAP, and Gaussian noise in Fig. 6a. The attacker has a single antenna simultaneously attacking 16 infrastructure antennas, so we set a budget of 50%. The original DLoc median localization accuracy is 1.04 meters, RAFA is able to increase this error by 0.71 meters. Furthermore, RAFA increases the error by more than 1 meter in 30 % of cases. This is significant as some safety-critical applications such as autonomous driving [66,67] are sensitive to even 0.1 meters of localization accuracy drop. Our two baselines, UAP and Gaussian are only able to drop the accuracy by 0.02 m, and 0.08 m respectively. Similar to our discussion before, this shows the benefit of our directed, robust attack.



(a) Black box attack (b) Comparison to PGD attack (c) Robust Training

Figure 7: *Other Attack Models and Defense on FIRE: (a) A black-box attack, where RAFA does not know about the model running on the base station is feasible and has similar attack performance to the white-box attack. (b) An unrealistic attack, where adversary has access to model inputs, performs better than RAFA, but is not practically feasible. (c) Our adversarial training approach can improve model robustness on FIRE.*

Next, we study how choosing different budgets will affect RAFA's potency on DLoc. We increase the budget from 10% to 50% and plot the DLoc accuracy on Fig. 6b. We also calculated the power ratio of the perturbation generated on DLoc in the second row of the Table. 2. As expected, as the perturbation budget increases, the power used by RAFA increases. Similarly, the attack efficacy increases. We highlight that the power used by RAFA for attacking DLoc is higher than that for FIRE. This is because we have a single-antenna attacker attacking a sixteen-antenna system spread out in space (one-dimensional control in a 16-dimensional space). RAFA adapts to this large space by increasing the power of its transmitted noise, achieving an error increase of 0.71 m at 50% budget.

Ablation Study on attacking DLoc: We conduct an ablation study on DLoc to understand the contribution of RAFA's different components, using the same 50% budget from above. Similar to ablation study settings on FIRE, we compare the original RAFA effectiveness with the cases when removing Ha, removing Sync and removing both, and the results are shown in Fig. 6c. Different from FIRE's ablation study results, it shows that removing Sync term will bring down the effectiveness the most, the median localization accuracy drop decreases by 73.4% compared to the original RAFA. This is because DLoc localizes the user by receiving the signal at four routers instead of one in FIRE case. Thus misalignment becomes much more significant when the attack signal reaches multiple routers and modeling this effect is necessary. Compared to the ablation study on FIRE, we confirm that different components in RAFA modeling have different impact on different applications. So, to expose vulnerabilities on generic wireless algorithms, we should leverage the end-to-end model with all the components of RAFA.

6.4 Comparison to Input-Aware Attacks

RAFA is an adversarial attack that aims to work in the real-world. Therefore, it does not assume access to the input. What is the impact of this assumption on RAFA's performance? What if we give RAFA access to the input. We evaluate this hypothetical case next. We compare RAFA's adversarial performance with input-aware attack, e.g, PGD. Since this attack is not realistic, we evaluate this in a simulator. We plot the result in Fig. 7b. Note that we implement the PGD by using Algorithm. 2, for each data point in the test dataset, we get a PGD perturbation that is robust to wireless properties, but this perturbation is not universal across data point.

The results show that an input-aware attack is more effective. At 30% budget on FIRE, the input-aware attack achieves an SNR drop of 10.97 dB, compared to the 7.77 dB drop of RAFA. This result highlights that the properties of the wireless medium (e.g., the adversary not knowing the channel from client to base station) provide some natural protection against adversarial attacks. However, even with this protection, real-world adversarial attacks are possible and effective.

6.5 Black Box RAFA

In this section, we evaluate RAFA under a preliminary black box setting to show its feasibility, where the attacker knows the model family but not the specific architecture or weights. In order to conduct the black box attack, the attacker can train a substitute model on a dataset with a similar distribution to attack. The attacker can later use the obtained perturbation from the substitute model to attack the true model.

In order to conduct the black box attack, we use a substitute model with a different architecture (4 less layers, different number of neurons, and batchnorm). Using the same pipeline for RAFA we attack on a different dataset collected in the same lab. We then use that perturbation to attack the original model in a trace-driven simulation. Fig. 7a compares the performance of RAFA in white box and black box scenarios under different perturbation budgets. As shown, the performance of the black-box adversary closely matches that of the white-box adversary, with only a minor drop. At 25% perturbation budget, the black-box adversary causes a 5.7dB of channel SNR drop for FIRE which is only 0.5dB less than the white box setting. This result shows that while access to the model helps, we can still get good performance without it.

6.6 Defense: Adversarial Training

RAFA's attacks demonstrate effectiveness across different systems and settings. To initiate a discussion on potential defense strategies, we show that FIRE can enhance its robustness to adversarial attacks with robust training. Adversarial training involves adding adversarial examples while training [57]. In our case, since we are defending against a UAP based attack, we use our `RW-PGD` algorithm in order to compute batch-wise perturbations. We choose the `RW-PGD` algorithm since it is similar but stronger than `RW-UAP` as it assumes that the attacker has access to the inputs processed by the ML model.

During training, we compute 5 random attacks on each batch with a budget of 10%. We then apply each attack to the entire batch before learning. While this method adds overhead during training, it significantly reduces the ability of RAFA to find successful attacks. With our initial study and parameters we find that training time goes from ∼5 mins to ∼125 mins; however, we expect that further tuning could significantly reduce this training time. Fig. 7c shows the effect of RAFA at the budget of 30% on FIRE. When applying RAFA on the original model, the average channel SNR provided by FIRE drops from 17.79dB to 10.16dB. Promisingly, the robust model maintains an SNR of 14.09dB, which is 38.6% higher than original model improving the model robustness by 57%. This result highlights the potential for building robust training approaches. We delegate a detailed study of such methods to future work.

7 Related Work

Adversarial Attacks in Other Domains: Adversarial attacks have been widely studied for measuring model robustness in computer vision for tasks including object detection [80, 82], image classification [19, 20, 33], and semantic segmentation [4, 12]. Beyond vision, adversarial attacks exist for natural language processing [14, 88, 91], reinforcement learning [25, 52, 63], and graph classification [92, 96]. While most of these works only expose theoretical vulnerabilities as the generated attacks are not physically realizable, recent studies show that real-world adversarial attacks are possible. The works of [5, 22, 46, 49, 56, 71, 82] generate real-world adversarial examples for the models in the vision domain. Compared to the wireless setting, there is less signal distortion and hardware imperfections in the vision domain. The authors in [50] attack voice assistant systems such as Alexa, but their attack generates loud guitar music which is easy to detect and defend against. [83] proposes the basic structure of robust adversarial attacks in vision domain, we extend this into wireless domain by modeling the effect of wireless transformations and further test them out in real-world rather than purely simulation. To the best of our knowledge, we are the first to consider real-world attacks in wireless systems.

ML-based Wireless Systems: Machine learning has been extensively used in different tasks in wireless systems including both sensing and communication. In sensing, ML has been leveraged for human motion sensing [3, 95], sleep monitoring [30, 51, 89], emotion detection [94], indoor positioning [1, 6, 15, 75], etc. In communication, ML is also widely used in MIMO systems [31, 45, 60], modulation and signal classification [54, 78], resource allocation and management, and MAC protocol design [13, 35, 86]. In this paper, we limit our analysis to physical layer ML-models.

Adversarial Attacks against Wireless Systems: Recent work has shown theoretical attacks [2, 7, 9, 17, 18, 23, 43, 68–70] on ML-based wireless systems. However, none of these are feasible in the real world as they consider unrealistic threat models such as no distortion exists during transmission or the availability of coordination between the base station and the attacker. The closest work to ours is [8], where the authors use generative models [26] to obtain universal perturbations. They demonstrate the attack on simulated data and are not feasible in the real-world because the attack model does not account for: (a) the effect of the wireless channel on adversarial noise, (b) the lack of time-synchronization between a client's and adversary's transmissions. Our work is the first to demonstrate real-world hardware-implemented adversarial attacks by explicitly incorporating robustness to real-world channel transformations and un-synchronized transmissions.

8 Concluding Discussion

We present RAFA, the first real-world adversarial attack design on machine learning-based wireless systems. Our results show that adversarial attacks are feasible in the real-world, in spite of channel distortions, hardware noise, and black-box assumptions. We conclude with some directions that future work may consider for expanding on our paper:

- **More Capable Adversary:** We consider a single antenna adversary and show its feasibility in conducting real world attack. A multi-antenna adversary has more degrees of freedom and can cause more damage. It also opens up new questions on synchronization between different antennas, the tradeoffs between antenna count, efficiency, etc.
- **Higher Layer Attacks:** We focus on physical layer ML systems. We envision future work will consider attacks at higher layers (e.g., MAC), which can explore new modalities such as frame injection attacks.
- **Robust Training and Other Defenses:** How do we train models that are not prone to adversarial attacks? We show it is feasible to defend, but can this be made faster and more robust? Adversarial training provides empirical robustness, can we provide formal guarantees on when a model does or does not work? Finally, can we design cross-layer defense mechanisms that are robust to attacks in the physical layer?

Acknowledgements – We are grateful to the Qualcomm Innovation Fellowship program and NSF RINGS Award 2148583 for supporting this work. We also thank the reviewers and our shepherd, Fadel Adib, for constructive feedback.

References

[1] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller. 3d tracking via body radio reflections. In *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, pages 317–329, 2014.

[2] A. Albaseer, B. S. Ciftler, and M. M. Abdallah. Performance evaluation of physical attacks against E2E autoencoder over rayleigh fading channel. In *proc. IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT*, pages 177–182. IEEE, 2020.

[3] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, 16(4):1996–2018, 2014.

[4] A. Arnab, O. Miksik, and P. H. Torr. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 888–897, 2018.

[5] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[6] R. Ayyalasomayajula, A. Arun, C. Wu, S. Sharma, A. R. Sethi, D. Vasisht, and D. Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[7] A. Bahramali, M. Nasr, A. Houmansadr, D. Goeckel, and D. Towsley. Robust adversarial attacks against dnn-based wireless communication systems. In *ACM Conference on Computer and Communications Security (CCS)*, pages 126–140. ACM, 2021.

[8] A. Bahramali, M. Nasr, A. Houmansadr, D. Goeckel, and D. Towsley. Robust adversarial attacks against dnn-based wireless communication systems. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 126–140, 2021.

[9] S. Bair, M. DelVecchio, B. Flowers, A. J. Michaels, and W. C. Headley. On the limitations of targeted adversarial evasion attacks against deep learning enabled modulation recognition. In *Proc. ACM Workshop on Wireless Security and Machine Learning, WiseML@WiSec*, pages 25–30. ACM, 2019.

[10] A. Bakshi, Y. Mao, K. Srinivasan, and S. Parthasarathy. Fast and efficient cross band channel prediction using machine learning. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[11] M. Balunovic and M. T. Vechev. Adversarial training and provable defenses: Bridging the gap. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[12] V. Besnier, A. Bursuc, D. Picard, and A. Briot. Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15701–15710, 2021.

[13] N. Z. binti Zubir, A. F. Ramli, and H. Basarudin. Optimization of wireless sensor networks mac protocols using machine learning; a survey. In *2017 International Conference on Engineering Technology and Technopreneurship (ICE2T)*, pages 1–5. IEEE, 2017.

[14] N. Boucher, I. Shumailov, R. Anderson, and N. Papernot. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004. IEEE, 2022.

[15] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan. A comparative study on machine learning algorithms for indoor positioning. In *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*, pages 1–8. IEEE, 2015.

[16] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE Computer Society, 2017.

[17] M. DelVecchio, V. Arndorfer, and W. C. Headley. Investigating a spectral deception loss metric for training machine learning-based evasion attacks. *CoRR*, abs/2005.13124, 2020.

[18] M. DelVecchio, B. Flowers, and W. C. Headley. Effects of forward error correction on communications aware evasion attacks. *CoRR*, abs/2005.13123, 2020.

[19] D. I. Dimitrov, G. Singh, T. Gehr, and M. T. Vechev. Provably robust adversarial examples. In *Proc. International Conference on Learning Representations, ICLR*. OpenReview.net, 2022.

[20] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu. Benchmarking adversarial robustness on image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331, 2020.

[21] ettus. USRP X310. https://www.ettus.com/all-products/x310-kit/.

[22] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[23] B. Flowers, R. M. Buehrer, and W. C. Headley. Communications aware adversarial residual networks for over the air evasion attacks. In *Proc. IEEE Military Communications Conference, MILCOM*, pages 133–140. IEEE, 2019.

[24] M.-G. Garcia and J. M. Páez-Borrallo. Tracking of time misalignments for ofdm systems in multipath fading channels. *IEEE Transactions on Consumer Electronics*, 48(4):982–989, 2002.

[25] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*, 2019.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020.

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

[28] B. Gopalakrishnan and N. Jindal. An analysis of pilot contamination on multi-user mimo cellular systems with many antennas. In *2011 IEEE 12th international workshop on signal processing advances in wireless communications*, pages 381–385. IEEE, 2011.

[29] F. Gringoli, M. Schulz, J. Link, and M. Hollick. Free your csi: A channel state information extraction platform for modern wi-fi chipsets. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, WiNTECH '19, page 21–28, 2019.

[30] Y. Gu, Y. Wang, Z. Liu, J. Liu, and J. Li. Sleepguardian: An rf-based healthcare system guarding your sleep from afar. *IEEE Network*, 34(2):164–171, 2020.

[31] D. He, C. Liu, T. Q. Quek, and H. Wang. Transmit antenna selection in mimo wiretap channels: A machine learning approach. *IEEE Wireless Communications Letters*, 7(4):634–637, 2018.

[32] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] H. Hirano, A. Minagi, and K. Takemoto. Universal adversarial attacks on deep neural networks for medical image classification. *BMC medical imaging*, 21(1):1–13, 2021.

[34] R. Jia, A. Raghunathan, K. Göksel, and P. Liang. Certified robustness to adversarial word substitutions. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proc. Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 4127–4140. Association for Computational Linguistics, 2019.

[35] S. Jog, Z. Liu, A. Franques, V. Fernando, S. Abadal, J. Torrellas, and H. Hassanieh. One protocol to rule them all: Wireless {Network-on-Chip} using deep reinforcement learning. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 973–989, 2021.

[36] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath. Pilot contamination problem in multi-cell tdd systems. In *2009 IEEE International Symposium on Information Theory*, pages 2184–2188. IEEE, 2009.

[37] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath. Pilot contamination and precoding in multi-cell tdd systems. *IEEE Transactions on Wireless Communications*, 10(8):2640–2651, 2011.

[38] S. Kagi and B. S. Mathapati. Localization in wireless sensor network using machine learning optimal trained deep neural network by parametric analysis. *Measurement: Sensors*, page 100427, 2022.

[39] I. Karmanov, F. G. Zanjani, I. Kadampot, S. Merlin, and D. Dijkman. Wicluster: Passive indoor 2d/3d positioning using wifi without precise labels. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2021.

[40] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[41] R. A. Kennedy, D. B. Ward, and T. D. Abhayapala. Nearfield beamforming using radial reciprocity. *IEEE Transactions on Signal Processing*, 47(1):33–40, 1999.

[42] B. Kim, Y. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus. Adversarial machine learning for nextg covert communications using multiple antennas. *Entropy*, 24(8):1047, 2022.

[43] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus. Channel-aware adversarial attacks against deep learning-based wireless signal classifiers. *IEEE Trans. Wirel. Commun.*, 21(6):3868–3880, 2022.

[44] B. Kim, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, and S. Ulukus. Adversarial attacks with multiple antennas against deep learning-based modulation classifiers. In *2020 IEEE Globecom Workshops (GC Wkshps*, pages 1–6. IEEE, 2020.

[45] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath. 5g mimo data for machine learning: Application to beam-selection using deep learning. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2018.

[46] A. Kurakin, I. Goodfellow, S. Bengio, et al. Adversarial examples in the physical world, 2016.

[47] L. Lan, G. Liao, J. Xu, Y. Zhang, and B. Liao. Transceive beamforming with accurate nulling in fda-mimo radar for imaging. *IEEE Transactions on Geoscience and Remote Sensing*, 58(6):4145–4159, 2020.

[48] J. Li, S. Qu, X. Li, J. Szurley, J. Z. Kolter, and F. Metze. Adversarial music: Real world audio adversary against wake-word detection system. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 11908–11918, 2019.

[49] J. Li, F. R. Schmidt, and J. Z. Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *Proc. International Conference on Machine Learning, ICML*, volume 97, pages 3896–3904, 2019.

[50] J. B. Li, S. Qu, X. Li, J. Szurley, J. Z. Kolter, and F. Metze. Adversarial music: Real world audio adversary against wake-word detection system. *arXiv preprint arXiv:1911.00126*, 2019.

[51] C.-T. Lin, M. Prasad, C.-H. Chung, D. Puthal, H. El-Sayed, S. Sankar, Y.-K. Wang, J. Singh, and A. K. Sangaiah. Iot-based wireless polysomnography intelligent system for sleep monitoring. *IEEE Access*, 6:405–414, 2017.

[52] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*, 2017.

[53] D. Liu, W. Ma, S. Shao, Y. Shen, and Y. Tang. Performance analysis of tdd reciprocity calibration for massive mu-mimo systems with zf beamforming. *IEEE Communications Letters*, 20(1):113–116, 2015.

[54] X. Liu, C. Zhao, P. Wang, Y. Zhang, and T. Yang. Blind modulation classification algorithm based on machine learning for spatially correlated mimo system. *IET Communications*, 11(7):1000–1007, 2017.

[55] Z. Liu, G. Singh, C. Xu, and D. Vasisht. Fire: enabling reciprocity for fdd mimo systems. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 628–641, 2021.

[56] B. Luo, Y. Liu, L. Wei, and Q. Xu. Towards imperceptible and robust adversarial example attacks against neural networks. In *Thirty-second aaai conference on artificial intelligence*, 2018.

[57] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[58] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[59] I. A. Najm, A. K. Hamoud, J. Lloret, and I. Bosch. Machine learning prediction approach to enhance congestion control in 5g iot environment. *Electronics*, 8(6):607, 2019.

[60] T. J. O'Shea, T. Erpek, and T. C. Clancy. Deep learning based mimo communications. *arXiv preprint arXiv:1707.07980*, 2017.

[61] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. *Advances in neural information processing systems*, 2017.

[62] P. S. Pati, S. S. Sahoo, D. Krishnaswamy, and R. Datta. A novel machine learning approach for link adaptation in 5g wireless networks. In *2020 2nd PhD Colloquium on Ethically Driven Innovation and Technology for Society (PhD EDITS)*, pages 1–2, 2020.

[63] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*, 2017.

[64] Qualcomm. 5G RF. https://www.qualcomm.com/news/releases/2021/02/qualcomm-announces-next-generation-5g-rf-front-end-solutions-featuring-use.

[65] Qualcomm. X70. https://www.qualcomm.com/news/releases/2022/02/new-snapdragon-x70-modem-rf-harnesses-worlds-first-5g-ai-processor-industry.

[66] K. Rehrl and S. Gröchenig. Evaluating localization accuracy of automated driving systems. *Sensors*, 21(17):5855, 2021.

[67] T. G. Reid, S. E. Houts, R. Cammarata, G. Mills, S. Agarwal, A. Vora, and G. Pandey. Localization requirements for autonomous vehicles. *arXiv preprint arXiv:1906.01061*, 2019.

[68] M. Sadeghi and E. G. Larsson. Adversarial attacks on deep-learning based radio signal classification. *CoRR*, abs/1808.07713, 2018.

[69] M. Sadeghi and E. G. Larsson. Adversarial attacks on deep-learning based radio signal classification. *IEEE Wireless Communications Letters*, 8(1):213–216, 2018.

[70] M. Sadeghi and E. G. Larsson. Physical adversarial attacks against end-to-end autoencoder communication systems. *IEEE Communications Letters*, 23(5):847–850, 2019.

[71] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1528–1540. ACM, 2016.

[72] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. T. Vechev. Fast and effective robustness certification. *NeurIPS*, 1(4):6, 2018.

[73] G. Singh, T. Gehr, M. Püschel, and M. Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–30, 2019.

[74] B. Sliwa, R. Falkenberg, T. Liebig, J. Pillmann, and C. Wietfeld. Machine learning based context-predictive car-to-cloud communication using multi-layer connectivity maps for upcoming 5g networks. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–7. IEEE, 2018.

[75] P. Sthapit, H.-S. Gang, and J.-Y. Pyun. Bluetooth based indoor positioning using machine learning algorithms. In *2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 206–212. IEEE, 2018.

[76] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[77] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[78] F. Wang, S. Huang, H. Wang, and C. Yang. Automatic modulation classification exploiting hybrid machine learning network. *mathematical Problems in engineering*, 2018, 2018.

[79] M. Wasilewska, H. Bogucka, and A. Kliks. Spectrum sensing and prediction for 5g radio. In *Big Data Technologies and Applications*, pages 176–194. Springer, 2020.

[80] X. Wei, S. Liang, N. Chen, and X. Cao. Transferable adversarial attacks for image and video object detection. *arXiv preprint arXiv:1811.12641*, 2018.

[81] E. Wong and J. Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In J. G. Dy and A. Krause, editors, *Proc. International Conference on Machine Learning, ICML*, volume 80, pages 5283–5292, 2018.

[82] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017.

[83] C. Xu and G. Singh. Robust universal adversarial perturbations. *CoRR*, abs/2206.10858, 2022.

[84] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 1129–1141, 2020.

[85] T. Xu, T. Zhou, J. Tian, J. Sang, and H. Hu. Intelligent spectrum sensing: When reinforcement learning meets automatic repeat sensing in 5g communications. *IEEE Wireless Communications*, 27(1):46–53, 2020.

[86] B. Yang, X. Cao, Z. Han, and L. Qian. A machine learning enabled mac framework for heterogeneous internet-of-things networks. *IEEE Transactions on Wireless Communications*, 18(7):3697–3712, 2019.

[87] R. Yang, J. Laurel, S. Misailovic, and G. Singh. Training certifiably robust neural networks against semantic perturbations. In *Proc. International Conference on Learning Representations, ICLR*. OpenReview.net, 2023.

[88] J. Y. Yoo and Y. Qi. Towards improving adversarial training of nlp models. *arXiv preprint arXiv:2109.00544*, 2021.

[89] B. Yu, Y. Wang, K. Niu, Y. Zeng, T. Gu, L. Wang, C. Guan, and D. Zhang. Wifi-sleep: sleep stage monitoring using commodity wi-fi devices. *IEEE internet of things journal*, 8(18):13900–13913, 2021.

[90] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, and C.-J. Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2020.

[91] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41, 2020.

[92] X. Zhang and M. Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275, 2020.

[93] Z. Zhang, M. Lee, and S. Choi. Deep-learning-based wi-fi indoor positioning system using continuous csi of trajectories. *Sensors*, 21(17):5776, 2021.

[94] M. Zhao, F. Adib, and D. Katabi. Emotion recognition using wireless signals. In *Proceedings of the 22nd annual international conference on mobile computing and networking*, pages 95–108, 2016.

[95] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi. Through-wall human pose estimation using radio signals. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[96] D. Zügner, O. Borchert, A. Akbarnejad, and S. Günnemann. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(5):1–31, 2020.