WiAdv: Practical and Robust Adversarial Attack against WiFi-based Gesture Recognition System

YUXUAN ZHOU, CSE, Hong Kong University of Science and Technology, China
HUANGXUN CHEN, Huawei Theory Lab, China
CHENYU HUANG, Tencent, China
QIAN ZHANG, CSE, Hong Kong University of Science and Technology, China

WiFi-based gesture recognition systems have attracted enormous interest owing to the non-intrusive of WiFi signals and the wide adoption of WiFi for communication. Despite boosted performance via integrating advanced deep neural network (DNN) classifiers, there lacks sufficient investigation on their security vulnerabilities, which are rooted in the open nature of the wireless medium and the inherent defects (e.g., adversarial attacks) of classifiers. To fill this gap, we aim to study adversarial attacks to DNN-powered WiFi-based gesture recognition to encourage proper countermeasures. We design WiAdv to construct physically realizable adversarial examples to fool these systems. WiAdv features a signal synthesis scheme to craft adversarial signals with desired motion features based on the fundamental principle of WiFi-based gesture recognition, and a black-box attack scheme to handle the inconsistency between the perturbation space and the input space of the classifier caused by the in-between non-differentiable processing modules. We realize and evaluate our attack strategies against a representative state-of-the-art system, Widar3.0 in realistic settings. The experimental results show that the adversarial wireless signals generated by WiAdv achieve over 70% attack success rate on average, and remain robust and effective across different physical settings. Our attack case study and analysis reveal the vulnerability of WiFi-based gesture recognition systems, and we hope WiAdv could help promote the improvement of the relevant systems.

CCS Concepts: • **Security and privacy** → **Domain-specific security and privacy architectures**; • **Computer systems organization** → *Neural networks.*

Additional Key Words and Phrases: Adversarial attack, Gesture recognition, Wireless sensing

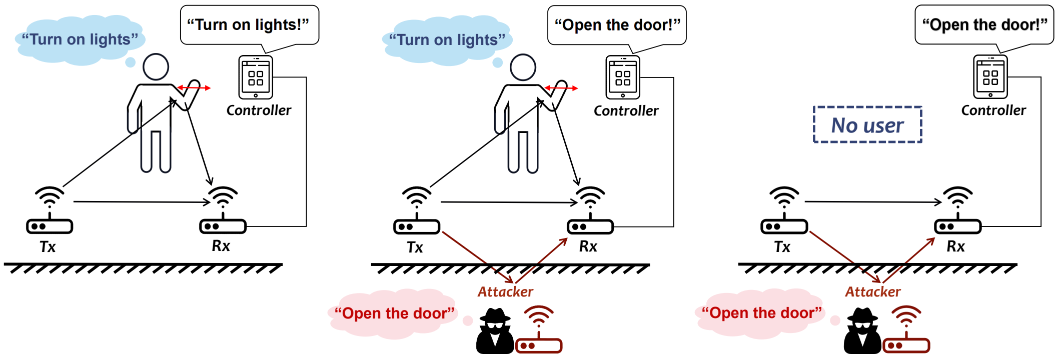
ACM Reference Format:
Yuxuan Zhou, Huangxun Chen, Chenyu Huang, and Qian Zhang. 2022. WiAdv: Practical and Robust Adversarial Attack against WiFi-based Gesture Recognition System. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 92 (June 2022), 25 pages. https://doi.org/10.1145/3534618

1 INTRODUCTION

In the ubiquitous intelligence era, WiFi, one of the most widely deployed communication techniques, has been recently repurposed as a powerful sensing medium. The non-intrusive nature of wireless signal and extensive adoption of WiFi chips in IoT devices make it appealing to enable WiFi-based gesture recognition for convenient interaction [1, 26, 37, 39, 40, 44], superior to introducing extra modalities including millimeter wave radar [21], camera [13, 19, 38] and sonar [18, 23, 41]. The recent studies of WiFi-based gesture recognition [17, 42, 44] have

Authors' addresses: Yuxuan Zhou, CSE, Hong Kong University of Science and Technology, China, yzhoudo@connect.ust.hk; Huangxun Chen, Huawei Theory Lab, China, chen.huangxun@huawei.com; Chenyu Huang, Tencent, China, hcyray@gmail.com; Qian Zhang, CSE, Hong Kong University of Science and Technology, China, qianzh@cse.ust.hk.

2474-9567/2022/6-ART92 $15.00
https://doi.org/10.1145/3534618



(a) The user controls the light with a gesture in smart home

(b) The adversary falsifies the command by relaying the signals

(c) The adversary spoofs the controller by relaying the signals

Fig. 1. Motivation scenarios of WiAdv: (a) depicts a normal use case of gesture recognition system. (b) illustrates the adversarial attack to the on-going gesture. (c) illustrates the adversarial attack without the on-going gesture.

established effective processing pipelines, and further integrated advanced deep neural networks (DNNs) to boost system performance.

However, the security vulnerabilities of these sensing systems lack systematical investigation. The most natural concern is whether it is possible to craft malicious wireless signals to fool WiFi sensing. This concern is rooted in the open nature of WiFi medium and the inherent defects (e.g., adversarial attacks) of classification techniques like DNNs. Most WiFi-based gesture recognition systems adopt a typical 3-stage pipeline: 1) collecting signals in the form of channel state information (CSI); 2) preprocessing raw signals to suppress interference and enhance motion-relevant features; 3) mapping extracted features to various gestures using classifiers like DNNs. The reflection signals are collected over-the-air, thus easily disturbed by malicious injection. To make matters worse, recent studies have shown the vulnerability of DNNs to adversarial attacks in the image domain [9, 15, 35], where carefully crafted images cause various DNN-based image classifiers to predict wrongly. Such potential vulnerabilities may be exploited to cause serious accidents. Take a smart home scenario for example as shown in Fig. 1. If the adversary fools the control system to accept injected malicious signals of 'open-the-door' or 'turn-on-the-gas' gestures, even if there is no user performing gesture, the users' personal and property safety will be severely threatened. Physical layer threats are not exclusive for wireless sensing. Actually, similar attacks have been investigated in wireless communication [32] and wireless localization [12, 20]. However, these existing works cannot forge "gesture" wireless signals. Thus, our major motivation is to fill the research gap to arouse awareness of the physical security issues in wireless sensing.

Although adversarial attacks against WiFi-based gesture recognition look viable in theory, we find it non-trivial to realize them in practice. First, the physical feasibility of WiFi signal perturbation is not as clear and straightforward as its image counterpart. It is intuitive to modify pixels to alter the content of an image, but it remains an unexplored problem to modify WiFi signals to craft motion-relevant features. Second, unlike image adversarial attacks, the perturbation space of WiFi adversarial attacks is inconsistent with the input space of DNN-powered classifier owing to multiple widely-used prepossessing modules on wireless signals. This reality makes it significantly harder to craft adversarial signals that remain effective after penetrating these modules. In addition, most processing modules are non-differentiable, preventing direct use of seminal gradient-based attacks in the image domain [9, 15]. Third, to achieve physically realizable attacks, it is indispensable to consider the dynamics of the ambient wireless environment and also the capability constraints of the victim and attacker devices. The crafted adversarial signals should be emitted by attack devices as desired; resist distortion by the complicated wireless environment to remain effective; be accepted by the victim devices as expected to strategically influence the system outputs.

Despite many challenges, this work finally validates the feasibility of physical adversarial attacks against WiFi-based gesture recognition systems. We design WiAdv to effectively construct physically realizable adversarial examples, which causes the victim system to misclassify the ongoing gesture. This work helps shed light on the underlying vulnerabilities of WiFi-based gesture recognition to encourage proper countermeasures.

Highlights of our original contributions in this paper are as follows. First, WiAdv features a signal synthesis scheme to craft adversarial WiFi signals with desired motion features. Our key insight is that the foundation of WiFi-based gesture recognition is to extract the dynamic multipath carrying the effect of users' gestures. Thus, we notice that state-of-the-art full-duplex devices can tactfully forward WiFi signals to mimic dynamic multipath. It would be indistinguishable between the artificial multipath and the natural ones from the perspective of WiFi receivers. Moreover, we characterize the relationship between the phases of signals and motion-relevant features based on analyzing the principle of WiFi sensing. Therefore, we bridge the physical perturbation space and our attack target of crafting adversarial "gestures" signals.

Second, Widav handles the difficulties caused by non-differentiable processing modules in the sensing pipeline. An intuitive solution would be to leverage differentiable operations to approximate non-differentiable modules [3] so that gradient-based attacks can be applied. However, we argue that this attack approach requires a white-box setting, *i.e.*, knowing the design details of sensing systems. Furthermore, it may work for attacking a specific system, but may not generalize well to others with various processing modules. This observation motivates us to design a black-box adversarial generation scheme to fit more practical attack scenarios. We design two different approaches, Constant Attack and Greedy Attack , to satisfy different attack constraints and expectations.

Third, we improve the practicality and robustness of our attack strategies by taking the ambient interference signals and processing jitters into consideration. Therefore, our attack strategies are immune to the dynamic impacts from the ambient wireless environment and have a higher probability of successfully fooling the sensing system. We conduct extensive experiments to evaluate our attack strategies against a representative state-of-the-art WiFi-based gesture recognition system, Widar3.0 [44] in realistic physical settings. Widar3.0 supports 6 gestures classification. For each type of gesture, we try to fool the system to misrecognize it as the other 5 gestures, *i.e.,* 30 source-target gesture pairs in total. Constant Attack successfully attacks 63.3% cases, while Greedy Attack handles 93.3% cases. We invite 15 volunteers to collect 543,500 CSI samples of 5,382 gestures in practical settings. Our experimental results demonstrate that our targeted attacks achieve an overall success rate of 73.64% and 70.35% for Constant Attack and Greedy Attack among their generated attack strategies respectively. The impact factor experiments, including tests on different rooms, attacker location, transmission power, *etc.*, validate the effectiveness and robustness of WiAdv. Besides the attack case study, we further analyze the vulnerability of WiFi-based gesture recognition systems revealed by WiAdv, and also discuss the potential defense schemes.

2 RELATED WORKS

WiFi-based Gesture Recognition Technique. Human gesture recognition has been extensively explored as an important component of human-computer interaction. Recently non-intrusive, ubiquitous, and privacy-preserving WiFi-based solutions have been developed rapidly. DNNs have been proven greatly successful in the image domain and leveraged heavily in other domains involving pattern matching. There is no exception in the domain of WiFi-based gesture recognition. Many recent works [17, 22, 44] leveraged DNN-based classifiers to improve gesture recognition performance. In particular, some works [17, 22] fed processed CSI signals (*e.g.*, after noise removal) to gesture classifiers. While others [44] first transformed CSIs into some distilled motion-relevant features, and then fed them into gesture classifiers. In general, these works adopt the sensing pipeline comprising CSI acquisition, preprocessing (*e.g.*, denoising, feature extracting, *etc.*), and classification.

Adversarial Attack. Adversarial attacks have been initially and widely explored in image domain [9, 15, 31, 35]. Some recent works start to explore adversarial examples in other domains such as natural language processing [30], LiDAR detection [8, 34, 36], speaker recognition [7, 11, 29], *etc.* Furthermore, recent works [11, 29, 36] explored the possibility to launch such attacks in the physical world. Our work aims to shed light on similar vulnerabilities in WiFi-based gesture recognition systems, and tries to seek physically realizable attack strategies. To the best of our knowledge, though prior work [24] has discussed the adversarial attack in radar-based gesture recognition, the feasibility of physical adversarial attack against WiFi-based systems has not been systematically investigated.

Wireless Physical Layer Attack. The wireless physical layer attack has been widely discussed in the wireless communication area [32]. There are many kinds of attacks including traffic analysis, eavesdropping, Denial of Service (DoS) attack, replay attack, masquerade attack, *etc.* These attacks mainly focus on obtaining private information and preventing normal service. There are also existing works investigating the physical security of wireless localization [12, 20]. They tamper the channel properties in order to increase the error of the localization systems. These works are different from our attack of crafting adversarial "gesture" signals to fool wireless sensing systems.

3 SYSTEM AND THREAT MODEL

In this section, we present the overview of victim systems, introduce the goal of our attack and illustrate the threat model.

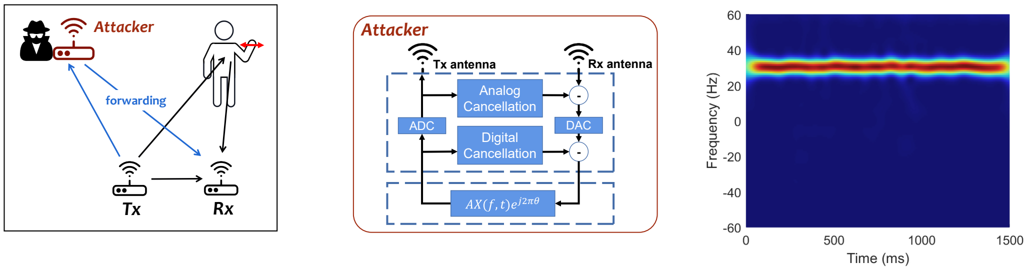
WiFi-based gesture recognition is to utilize WiFi signals to infer the ongoing gesture of the nearby user. Most systems aim to work with the control hub of the smart home as shown in Fig. 1(a), where the user performs gestures to the system in situ to get convenient services. Especially, our target victim systems are the DNN-powered WiFi-based gesture recognition systems. They take raw CSI data as input, extract some high-level features with some preprocessing modules, and leverage DNN technique to improve the accuracy of ongoing gesture prediction. They usually require multiple receivers (*e.g.*, 6 receivers in the typical setup of Widar3.0) and hundreds of CSI measurements per second (*e.g.*, 250 to 1000 packets per second in Widar3.0) to capture comprehensive signals about human activities for better recognition.

The goal of our attack has three folds. Firstly, our attack is a targeted adversarial attack. The adversary determines the desired gesture and then fools the victim system to output it. Secondly, our attack is a black-box attack. The adversary has no prior knowledge about the detail of the signal preprocessing and the DNN classifier of the victim system. Thirdly, the adversary is able to mount the attack either with (Fig. 1(b)) or without (Fig. 1(c)) the presence of the user. For example, triggering "turn on the gas" or "keep wheelchair moving" maliciously when the user is present while triggering "open the door" deliberately when the user is not at home. If the attack succeeds in fooling the victim system, the relevant users' property and personal safety will be at risk.

In our work, the adversary is equipped with a full-duplex device, which can modify and relay the received signals. In addition, the adversary can detect the start time of the gesture with the existing motion detection systems. When the target gesture is determined, the adversary prepares the attack strategies in advance. Then the adversary can bring the full-duplex device around the victim and launch the attacks behind the wall, as shown in Fig. 1(b)-(c).

4 SYSTEM DESIGN

This section describes the detailed design of WiAdv. We first explain the physical basis of WiFi-based gesture recognition and then introduce how to address the physical feasibility of WiFi-domain adversarial attacks. Then



(a) Crafting dynamic multipath while a user is performing a gesture

(b) Full-duplex attacker in Fig. 2(a)

(c) Crafted DFS at the receiver when f_D = 30 Hz

Fig. 2. Crafting dynamic multipath with a full-duplex device.

we describe our attack scheme to generate effective adversarial examples given the difficulties from limited perturbation space and various non-differentiable processing modules. Afterward, we propose improvement measures to address the robustness of attack strategy in practical attacks. At last, we integrate all components to summarize our attack process.

4.1 Preliminary: WiFi-based Gesture Recognition

WiFi-based sensing systems exploit channel state information (CSI) to infer surrounding dynamics. CSI represents the amplitude attenuation and phase change of WiFi signals caused by the environment between the WiFi transmitter and the receiver. The CSI of WiFi packets arriving at time t and at frequency f under the multipath phenomenon can be represented as follows:

$$H(f,t) = \sum_{l=1}^{L} \alpha_l(f,t) e^{-j2\pi f \tau_l(f,t)}, \tag{1}$$

where L is the number of multi-paths, α_l and τ_l are the complex attenuation and propagation delay of the l-th path. When a person performs a gesture near the transmitter and the receiver, the body reflection also contributes to the multipath effect, so that CSI measured at the receiver will carry the human gesture information, *e.g.*, location, velocity and *etc.* Doppler frequency spectrum (DFS) profile, which can be extracted from CSI, embodies most information of velocity distribution for gesture recognition. Without loss of generality, we use DFS to represent the dynamic information carried by CSI in our analysis. We can transform CSI in terms of DFS as follows [27]:

$$H(f,t) = H_s(f) + \sum_{l \in P_d} a_l(t) e^{j2\pi \int_{-\infty}^{t} f_{D_l}(u)du}, \tag{2}$$

where H_s is a constant that represents the sum of all static signals with zero DFS (*e.g.*, Line-of-Sight signals), and P_d is the set of dynamic signals with non-zero DFS (*e.g.*, signals reflected by the body limbs), f_{D_l} is the Doppler frequency shift of l-th reflection path.

4.2 Physical Basis of WiFi Adversarial Attack

In this section, we introduce how to address the challenge of crafting adversarial WiFi signals with desired motion features, and further maintaining their effectiveness across various environments.

4.2.1 *Generating Artificial Multipath.*

It is noticed that WiFi signals are open mediums in the air, and the foundation of WiFi-based gesture recognition is to extract the dynamic multipath carrying the effect of users' gestures. Therefore, a feasible approach for the attacker to fool these systems is to generate wireless signals on purpose to mimic dynamic multipath as the real reflection of a gesture as shown in Fig. 2(a). A state-of-the-art full-duplex transceiver can tactfully forward WiFi signals to achieve the above attack goal. The typical structure of full-duplex transceivers is shown in Fig. 2(b). It can receive and forward wireless signals simultaneously via feeding the transmitted signals back to the receiving chain for self-interference cancellation [6, 10, 28]. Each forwarded signal can be regarded as one emulated reflection path. Before forwarding, the attacker can manipulate the signals to fabricate dynamic reflection. As shown in Fig. 2(b), the parameters that can be controlled by the adversary are the amplitude A and phase θ added to the original received signal $X(f,t)$, which determine the fundamental physical perturbation space of adversarial attacks.

4.2.2 *Crafting Dynamic Reflection as Human Gestures.*

Next, we would like to further elaborate on how to control signal phases θ to mimic desired dynamic reflection as real gestures. Let us revisit the principle of WiFi-based gesture recognition. If we consider the signal of one reflection path, its CSI at time t_0 is

$$H(f,t_0) = \alpha_0 e^{-j2\pi f \tau_0}, \tag{3}$$

where α_0 is the attenuation and τ_0 is the propagation delay. If the path length changes at a speed of v due to a on-going gesture, after a short time period t, the path length change is $\Delta l_{path} = vt$, and the propagation delay change is $\Delta\tau = \frac{vt}{c}$, where c is light speed. Thus, the CSI of the signal becomes

$$H(f,t_0+t) = \alpha_0 e^{-j2\pi f(\tau_0+\frac{vt}{c})} = H(f,t_0)e^{-j2\pi f\frac{vt}{c}}, \tag{4}$$

As shown in Equation 4, the phase change rate caused by reflection $\frac{\mathrm{d}\theta}{\mathrm{d}t} = 2\pi f\frac{v}{c}$ is clearly associated with the representative motion-relevant features, Doppler frequency shift of the signal $f_D = f\frac{v}{c}$, *i.e.*, $\frac{\mathrm{d}\theta}{\mathrm{d}t} = 2\pi f_D$. Thus, if the adversary would like to mimic a real gesture that results in Doppler frequency shift f_D on the signals, *i.e.*, there is a peak in frequency f_D bin of the DFS profile derived by the WiFi receiver, it can set the change rate of the unwrapped phase of one emulated reflection path proportional to f_D as follows:

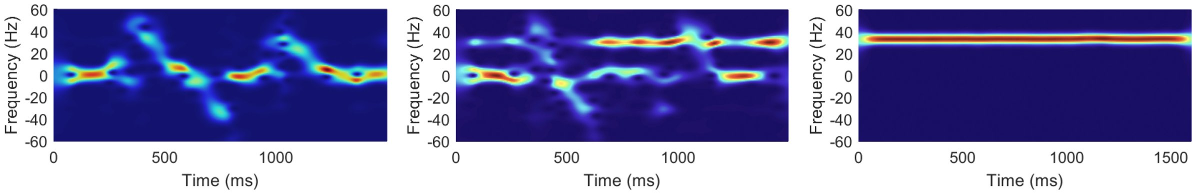
$$\theta^t_{i,\text{unwrap}} = 2\pi f_D t, i \in 0,...,N, \tag{5}$$

The practical phase θ^t_i will be wrapped to the range $[-\pi,\pi]$. Fig. 2(c) depicts the DFS of a sample of received signals when the attacker in Fig. 2(a) scenario keeps a constant phase change rate. These results show that the attacker can affect key motion features, *i.e.*, the DFS profiles of the WiFi receiver as desired with carefully designed perturbation on phases.

By bridging the signal phases θ in the physical perturbation space and motion-relevant features, *i.e.*, DFS, we address the feasibility of crafting adversarial WiFi signals with desired motion features. Thus, if an attacker has derived the DFS profiles of adversarial examples, it can configure the phase change rate of emulated reflection paths following the phase-DFS relationship to physically synthesize adversarial signals.

4.2.3 *Reinforcing Crafted Reflection.*

Besides crafting dynamic reflection, the attacker should make crafted adversarial signals remain effective under distortion by the various wireless environment. Thus, we further exploit the other part of the perturbation space, the amplitude A to make the crafted reflections more prominent than other dynamic reflections in the environment. To put it simply, we increase the transmission power of the attacker, *i.e.*, the amplitude A to make the crafted dynamic reflection the dominant component of DFS profiles. As shown in Fig. 3, we collect three sample DFS profiles when the user performs the "Pushing and Pulling" gesture, and the attacker sets different levels of transmission power. All three cases use the same phase perturbation shown in Fig. 2(c). When the attacker forwards the signal at relatively low power, the DFS profile shown in Fig. 3(a) still presents the motion features of "Pushing and Pulling". With increasing transmission power of the



(a) Low attacker transmit power (b) Median attacker transmit power (c) High attacker transmit power

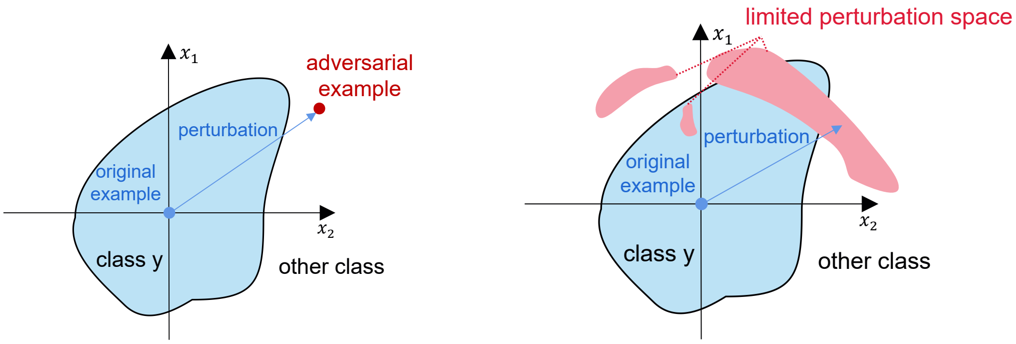
Fig. 3. DFS profiles under the same adversarial attack with different power on the benign "Pushing and Pulling" sample.

attacker, DFS profiles at the receiver are gradually dominated by the forwarded signals of the attacker as shown in Fig. 3(b). When the strength of the forwarded signal is much higher than the strength of the signal reflected from the user's body limbs, DFS profiles of the crafted reflections cover those of the natural ones, as shown in Fig. 3(c). We denote it as "DFS coverage" phenomenon in the following parts. Thus, we can enable effective crafted reflection across various environments to a large extent via keeping enough transmission power. The detailed evaluation of the attacker transmission power is shown in Section 5.8.2.

4.3 Adversarial Wireless Example Generation

In this section, we present our scheme to generate effective adversarial examples.

4.3.1 Perturbation Space Analysis. Since we have associated the DFS profiles with the physical perturbation space, the phase θ and amplitude A of the forwarded signals of the attacker in Section 4.2, all the "perturbation" and the "input" mentioned in the following parts refer to the perturbation on DFS for simplification. Thus, our objective is to obtain adversarial DFS profiles that fool the victim system to output our desired gesture type rather than the real on-going gesture type. The DFS profiles shown in Fig. 3 look analogous to images. However, we argue that there are significant differences between perturbation spaces of DFS profiles and images.



(a) Free perturbation space for image adversarial attack (b) Limited perturbation space for wireless adversarial attack

Fig. 4. Sketch of decision boundary in 2-D feature space.

As illustrated in Fig. 4, the basic principle of adversarial attack is to search for a proper perturbation that pushes the original sample (*e.g.*, an image or DFS/CSI of an on-going gesture) to cross the decision boundary of the classifier. In image-based adversarial attacks, each image pixel of an image can be perturbed freely and

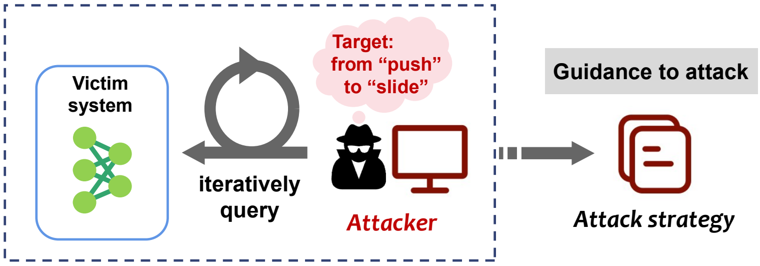


Fig. 5. Overview of attack strategy generation: For each source-target gesture pair, the adversary iteratively queries the victim system to fine-tune its adversarial strategy, *i.e.*, adversarial DFS profiles until obtaining a satisfying one. The adversary generally traverses all possible source-target gesture pairs for the victim system.

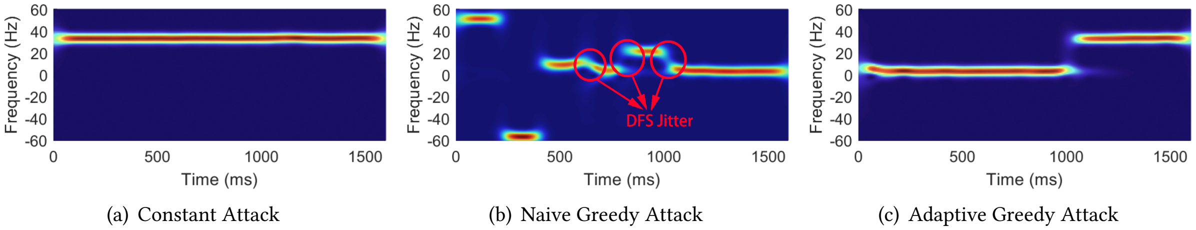
independently. In other words, the searchable perturbation space for image adversarial attacks is relatively large and continuous. However, the searchable perturbation space on DFS profiles is limited. The DFS profiles derived from Equation 2 contain three dimensions $F * T * N$, where F is the range of frequency in DFS (*i.e.*, the vertical axis of Fig. 3), T is the time span of a gesture (*i.e.*, the horizontal axis of Fig. 3), and N is the number of receivers of the victim system (*e.g.*, $N = 6$ in Widar3.0). It turns out that there are constraints in all three dimensions.

First, many WiFi-based gesture recognition systems only extract the major dynamic component for robust classification. Thus, it makes no sense to craft multiple Doppler shift peaks in a single timeslot t of an attack strategy. This greatly limits the perturbation space on frequency dimension F. Second, the perturbation space on frequency dimension F of adjacent timeslots is not totally independent. The essence of DFS is frequency variation for a short time as shown in Equation 2. Thus, in order to inject a desired DFS to the victim receiver, the adversary should maintain it for enough time before injecting the next desired DFS. This constraint further limits the whole perturbation space on the frequency-time $F * T$ dimension. Third, the last dimension N is limited by the hardware capability of the attack device. When the attacker is only equipped with a common omnidirectional antenna (adopted by our prototype), the DFS of all receivers will be almost the same. This is also the main reason why we cannot simply take an existing sample of the target gesture to launch substitution attacks.

Much limited DFS perturbation space as depicted in Fig. 4(b) makes it harder to search for effective attack strategies. However, the natural imperceptibly of wireless/WiFi signals gives us more flexibility on the dimension of perturbation amplitude than that of image adversarial attack. More fortunately, it is closely tied to the DFS coverage phenomenon illustrated in Section 4.2.3. Leveraging the strength dimension of the adversarial forwarded signal not only addresses the environment distortion issue mentioned in Section 4.2.3, but also helps push original signals to cross the decision boundary of the gesture classifier.

As the comparison shown in Fig. 3, the strong perturbation as in Fig. 3(c) results in more concentrated DFS profiles rather than the scattered ones as in Fig. 3(a-b). In a representative victim system, Widar3.0, owing to the sparsity requirement implied by the Equation 6, the concentrated DFS profiles as in Fig. 3(c) are more likely to induce dramatic changes on the BVP (20 × 20 matrix, direct input to the gesture classifier) to trigger misclassification.

4.3.2 Attack Formulation. Now we understand the searchable perturbation space on DFS profiles, the problem that remains now is how to obtain the desired adversarial DFS profiles to launch a successful adversarial attack. Many pieces of research on image adversarial attacks applied the gradient-based approaches to iteratively search in the perturbation space [9, 15, 35]. The adversary is required to have prior knowledge (the model structure and parameters) of the classification systems. However, in WiFi-based gesture recognition systems, there usually exist various non-differentiable preprocessing modules. The representative examples are the processing modules



(a) Constant Attack (b) Naive Greedy Attack (c) Adaptive Greedy Attack

Fig. 6. DFS profiles of adversarial examples from the source gesture "Pushing and Pulling" to the target gesture "Sweeping".

in Widar3.0 illustrated in Section 5.1. These non-differentiable modules limit the naive use of gradient-based adversarial attacks. Some prior works leveraged several gradient-free methods to handle the problem [3, 25], but their methods cannot generalize well to different kinds of processing modules.

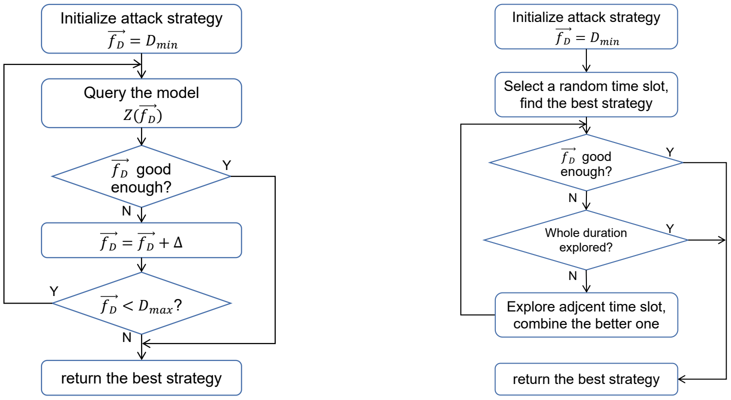
Consequently, we formulate the problem as an end-to-end black-box attack, taking both the limited perturbation space and various non-differentiable modules into consideration. We regard those non-differentiable modules as a part of the whole end-to-end recognition system. The adversary only knows the predicted labels and corresponding confidence of the victim system, having no access to the detailed structure/parameters of the classifier and preprocessing modules. Fig. 5 illustrate the general process of attack strategy generation. We define the *attack strategy* $\vec{f_D}$ as a series of Doppler frequency f_{D_i}, where i means the i-th time slot. Each attack strategy correlates to one pair of the source gesture x and the target gesture t. We denote Z as the whole end-to-end victim system consisting of the preprocessing modules and the DNN classifier, *i.e.*, $Z(\vec{f_D}) = x$ means that the DFS profile $\vec{f_D}$ is first processed, then fed into DNN models, and classified as gesture x. In addition, the attack strategy lasts the same duration as its associated source gesture x. The process shown in Fig. 5 is conducted offline to generate the effective attack strategy, then the adversary executes the strategy by following the principle of adversarial signal synthesis elaborated in Section 4.2 to trigger misclassification from source gesture x to targeted gesture t.

In practice, we argue that it is more efficient to adopt a hierarchical attack strategy generation scheme containing two discrete attack approaches. We can first search coarse-grained attack strategies in a more constrained perturbation space with lower complexity, less time cost, as well as lower requirement on synchronization (Constant Attack in Section 4.3.3). If the first attempt fails on certain source-target gesture pairs, we further conduct an advanced approach (Greedy Attack in Section 4.4.2) to search for more fine-grained attack strategies in a larger perturbation space. In the following, we introduce the details of two attack approaches.

4.3.3 *Constant Attack.*

Constant Attack means that the injected DFS remains the same through the whole gesture duration, *i.e.*, $f_{D_1} = f_{D_2} = ... = f_{D_n} = f$, denoted as $\vec{f_D} = f$, where f is a constant. A sample of generated attack strategy by Constant Attack is shown in Fig. 6(a). As shown in Fig. 7(a), the basic idea of the Constant Attack is to discretely search all optional DFS, and the time complexity depends on the resolution Δ. We also introduce an early-stop threshold τ, allowing the approach to return in advance if the current attack strategy is good enough.

The detailed procedure of the Constant Attack is as follows. The algorithm will simulate different DFS peaks throughout the whole duration. For each source-target gesture pair, the algorithm will search from the minimum value D_{min} to the maximum value D_{max} to find a proper Doppler frequency shift, where D_{min} and D_{max} are defined by the sensing system. If the adversary does not know these parameters of the victim system, a proper range of Doppler frequency can be estimated. This work assumes the frequency range is from -60 Hz to 60 Hz. In each iteration, the algorithm imitates $\vec{f_D} = f$ and queries the end-to-end system Z with $\vec{f_D}$. If this example is classified as the desired target gesture y, and the confidence value is larger than τ, the algorithm terminates and


Initialize attack strategy
$\overrightarrow{f_D} = D_{min}$
Query the model
$Z(\overrightarrow{f_D})$
$\overrightarrow{f_D}$ good enough?
Y
N
$\overrightarrow{f_D} = \overrightarrow{f_D} + \Delta$
Y
$\overrightarrow{f_D} < D_{max}$?
N
return the best strategy
Initialize attack strategy
$\overrightarrow{f_D} = D_{min}$
Select a random time slot, find the best strategy
$\overrightarrow{f_D}$ good enough?
Y
N
Whole duration explored?
Y
N
Explore adjcent time slot, combine the better one
return the best strategy

(a) Constant Attack

(b) Greedy Attack

Fig. 7. Flow graphs for basic ideas of Constant Attack and Greedy Attack .

returns $\vec{f_D}$. Otherwise, the algorithm sets $f = f + \Delta$ and enters the next iteration. If no satisfying attack strategy is found after traversing the whole range from D_{min} to D_{max}, the algorithm will output the best-ever attack strategy classified as targeted gesture y with the highest confidence. However, due to much limited perturbation space exploited, Constant Attack may return nothing, *i.e.*, we cannot find an effective attack strategy for certain source-target gesture pairs. The details of the failure cases in Constant Attack are presented in Section 5.5.

4.3.4 Naive Greedy Attack. To complement Constant Attack , we propose an advanced approach, Greedy Attack . Compared with Constant Attack , Greedy Attack searches attack strategies in a larger perturbation space by dividing the whole gesture duration in a time slot manner. To search efficiently, Greedy Attack applies the greedy search idea. As shown in Fig. 7(b), the basic idea of Greedy Attack is to find the best local strategy f_D in a time slot, which is equivalent to applying Constant Attack in that time slot, and then gradually extend to adjacent slots to construct a more fine-grained and powerful attack strategy. Fig. 6(b) shows a sample of the generated attack strategy by Greedy Attack .

The detailed procedure of the Greedy Attack is as follows. For each source-target gesture pair, we first divide the duration into multiple time slots with the same length L. Initially, the algorithm sets $\vec{f_D} = \epsilon$. The algorithm will start from a random time slot $\{i, ..., i + L - 1\}$and find the best Doppler frequency $\{f_{D_i} = ... = f_{D_{i+L-1}} = f_0\}$. Then the algorithm looks at adjacent time slots $\{i - L, ..., i - 1\}$ and $\{i + L, ..., i + 2L - 1\}$, searches for the best f_D in the time slots, keeping $\{f_{D_i} = ... = f_{D_{i+L-1}} = f_0\}$. Until now, the algorithm records two strategies: $\{f_{D_i}, ..., f_{D_{i+2L-1}}\}$ and $\{f_{D_{i-L}}, ..., f_{D_{i+L-1}}\}$. f_D of all other not-mentioned time slots are initialized as ϵ. Next, the algorithm compares these two strategies, selects the better one with higher confidence of target gesture, and records it. For example, the algorithm records the strategy $\{f_{D_{i-L}}, ..., f_{D_{i+L-1}}\}$. After that, the algorithm starts next iteration by searching updated adjacent time slots $\{i-2L, ..., i-L+1\}$ and $\{i+L, ..., i+2L-1\}$, keeping $\{f_{D_{i-L}}, ..., f_{D_{i+L-1}}\}$. The algorithm keeps searching until the whole duration is covered or a satisfying attack strategy (*i.e.*, the confidence of target gesture y is greater than τ) is found. Similarly, if no satisfying strategy is found, the algorithm will output the best-ever attack strategy classified as targeted gesture y with the highest confidence. If no strategy is classified as targeted gesture y, the adversary may restart Greedy Attack with another random starting time slot. It is worth emphasizing the importance of the default initialization ϵ. In order to suppress the influence of surrounding environments, the default DFS is set to ϵ when no explicit configuration is applied. Since we cannot create a

DFS peak at 0 Hz (f_D = 0 Hz implies stillness), we make ϵ as small as possible. This operation helps improve robustness based on the DFS coverage phenomenon.

Based on our evaluation, Naive Greedy Attack can generate attack strategies for 47% more source-target gesture pairs than Constant Attack in the offline stage of attack strategy generation, which validates the advanced capability of Naive Greedy Attack .

4.4 Robustness Improvement

Although Naive Greedy Attack achieves a high attack success rate in the offline generation stage, our physical experiments show that it exhibits lower performance than expected in the realistic setting. In this section, we analyze the cause leading to the instability of the attack strategies and propose an improvement measure for Naive Greedy Attack to generate more robust and practical attack strategies.

4.4.1 Limitation of Naive Greedy Attack . Based on our analysis, the instability of Naive Greedy Attack is mainly caused by the DFS jitter. We define *DFS peak* as the maximum value of DFS across all frequencies at a moment, which indicates the major dynamic component. We find that the DFS peak will be slightly diffused and connected to the DFS peaks in the adjacent time slot. This phenomenon is denoted as *DFS jitter.* The duration from 500 ms to 1000 ms in Fig. 6(b) shows a sample of DFS jitter. Based on our evaluation, DFS jitter brings uncertainty to the generated attack strategies and leads to downgraded performance. Through in-depth analysis, it turns out that the DFS jitter phenomenon occurs more frequently when there are multiple short pieces with consecutive DFS changes as shown in Fig. 6(b). Thus, we propose Adaptive Greedy Attack to reduce the appearance of DFS jitter and improve the stability of our attack.

4.4.2 Adaptive Greedy Attack. The basic idea of Adaptive Greedy Attack is to minimize the changing of DFS as far as possible to avoid the impact of DFS jitter. Adaptive Greedy Attack introduces a new variable, the division factor α to control the time slot length by $L = \frac{d}{\alpha}$, where d is the whole duration length of the source gesture. The Adaptive Greedy Attack initially starts with a long time slot length $L = \frac{d}{2}$, where $\alpha = 2$. And if the algorithm fails to find a satisfying strategy, it will restart with a shorter length L by increasing α by 1. The algorithm will stop when L decreases to a predefined minimum length, below which the generated strategy is entirely impractical. As a result, the Adaptive Greedy Attack will preferentially output the attack strategy with a longer time slot. It is noticed that Greedy Attack is equivalent to Constant Attack when we set $\alpha = 1$. A sample of the attack strategy generated by Adaptive Greedy Attack is presented in Fig. 6(c). We can find that the DFS peak only changes once in the whole duration and the DFS jitter occurs less compared to Naive Greedy Attack in Fig. 6(b).

In our comparative experiments, Naive Greedy Attack (short time slot length) and Adaptive Greedy Attack both generate attack strategies for most origin-target gesture pairs, but the attack success rates (ASR) in physical evaluation differ greatly. Naive Greedy Attack only achieves 26.19% ASR, while Adaptive Greedy Attack achieves 73.81% ASR. The difference in the robustness of the attack strategies causes the gap between these results. All the "Greedy Attack " mentioned in the following parts refer to the Adaptive Greedy Attack .

4.5 Putting All Things Together

We are now ready to introduce the complete process of the adversarial attack against WiFi-based gesture recognition systems. The ultimate goal of the adversary is to generate adversarial wireless signals so that the system output the wrong prediction that the adversary desires. The whole attack pipeline is depicted in Fig. 8 with two stages.

In the offline stage, the adversary needs to generate the attack strategy given a victim system. The adversary will traverse all possible source-target gesture pairs to prepare a stock of attack strategies for the usage in the

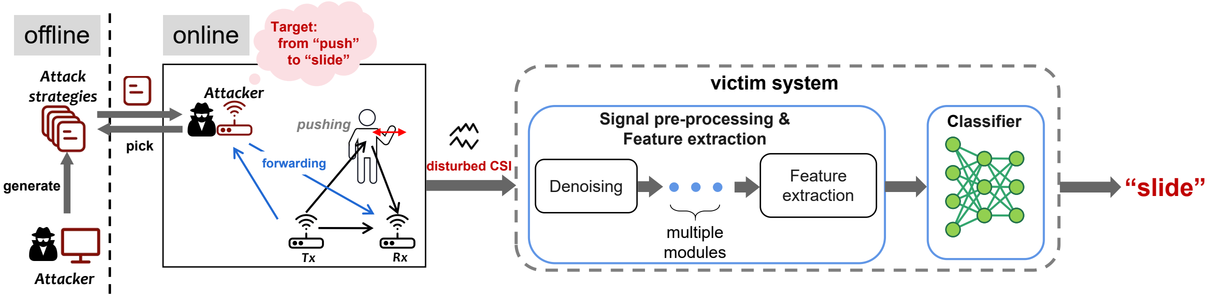


Fig. 8. WiAdv overview: (1) offline stage: Before launching the physical attack, the attacker traverse all possible source-target gesture pairs to prepare a stock of attack strategies against the victim systems (Fig. 5). (2) online stage: the attacker executes the selected attack strategy by emitting synthesized adversarial signals properly. The adversarial signals will be injected into the victim system and lead to the desired classification result by the attacker.

online stage. The adversary digitally synthesizes the generated DFS with an approximated gesture duration and verify the effectiveness of the attack strategies during the offline stage.

In the online stage, the adversary sets an attack target and retrieves relevant attack strategy from the output of the offline stage. Due to the "DFS coverage" illustrated in Section 4.2.3, WiAdv does not need to know the exact on-going gesture, but roughly estimates the gesture duration and start time to improve the attack stealthiness. The duration estimation could be simply the average of the typical duration of gestures supported by the victim system, or may be inferred from the user's past daily behavior since people's daily life is usually regular and highly predictable [4]. The estimation of start time could be enabled by the existing motion detection approaches [45]. For the "no people attack" scenario as shown in Fig. 1(c), the adversary has more flexibility on the start time and duration of attack. Next, the adversary follows the generated attack strategy and utilizes a full-duplex device to forward adversarial signals synthesized by the scheme elaborated in Section 4.2 to trigger misclassification from source gesture x to targeted gesture t. The reflection signal of the true gesture superposed by crafted reflection signals will finally be received by the victim system. The victim system is a DNN-powered WiFi-based gesture recognition system with a typical sensing pipeline. More specifically, the received data will pass through multiple preprocessing modules, including filtering, denoising, feature extraction, *etc*, and finally be fed into a DNN-based gesture classifier for prediction. Under adversarial attacks, the victim system is highly likely to predict the desired gesture of the attacker.

5 CASE STUDY: ATTACKING WIDAR3.0

In this section, we would like to illustrate how we validate WiAdv by applying WiAdv on a state-of-the-art WiFi-based gesture recognition system. We first briefly introduce the representative victim system, Widar3.0 [44] for evaluating our attack strategies. Then we illustrate the attack experiment setup, prototype implementation, and evaluation methodology. Finally, we elaborate on the attack results of WiAdv in comprehensive respects.

5.1 Brief Introduction of Victim System

We choose Widar3.0 as the representative victim system for two reasons. First, Widar3.0 achieves state-of-the-art performance on WiFi-based gesture recognition. Prior to them, many works generalized poorly across different environments, because they did not fully eliminate environment information unrelated to gestures. Thus, the gesture classifier trained in one environment usually has poor performance in another environment. Widar3.0 proposes a novel method to extract domain-independent features, body-coordinate velocity profiles (BVP), to boost generalization performance in unseen scenarios. BVP represents the distribution of signal power over velocity components on the body coordinates, following the relationship: $D^{(i)} = c^{(i)}A^{(i)}V$, where V is the BVP, $D^{(i)}$ is the DFS at the i-th receiver, $c^{(i)}$ is the scaling factor due to propagation loss, $A^{(i)}$ is the assignment matrix depending on the location of the receivers. They leverage the sparsity of BVP to formulate an optimization problem for accurate BVP estimation as follows:

$$min_V \sum_{i=1}^{M} |EMD(A^{(i)}V, D^{(i)})| + \eta||V||_0, \tag{6}$$

where M is the number of WiFi receivers, $EMD(\cdot, \cdot)$ is the Earth Mover's distance between two distributions. Widar3.0 feeds the estimated BVPs from Equation 6, each of which is in the form of a 20×20 matrix, to a CNN-based classifier for gesture recognition. To sum up, Widar3.0 follows a typical "signal collection-preprocessing-feature classification" pipeline and achieves great performance, which makes it an appropriate representative for WiFi-based gesture recognition systems. Second, their dataset and model are open-sourced, which significantly facilitates the reproducibility of their model for evaluation.

It is worth mentioning that selecting Widar3.0 to validate WiAdv does not mean that our work only targets at Widar3.0. The design rationale of WiAdv is based on the general "signal collection-preprocessing-feature classification" pipeline of the DNN-powered WiFi-based gesture recognition systems.

5.2 Victim System Setup

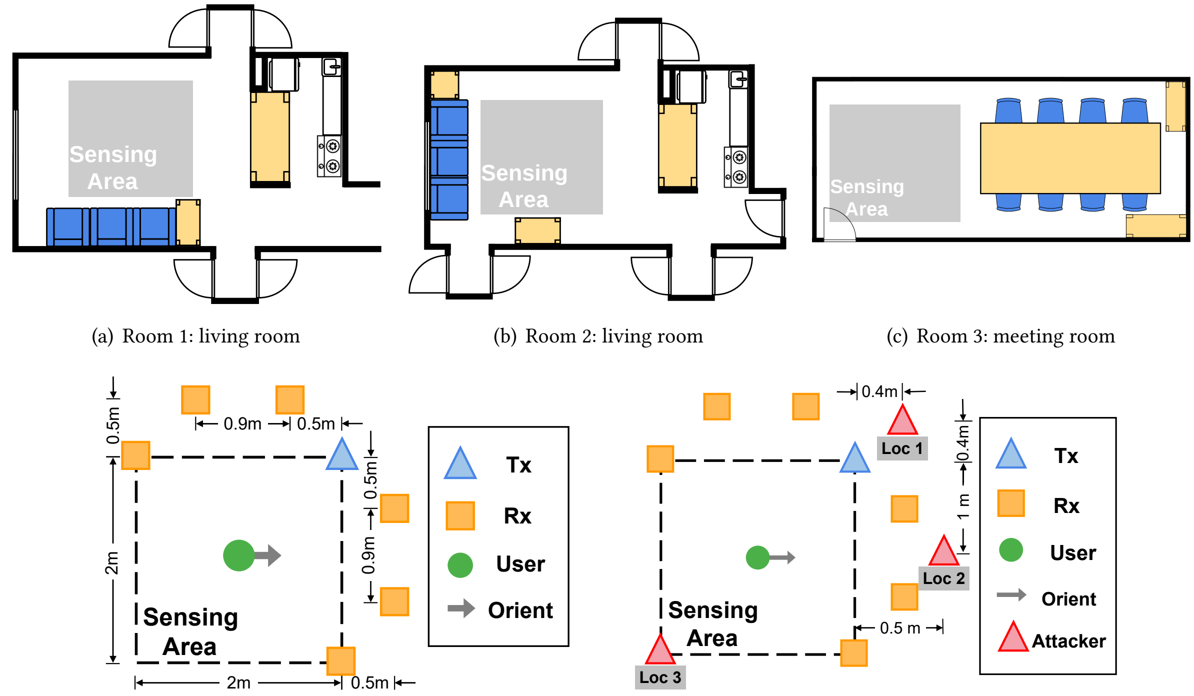
We replicate the Widar3.0 deployment setting in three indoor environments. The layouts of our evaluation environments are shown in Fig. 9(a-c): a living room with limited furniture like a sofa and desk, a living room with some additional storage boxes around the sensing area, and a narrow meeting room with desks and chairs. As shown in Fig.9(d), the device setup of Widar3.0 consists of one transmitter and six receivers. The locations of the transmitters and receivers are the same as the typical setup of Widar3.0. We use off-the-shelf mini-desktops equipped with the Intel 5300 wireless NIC as the receivers, the same as those used in Widar3.0. The receivers activate three antennas placed in a line with half-wavelength separations. They are set to monitor mode on channel 140 at 5.7 GHz. We install the Linux CSI Tools [16] on the receivers to collect CSI measurements for evaluation. The transmitter, *i.e.*, a single-antenna N210 USRP radio broadcasts 1000 Wi-Fi packets per second on channel 140 with 5dBm transmission power. All transceivers are put in 110 cm height so that the motion of users with different heights will be clearly detected.

We reproduce the gesture classification function of Widar3.0 with their open-source dataset, CSI preprocessing code in Matlab, and model training code in Python. To ensure that our reproduction achieves comparable performance, we collect extra gesture-relevant CSI data to further fine-tune the model. Our reproduced version achieves 99.4% accuracy, while the reference in Widar3.0 is 92.7%. Widar3.0 supports recognizing six gestures: "Pushing and Pulling", "Sweeping", "Clapping", "Sliding", "Drawing Circle", and "Drawing Zigzag".

5.3 Attack Prototype Implementation

Physically, we use another N210 USRP radio to emit generated adversarial wireless signals to launch attacks to Widar3.0. Due to the lack of full-duplex devices, we adopt an alternative solution with the equivalent effect. Specifically, we directly connect the attacker to the transmitter and keep them synchronized with a MIMO cable. The attacker emits the signals slightly behind the transmitter to emulate the forwarding behavior of a full-duplex device. We confirm the validity of this substitution in Section 5.8.4 and discuss the effect of processing time in the real full-duplex device in Section 5.8.3.

In our practice, we control them by a Personal Computer (PC) with a 3.59 GHz CPU and 16 GHz RAM. To comply with the realistic scenarios, we set the transmission power of the attacker as -5 dBm, 10 dBm lower than



(a) Room 1: living room
(b) Room 2: living room
(c) Room 3: meeting room
(d) Sensing area setup
(e) Attacker location

Fig. 9. Experiment setup: (a)-(c) show the layouts of our three evaluation environments. (d) shows the detailed setup of the sensing areas (gray squares in (a)-(c)), a replicated version of Widar3.0 setting. (e) shows the locations of the attacker tested in our evaluation.

that of the transmitter. In addition, we set the delayed emission time of the attacker against that of the transmitter as 100 ns based on the typical processing time of full-duplex devices.

We implement WiAdv's attack strategy generation in Python, then implement attack signal synthesis and modulation in MATLAB and GNURadio[14]. These attack signals are fed to the RF chain of the attacker USRP radio for emission. In our implementation, the minimum Doppler frequency D_{min}, the maximum Doppler frequency D_{max}, and the frequency resolution Δ are set to -60 Hz, 60 Hz, and 12 Hz, respectively during the attack strategy generation.

5.4 Evaluation Methodology

There are a total of 15 volunteers participating in our experiments, including 11 males and 4 females. The heights and ages of the volunteers vary from 157 to 184 cm and 22 to 25 years old, respectively. The volunteers are invited to do six gestures supported by Widar3.0 in three different environments. For evaluating the effect of one attack strategy, the volunteer will repeat the gestures three times. We collect around 543,500 CSI samples for 5,382 times of gesture performing. All experiments are approved by our IRB.

5.4.1 Evaluation Metrics. We adopt the attack success rate (ASR), *i.e.*, the ratio of successful attacks overall attack attempts to measure the attack effectiveness. A successful attack means that upon emitting adversarial signals, the victim system classifies them as the targeted gesture type desired by the attacker. Formally, ASR can be defined as: $ASR = \frac{N^{suc}_{A\rightarrow B}}{N^{all}_{A\rightarrow B}}$, where $N^{suc}_{A\rightarrow B}$ and $N^{all}_{A\rightarrow B}$ are the number of adversarial examples that are successfully classified as the target gesture B and the number of total adversarial examples that target at gesture B when the user performs gesture A, respectively.

5.4.2 Evaluation Goal. Our case study on attacking Widar3.0 would like to answer the following questions empirically:

(1) Can WiAdv successfully find effective adversarial examples against a state-of-the-art WiFi-based gesture recognition system? (Section 5.5)
(2) Can WiAdv successfully launch the attack in realistic scenarios using the physical radio to emit the adversarial signals? (Section 5.6)
(3) Can WiAdv maintain attack effectiveness against the diversity of environments and users? (Section 5.7)
(4) What requirements does WiAdv impose on the capability of the attacker, including the location of the attacker, power of emitted adversarial signals, processing delay, and multipath handling? (Section 5.8)

5.5 Effectiveness of Attack Strategy Generation

In this part, we directly retrieve gesture samples from the original Widar3.0 dataset and then enforce WiAdv on them. The resultant input to the gesture classifier of Widar3.0 would be the superposition of benign gesture samples and adversarial signals. We compare the gesture type output by the victim system and the targeted type of attack to evaluate the ability of WiAdv on finding effective adversarial examples against Widar3.0. For each gesture type, we test all attack cases with the other five types as the target type, *i.e.*, 30 kinds of source-target cases.

Table 1. Results of Constant Attack Strategy Generation.

Target Source	Push&pull	Sweep	Clap	Slide	Draw Circle	Draw Zigzag
Push&pull	-	✓		✓	✓	✓
Sweep		-	✓	✓	✓	✓
Clap		✓	-	✓		
Slide	✓	✓		-	✓	
Draw Circle		✓		✓	-	✓
Draw Zigzag		✓		✓	✓	-

Table 2. Results of Greedy Attack Strategy Generation.

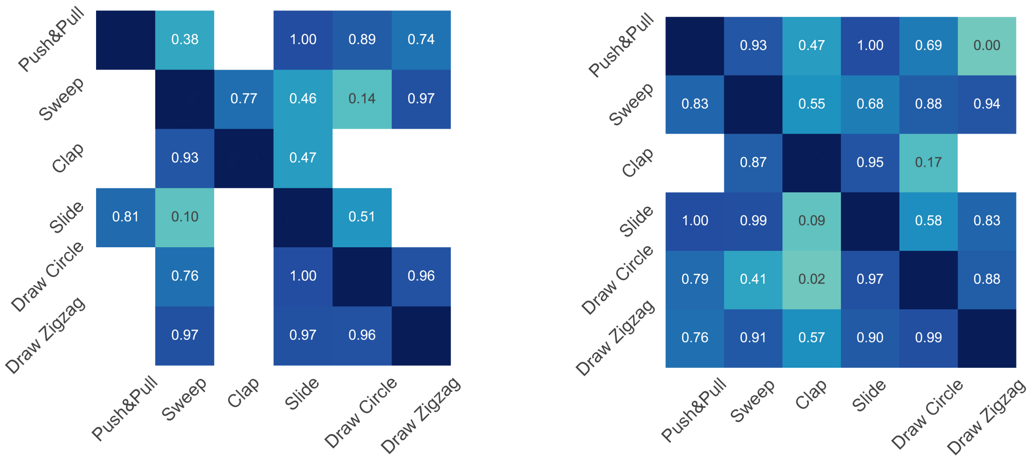
Target Source	Push&pull	Sweep	Clap	Slide	Draw Circle	Draw Zigzag
Push&pull	-	✓	✓	✓	✓	✓
Sweep	✓	-	✓	✓	✓	✓
Clap		✓	-	✓	✓	
Slide	✓	✓	✓	-	✓	✓
Draw Circle	✓	✓	✓	✓	-	✓
Draw Zigzag	✓	✓	✓	✓	✓	-

The experimental results are shown in Table 1 and Table 2 for Constant Attack and Greedy Attack respectively. The results show that the Constant Attack in WiAdv successfully fools the victim system in 19 out of 30 attack cases, while the Greedy Attack works on 28 out of 30 attack cases. These results are within the expected since the Greedy Attack searches in a larger perturbation space and are able to generate more kinds of adversarial signals.

For example, in the cases with "Pushing and Pulling" as the target gesture, Constant Attack has much poorer performance than Greedy Attack . The potential reason is "Pushing and Pulling" features a prominent motion direction changing pattern as shown in Fig. 3(a), and the Constant Attack cannot fabricate viable adversarial examples within its limited perturbation space. However, Greedy Attack pays more time cost. In our evaluation on query efficiency, Constant Attack query the victim model at most 20 times to generate an attack strategy, while Greedy Attack generates the attack strategy with about 53 queries on average. Thus, a hybrid scheme could be a practical option: the attacker initially tries Constant Attack as it costs less to generate a coarse-grained strategy; if the desired attack strategy cannot be successfully generated with high confidence, the adversary switches to advanced Greedy Attack to get a more fine-grained attack strategy. Limited unsuccessful cases for Greedy Attack occur on cases with "Clapping" as the source gesture. The relatively large difference between the source and target gestures could be the potential impact factor. The typical duration of the "Clapping" gesture (about 1 second) is generally shorter than other gestures (about 1.7 seconds on average), especially against "Pushing and Pulling" and "Draw Zigzag". Thus, the relevant attack cases are more challenging.

5.6 Physical Realizability of WiAdv

In this part, we proceed to evaluate the physical realizability of WiAdv. We fed the generated adversarial signals for 30 source-target gesture pairs in Section 5.5 to the physical radio and test them in the physical environments illustrated in Section 5.2. The results are summarized overall test trials on 15 volunteers across three environments and three attacker locations as shown in Fig. 9. The detailed results are shown in Fig. 10. In a nutshell, Constant Attack achieves average ASR 73.64% for 19 successful cases shown in Table 1, while Greedy Attack achieves average ASR 70.35% for 28 successful cases shown in Table 2. For the 19 common successful cases between Table 1 and Table 2, Greedy Attack achieve average ASR of 79.15%, around 5 point higher than Constant Attack .



(a) Constant Attack

(b) Greedy Attack

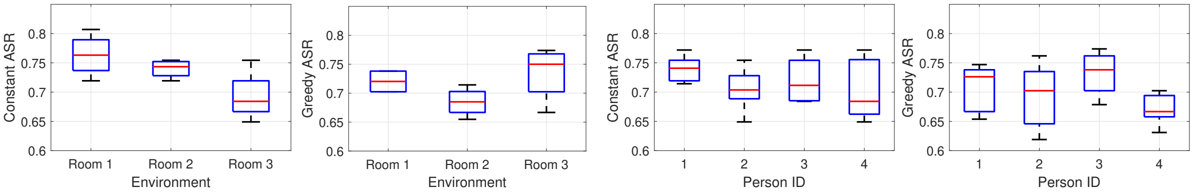
Fig. 10. Overall ASR of Constant Attack and Greedy Attack (blanks mean the lack of the attack strategy).

However, it is also noticed that Greedy Attack has poorer performance than Constant Attack on multiple cases. The potential reason could be that fine-grained attack strategies generated by the Greedy Attack may be more sensitive to disturbance from the environment. This also explains the slightly lower overall ASR of Greedy Attack compared with Constant Attack . For attack strategies targeting at the more challenging cases, *i.e.*, the 9 different successful cases between Table 2 and Table 1, it is more difficult for them to succeed in the physical environments. It is observed that in terms of target gestures, the average ASR of cases with "Clapping" as the target gesture is far lower than that of other cases. The relatively large difference in duration between "Clapping" and others gestures may contribute to the difficulty of pushing other gestures to "Clapping". We also notice other low-ASR cases, like "Pushing and Pulling" → "Drawing Zigzag". It is likely that the generated adversarial examples fall into local optimal not far away from the decision boundary of the classifier. Thus, when emitting them in physical environments, their attack effect is influenced by the randomness of the wireless environment.

Besides testing scenarios with on-going gestures (Fig. 1 (b)), we also evaluate our attack strategies in the scenario without on-going gestures (Fig. 1 (c)). The duration of adversarial examples is selected as typical gesture duration. We launched the Constant Attack and Greedy Attack over 150 times, and achieve the average ASR of 77.2% and 75.0% respectively.

5.7 Extrinsic Impact Factor Evaluation

In this part, we evaluate the attack effectiveness across various extrinsic impact factors beyond the control of the attacker. We evaluate two major extrinsic factors: environment and person variety. Ideally, it is desired that our attack strategies have stable performance against different factor settings. To evaluate the impact of a specific factor, we enforce the control variable method. We invite multiple volunteers to perform gestures in different settings of a single factor, remaining other factors unchanged.



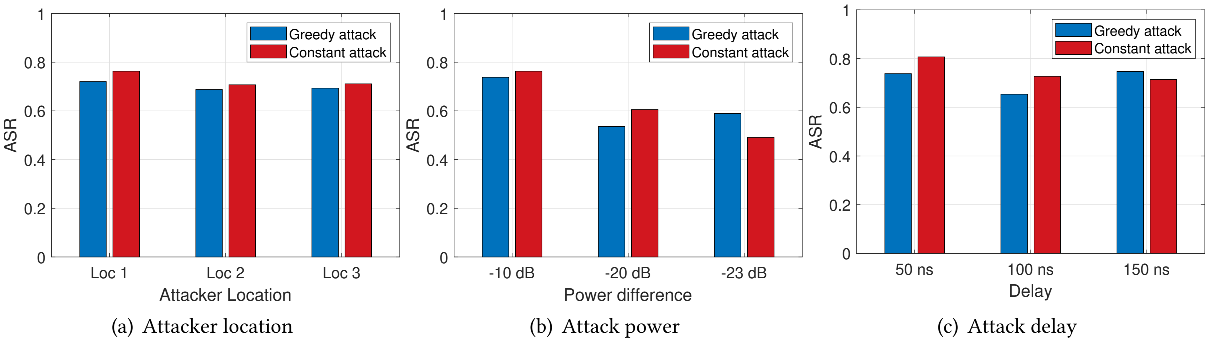
(a) ASR distributions for cross-environment

(b) ASR distributions for cross-person

Fig. 11. ASR distributions for extrinsic impact factor evaluation.

5.7.1 Impact of Environment. We evaluate our attack strategies on three different environments, whose layouts are shown in Fig. 9(a)-(c). There are 4 volunteers involved in this evaluation and the results of each environment are summarized and averaged across them. The ASR distributions of Constant Attack and Greedy Attack across three environments are shown in Fig. 11(a). Red solid lines indicate the median ASRs, blue rectangles indicate the ASRs from 25% percentile to 75% percentile, and black dotted lines indicate the minimum and the maximum ASRs. In a nutshell, our attack strategies maintain certain attack effectiveness against the diversity of environments. The average ASR across different environments achieve around 72%. The results of the third environment, *i.e.*, room 3 experience relatively large variance. We think the potential reason is that the third room has a much smaller space and is packed with furniture. Thus, it is more likely that reflection signals from nearby objects affect the adversarial examples.

5.7.2 Impact of Person Variety. The users with different figure scales and habits may impact the results of the victim system, Widar3.0. To evaluate our attack strategies on different people, we invite four representative volunteers (3 male and 1 female) based on their heights and weights. Their heights and weights are (178cm, 72kg), (180cm, 65kg), (158cm, 49kg) and (173cm, 78kg), respectively. They are asked to conduct experiments in different environments. The results of each volunteer are summarized and averaged across multiple environments. As shown in Fig. 11(b), the median ASRs remain over 65% across four volunteers.



(a) Attacker location (b) Attack power (c) Attack delay

Fig. 12. ASR for intrinsic impact factor evaluation.

5.8 Intrinsic Impact Factor Evaluation

In this part, we proceed to evaluate attack effectiveness across various intrinsic impact factors closely related to the attacker. We evaluate four major intrinsic factors: location, transmission power, full-duplex processing delay, and multipath handling capacity of the attacker. We adopt the same methodology as extrinsic impact factor evaluation.

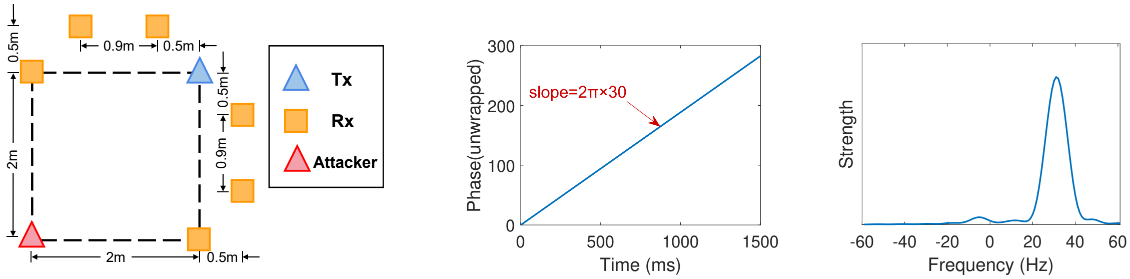
5.8.1 Impact of Attacker Location. We evaluate our attack strategies on three different locations of the attacker, as shown in Fig. 9(e). For the results of each location, we summarize and average test trials with various environments and volunteers. As depicted in Fig. 12(a), our attack strategies maintain effectiveness across different attacker locations, achieving around 70% mean ASRs. The attack cases on the first location achieve the best attack effect for both Greedy Attack and Constant Attack . We think it is mainly because the line-of-sight paths between location 1 and all receivers are unobstructed. If the attacker is located at location 2 or 3, the body trunk, and limbs of the victim user may obstruct the line-of-sight paths between the attacker and some receivers.

5.8.2 Impact of Attack Power. We evaluate our attack strategies with various transmission power difference between the attacker and the transmitter, *i.e.*, $P_{att} - P_{tx}$. We use the Rohde & Schwarz FSH6 spectrum analyzer to measure the transmission power. For each attack strategy, we evaluate it under three cases: -10 dB, -20 dB, and -23 dB. More specifically, the transmission power of the transmitter is fixed to 5 dBm, and the transmission power of the attacker is set to -5 dBm, -15 dBm, and -18 dBm, respectively. The ASR results are shown in Fig 12(b). When the power of the attacker descends to 20 dB lower than the transmitter, the average ASRs drop significantly. It is found that the adversarial DFS at 1 or 2 out of 6 receivers will not look as expected. When the power difference increases to 23 dB, the adversarial DFS at 3 or 4 receivers is distorted. When the power difference is further enlarged, the adversarial DFS at most receivers will mainly depend on the reflection of the ambient environments, rather than adversarial examples emitted by the attacker.

It is noticed that the attenuation of 2.4 GHz signal across a concrete wall with a thickness of 18 inches is about 18 dB [2] and the 5 GHz signal across a concrete wall with a thickness of 10 cm is about 22 dB [33]. Thus, it is feasible to launch our attacks in through-the-wall scenarios.

5.8.3 Impact of Attacker Delay. It takes 40 ns for the state-of-the-art full-duplex device [28] to process the received signals before further forwarding them. Some low-end full-duplex devices may need a longer processing time. In addition, estimating the start time of the gesture may induce extra processing delay. Thus, we evaluate our attack strategies under three processing delay settings: 50 ns, 100 ns, 150 ns. As shown in Fig. 12(c), our attacks can tolerate longer processing delay.

5.8.4 Impact of Multipath Effect. Besides the LOS signals from the transmitter, the full-duplex attacker also receives multipath signals reflected from the environment. Therefore, the attacker will not only relay the LOS signals from the transmitter but also all received multipath signals. In our emulated full-duplex setting, the multipath signals from the environment to the attacker are implicitly suppressed. To clarify and exclude the effect of the multipath signals on the attacker, we conduct a simulation following the setup shown in Fig. 13(a) using MATLAB WLAN Toolbox.



(a) Simulation Setup. The attacker is located at the position that can receives the most multipath signals.

(b) The phases of crafted adversarial DFS with $f_D = 30$ Hz.

(c) DFS profiles of all six receivers in the simulation.

Fig. 13. Simulation for evaluating the impact of multipath effect on the attacker.

In the simulation, all channels are configured as the multipath channel with 14 distinct paths, a 200 ns maximum delay, and a 30 ns root-mean-square delay spread. We craft the adversarial DFS with $f_D = 30$ Hz in the simulation as depicted in Fig. 13(b), and let the attacker send the corresponding adversarial signals. We collect received signals at all receivers and examine their DFS profiles. The resultant DFS at the receivers is shown in Fig. 13(c), the same for all six receivers. In the setting with a rich multipath effect, the DFS of the receivers matches the enforced adversarial signals. This indicates that the multipath effect at the attacker has minimal impact on the attack effectiveness. We think that no matter how many signal paths arrive at the attacker, its attack strategy has the same impact on all of them simultaneously. Thus, the adversarial DFS profiles at the receivers are relatively independent on the number of forwarded paths from the attacker. This validates the rationality of the emulated full-duplex setting in our evaluation.

6 SECURITY ANALYSIS AND DISCUSSION

In the previous section, we adopt Widar3.0 as a representative attack case to study and verify the prototype of WiAdv. Nevertheless, the core of WiAdv targets the general "signal collection ⟶ preprocessing ⟶ feature classification" pipeline of most WiFi-based gesture recognition systems. Thus, in this section, we would like to first analyze the generality of WiAdv, then analyze the vulnerability revealed by WiAdv and potential defense schemes, and finally discuss the potential attack improvement and future work.

6.1 Generality of WiAdv

In this part, we would like to discuss the capability of WiAdv to generalize beyond our evaluation case, Widar3.0. Specifically, we will analyze the WiAdv generality in handling the diversity of preprocessing, recognition backbone, gesture set, and practical system setting across various WiFi-based gesture recognition systems.

6.1.1 Preprocessing Diversity. It is noticed that some gesture recognition system [17] do not explicitly extract DFS profiles from CSI. However, WiFi-based sensing is bound to exploit the dynamic components to predict gestures, and our attack essentially crafts additional dynamic multipath signals like real gestures to generate adversarial examples. Thus, WiAdv can generalize to attack them in principle. For instance, the adversary can assume virtual multipath caused by the full-duplex device and then create the CSI data to query the system while preparing attack strategies. Some systems may not extract the primary dynamic components of the features like Widar3.0 does. As a result, the "DFS coverage" in Fig. 3 will not occur in these systems. However, the attacks will be even simpler in this case. For example, the adversary can create more than one peak in DFS, similar to Fig. 3(b). The constraint to the perturbation space can be relaxed accordingly.

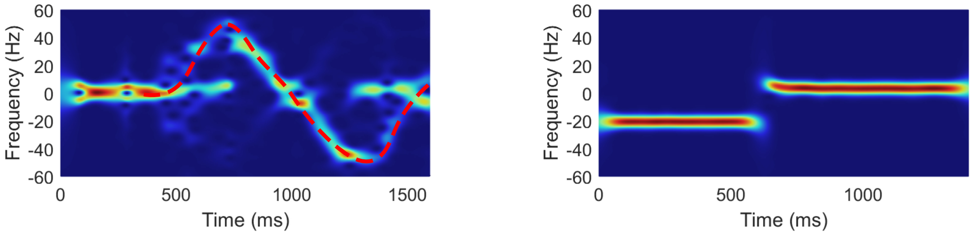
6.1.2 Recognition Backbone Diversity. In the existing gesture recognition systems, the recognition backbone falls into two categories: DNN-based and non-DNN ones. For the former, the vulnerability of DNN models against adversarial attacks has been empirically verified. In our case study against Widar3.0, we have trained two DNN models with different training sets, *e.g.*, data collected on different dates in Widar3.0 dataset. The attacks on both models achieve quite similar performance in adversarial example generation. For the latter, gesture recognition is bound to conduct some sort of pattern matching. For example, [26] do not leverage DNN-based classifier. It segments and encodes the DFS of a gesture depending on the positive and negative amplitude, and then classifies the DFS with the pattern matching method. Our black-box searching algorithm can be adapted to find adversarial examples fit for attacking such systems.

6.1.3 Gesture Set Diversity. Various systems could have different predefined gesture sets. In principle, the change of gesture set does not affect the effectiveness of WiAdv. But in practice, it is better to estimate the typical duration of gesture types in the targeted victim system, and take them into consideration in attack strategy generation in order to make the attack less detectable. For example, in the attack cases study against Widar3.0, we estimate the duration for six supported gestures as follows: 1.6 s for "Pushing and Pulling", 2.1 s for "Sweeping", 1 s for "Clapping", 1.4 s for "Sliding", 1.7 s for "Drawing Circle", and 1.8 s for "Drawing Zigzag". Volunteers are not aware of this information and can perform gestures based on their habits. However, these estimations can help reduce the difference between the emission duration of the attacker and the ongoing gesture duration to make the attack less detectable.

6.1.4 System Setting Diversity. WiAdv does not impose hard constraints on the setting of the victim system. However, it is worth mentioning that some practical adaption may be needed to apply WiAdv on a specific victim system. For example, Widar3.0 takes user location and orientation into consideration, it is better for the attacker to leverage the same estimation technique used in Widar3.0 to obtain this information. To improve the attack success rate, the adversary may leverage existing device localization methods [45] to localize transceivers of the victim system for adjusting proper attack power. In terms of hardware specification, WiAdv has no requirement on the victim system. However, WiAdv requires the adversary to receive and modify the signal at the same time, *e.g.*, full-duplex capability. Thus, existing low-cost COTS devices (*e.g.*, smartphone) are not qualified to act as the adversary to launch an attack. However, there is still a chance to make WiAdv deployable on COTS devices with the help of a reconfigurable intelligent surface (RIS) to passively reflect the signal and introduce additional phase shift, which is one of our future works.

6.2 Vulnerability of WiFi-based Gesture Recognition

In our attack case study against Widar3.0, it is interesting to note that the classification basis of the WiFi-based gesture recognition system may not completely coincide with the physical movement model. In Fig. 14, we show a comparison sample between benign gesture samples and adversarial samples enforced by WiAdv. Both of them are collected from the same receiver and classified as "Pushing and Pulling" by Widar3.0. However, the DFS profile shown in Fig. 14(a) seems to be more consistent with the physical model of the "Pushing and Pulling" gesture. In contrast, the physical interpretation of the adversarial DFS in Fig. 14(b) is that there is an object moving away from the transmitter-receiver link during the first 500 ms and staying still for the rest of the time. Unfortunately, such counter-intuitive adversarial examples successfully fool the victim system to output the desired gesture class of "Pushing and Pulling". It follows that these systems may classify DFS profiles with distinct movement characteristics as the same gesture, which leaves room for adversarial attacks. Further exploration is needed to investigate the gap between the gesture classification model and the physical movement model to improve the system's robustness against attacks.



(a) Benign DFS Profile

(b) Adversarial DFS Profile

Fig. 14. Benign and advesarial DFS profiles classified as "Pushing and Pulling" by Widar3.0.

Another interesting observation is that multiple attack strategies with the same target gesture may exhibit certain similarities in spite of different source gestures. For example, the attack strategies with "Sliding" as the target gesture are almost the same across different source gestures. This implies the potential universality of some attack strategies, which may increase the threats.

6.3 Defense Schemes

We discuss some countermeasures to WiAdv. There could be two types of defense methodology: blocking or detecting.

The blocking methodology tries to directly isolate the negative impacts of attackers. Following this idea, one intuitive method is to bound the WiFi signal with geofencing so that propagation from the transmitter to the attack is cut off or weakened. Accordingly, the attack signal will not affect the receivers, as well as the gesture recognition. Relevant approaches could be reducing the transmit power, or painting the walls with electromagnetic shielding paints. However, they are impractical and undesirable as both options will degrade network connectivity. Another defense method is the beamforming-based interference suppresses approach developed in wireless communication systems. This method will defend against a malicious attack based on the signal direction. However, in the case of WiFi-based gesture recognition, the signal direction of the user reflection is undetermined. What's more, it is flexible for the attacker to change the location as shown in the evaluation in Section 5.8.1. It is ineffective to adopt this method as the defense scheme.

The detecting methodology tries to identify the attackers based on some abnormal characteristics. Following this idea, defense methods for the replay attack could be an option, *e.g.*, abnormal transmission power detection method, and pause-and-detect method. However, WiAdv can be launched at a rarely low power comparable to the multipath signals, based on our evaluation in Fig. 12(b). As a result, detecting the abnormally high transmission power cannot defend WiAdv. The pause-and-detect method requires the transmitter to randomly pause transmission for a short time, and the receivers to report any adversarial transmission. As WiAdv only relays the transmitted signal and does not actively transmit any signal, the receivers only regard the adversarial signals as extra multipath signals.

The outlier detection module proposed in the extended Widar3.0 [43] could be another defense option. They empirically prove that the samples from predefined gestures are clustered in the feature space, while the outlier samples mostly occur at the edge zone of the clusters. This method is able to detect the outlier samples away from the central zone of the cluster without additional training or modifying the original model. However, our attack is to search for the attack strategies with high confidence in the feature space. When our attack targets at the extended Widar3.0, some gestures which are tough to attack in the original Widar3.0 may be affected, yet most of the attack strategies will still be generated and launched successfully. Our future work is to generate attack strategies with high confidence for more gestures.

In our attack case study, we find a viable defense scheme to detect WiAdv in certain cases. It is mainly due to the commercial off-the-shelf (COTS) omnidirectional antenna used in our evaluation. The omnidirectionally makes the attack signal have the same effect on nearby receivers. In other words, there is a notable feature from the perspective of the victim system: the DFS of all receivers are almost the same under such attack. Thus, to defend WiAdv in this setting, the victim system could insert an additional module to detect the similarity of all links after obtaining DFS. However, this defense method could be broken by selectively influencing parts of receivers, or crafting different DFS to different receivers with the help of beamforming and directional antennas, which is one of our future works.

6.4 Attack Improvement and Future Work

To achieve the goal of the black-box attack, we design a straightforward yet effective query feedback-based scheme to generate attack strategies. In terms of time cost to build the corpus of adversarial DFS profiles, Constant Attack takes at most 20 queries per attack strategy generation, while Greedy Attack takes about 53 queries on average. Each query takes almost the same time as the normal execution time of the target victim system. The relatively low query cost is partially due to the limited perturbation space on the wireless signal as illustrated in Section. 4.3.1.

We have also considered integrating advanced approaches to further reduce the query cost. It is found that most relevant techniques will estimate the gradients of the DNN models, and then adopt gradient-based adversarial attacks afterward. Unfortunately, this methodology is not suitable for our attack scenario. For example, in surrogate-model-based [3] methods, it would be tough to train a surrogate model to include the preprocessing modules and the original model. Gradient-estimation-based [5] black-box methods require continuous perturbation space to estimate the gradient, which is infeasible in the limited perturbation space on the wireless signal due to the characteristic of the signal processing modules. In addition, we are also concerned that introducing more fine-grained approaches may make the effectiveness of WiAdv dependent on certain signal-processing modules or DNN models, losing the generality to other systems. For example, if we tailor the attack for the optimization-based components in Widar3.0, our methods will lose the ability to attack those systems without optimization-based components.

In fact, we think simple yet effective black-box methods demonstrate the vulnerability of the DNN-based gesture recognition better. As shown in our evaluation, current black-box methods have achieved quite high ASR, revealing the inherent defects of the systems to urge countermeasure development.

In the meantime, we also think there is still potential space for us to design stronger black-box attacks in the future. On one hand, we can incorporate more advanced hardware capability into the attack. For example, by replacing the omnidirectional antenna with the directional antenna, the adversary can only affect the selected receivers. In addition, if there are multiple radios equipped with directional antennas, the adversary can impose different perturbations on different receivers. Thus, the adversary can design stronger black-box methods to exploit these capabilities. On the other hand, we would like to explore how to improve the adaptation flexibility of WiAdv when some prior knowledge of the victim is available. In this way, if the adversary only attempts to attack a certain system, it can easily adapt the system to become more dangerous, especially for the target system.

7 CONCLUSION

In this paper, we study the security issues of WiFi-based gesture recognition systems via designing WiAdv to construct physically realizable adversarial examples against them. WiAdv leverages the full-duplex transceiver to craft dynamic multipath reflection as human gestures and features two black-box attack approaches to generate robust and practical adversarial examples. Moreover, we reproduce a representative state-of-the-art WiFi-based gesture recognition system and evaluate our attack strategies on the system. Our experiment results show that our attack schemes not only achieve a high success rate but also maintain robustness across different physical settings. We hope this work can help shed light on the underlying vulnerabilities of WiFi-based gesture recognition to encourage proper countermeasures.

ACKNOWLEDGMENTS

This work is partially supported by the RGC under Contract CERG 16204418, 16203719, 16204820, and R8015. The authors would like to thank the anonymous editors and reviewers for the valuable comments and suggestions.

REFERENCES

[1] Heba Abdelnasser, Moustafa Youssef, and Khaled A. Harras. 2015. WiGest: A ubiquitous WiFi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 1472–1480. https://doi.org/10.1109/INFOCOM.2015.7218525

[2] Fadel Adib and Dina Katabi. 2013. See through Walls with WiFi!. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (Hong Kong, China) *(SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 75–86. https://doi.org/10.1145/2486001.2486039

[3] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 274–283. https://proceedings.mlr.press/v80/athalye18a.html

[4] Daniel Austin, Robin M Cross, Tamara Hayes, and Jeffrey Kaye. 2014. Regularity and predictability of human mobility in personal space. *PloS one* 9, 2 (2014), e90256.

[5] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2018. Practical Black-box Attacks on Deep Neural Networks using Efficient Query Mechanisms. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[6] Dinesh Bharadia and Sachin Katti. 2014. Fastforward: Fast and constructive full duplex relays. *ACM SIGCOMM Computer Communication Review* 44, 4 (2014), 199–210.

[7] Sourav Bhattacharya, Dionysis Manousakas, Alberto Gil CP Ramos, Stylianos I Venieris, Nicholas D Lane, and Cecilia Mascolo. 2020. Countering acoustic adversarial attacks in microphone-equipped smart home devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–24.

[8] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao. 2019. Adversarial Sensor Attack on LiDAR-Based Perception in Autonomous Driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 2267–2281. https://doi.org/10.1145/3319535.3339815

[9] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. 39–57. https://doi.org/10.1109/SP.2017.49

[10] Bo Chen, Yue Qiao, Ouyang Zhang, and Kannan Srinivasan. 2015. AirExpress: Enabling Seamless In-Band Wireless Multi-Hop Transmission. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (Paris, France) *(MobiCom '15)*. Association for Computing Machinery, New York, NY, USA, 566–577. https://doi.org/10.1145/2789168.2790114

[11] Guangke Chen, Sen Chenb, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. 2021. Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems. In *2021 IEEE Symposium on Security and Privacy (SP)*. 694–711. https://doi.org/10.1109/SP40001.2021.00004

[12] Y. Chen, W. Trappe, and R. P. Martin. 2007. Attack Detection in Wireless Localization. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*. 1964–1972. https://doi.org/10.1109/INFCOM.2007.228

[13] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. 2018. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8359–8367.

[14] GNU Radio Website. accessed February 2012. http://www.gnuradio.org

[15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. https://doi.org/10.48550/ARXIV.1412.6572

[16] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM computer communication review* 41, 1 (2011), 53–53.

[17] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 289–304. https://doi.org/10.1145/3241539.3241548

[18] Kaustubh Kalgaonkar and Bhiksha Raj. 2009. One-handed gesture recognition using ultrasonic Doppler sonar. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 1889–1892.

[19] Tianxing Li, Qiang Liu, and Xia Zhou. 2016. Practical human sensing in the light. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 71–84.

[20] Z. Li, W. Trappe, Y. Zhang, and Badri Nath. 2005. Robust statistical methods for securing wireless localization in sensor networks. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* 91–98. https://doi.org/10.1109/IPSN.2005.1440903

[21] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–19.

[22] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. 2018. Signfi: Sign language recognition using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–21.

[23] Rajalakshmi Nandakumar, Alex Takakuwa, Tadayoshi Kohno, and Shyamnath Gollakota. 2017. Covertband: Activity information leakage using music. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–24.

[24] Utku Ozbulak, Baptist Vandersmissen, Azarakhsh Jalalvand, Ivo Couckuyt, Arnout Van Messem, and Wesley De Neve. 2021. Investigating the significance of adversarial attacks and their relation to interpretability for radar-based human activity recognition systems. *Computer Vision and Image Understanding* 202 (2021), 103111.

[25] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.

[26] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-Home Gesture Recognition Using Wireless Signals. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking* (Miami, Florida, USA) *(MobiCom '13)*. Association for Computing Machinery, New York, NY, USA, 27–38. https://doi.org/10.1145/2500423.2500436

[27] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Kyle Jamieson. 2017. Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 1–10.

[28] Yue Qiao, Ouyang Zhang, Wenjie Zhou, Kannan Srinivasan, and Anish Arora. 2016. PhyCloak: Obfuscating Sensing from Communication Signals. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. USENIX Association, Santa Clara, CA, 685–699. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/qiao

[29] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. 2019. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 5231–5240. https://proceedings.mlr.press/v97/qin19a.html

[30] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1085–1097. https://doi.org/10.18653/v1/P19-1103

[31] Meng Shen, Zelin Liao, Liehuang Zhu, Ke Xu, and Xiaojiang Du. 2019. Vla: A practical visible light-based attack on face recognition systems in physical world. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–19.

[32] Yi-Sheng Shiu, Shih Yu Chang, Hsiao-Chun Wu, Scott C-H Huang, and Hsiao-Hwa Chen. 2011. Physical layer security in wireless networks: A tutorial. *IEEE wireless Communications* 18, 2 (2011), 66–74.

[33] William C Stone et al. 1997. Electromagnetic signal attenuation in construction materials. (1997).

[34] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. 2020. Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX

Association, 877–894. https://www.usenix.org/conference/usenixsecurity20/presentation/sun

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. *arXiv:1312.6199 [cs]* (Feb. 2014). http://arxiv.org/abs/1312.6199 arXiv: 1312.6199.

[36] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. 2020. Physically Realizable Adversarial Examples for LiDAR Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[37] Raghav H. Venkatnarayan, Griffin Page, and Muhammad Shahzad. 2018. Multi-User Gesture Recognition Using WiFi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services* (Munich, Germany) *(MobiSys '18).* Association for Computing Machinery, New York, NY, USA, 401–413. https://doi.org/10.1145/3210240.3210335

[38] Minsi Wang, Bingbing Ni, and Xiaokang Yang. 2017. Recurrent Modeling of Interaction Context for Collective Activity Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[39] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-Eyes: Device-Free Location-Oriented Activity Identification Using Fine-Grained WiFi Signatures. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking* (Maui, Hawaii, USA) *(MobiCom '14).* Association for Computing Machinery, New York, NY, USA, 617–628. https://doi.org/10.1145/2639108.2639143

[40] Wei Xi, Dong Huang, Kun Zhao, Yubo Yan, Yuanhang Cai, Rong Ma, and Deng Chen. 2015. Device-Free Human Activity Recognition Using CSI. In *Proceedings of the 1st Workshop on Context Sensing and Activity Recognition* (Seoul, South Korea) *(CSAR '15).* Association for Computing Machinery, New York, NY, USA, 31–36. https://doi.org/10.1145/2820716.2820727

[41] Koji Yatani and Khai N. Truong. 2012. BodyScope: A Wearable Acoustic Sensor for Activity Recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (Pittsburgh, Pennsylvania) *(UbiComp '12).* Association for Computing Machinery, New York, NY, USA, 341–350. https://doi.org/10.1145/2370216.2370269

[42] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. 2018. CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18).* Association for Computing Machinery, New York, NY, USA, 305–320. https://doi.org/10.1145/3241539.3241570

[43] Yi Zhang, Yue Zheng, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2021. Widar3.0: Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. https://doi.org/10.1109/TPAMI.2021.3105387

[44] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-effort cross-domain gesture recognition with Wi-Fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services.* 313–325.

[45] Yanzi Zhu, Zhujun Xiao, Yuxin Chen, Zhijing Li, Max Liu, Ben Y. Zhao, and Heather Zheng. 2020. Et Tu Alexa? When Commodity WiFi Devices Turn into Adversarial Motion Sensors. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020.* The Internet Society. https://www.ndss-symposium.org/ndss-paper/et-tu-alexa-when-commodity-wifi-devices-turn-into-adversarial-motion-sensors/