Time to Think the Security of WiFi-based Behavior Recognition Systems

Jianwei Liu, *Student Member, IEEE,* Yinghui He, *Student Member, IEEE,* Chaowei Xiao, *Member, IEEE,* Jinsong Han, *Senior Member, IEEE,* and Kui Ren, *Fellow, IEEE*

Abstract—Behavior recognition plays an essential role in numerous behavior-driven applications (*e.g.*, virtual reality and smart home) and even in the security-critical applications (*e.g.*, security surveillance and elder healthcare). Recently, WiFi-based behavior recognition (WBR) technique stands out among many behavior recognition techniques due to its advantages of being non-intrusive, device-free, and ubiquitous. However, existing WBR research mainly focuses on improving the recognition precision, while rarely studying the security aspects. In this paper, we reveal that WBR systems are vulnerable to manipulating physical signals. For instance, our observation shows that WiFi signals can be changed by jamming signals. By exploiting the vulnerability, we propose two approaches to generate physically online adversarial samples to perform untargeted attack and targeted attack, respectively. The effectiveness of these attacks are extensively evaluated over four real-world WBR systems. The experiment results show that our attack approaches can achieve 80% and 60% success rates for untargeted attack and targeted attack in physical world, respectively. We also show that our attack approaches can be generalized to other WiFi-based sensing applications, such as user authentication.

Index Terms—Behavior recognition, WiFi, Genetic algorithm, Adversarial sample.

1 Introduction

Behavior recognition is a key enabler for a wide range of essential human-centric applications (*e.g.*, virtual/augmented reality and smart home) and even the safety-critical applications (*e.g.*, healthcare and security surveillance). Traditional approaches utilize cameras [1], [2], [3], sonar [4], [5], [6], or wearable devices [7], [8] to capture behavior information, including gesture, activity, and the like. However, these approaches have their respective drawbacks, including the risk of visual privacy leakage, limited sensing range, and inconvenience inherent in using on-body sensor. Compared to these methods, WiFi-based solutions stand out by the advantages of being non-intrusive, contactless, device-free, and ubiquitous [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

Existing WiFi-based behavior recognition systems extract behavior-relevant features from WiFi signals by measuring signals' channel state information (CSI). Previous studies of CSI-based behavior recognition system (termed as CBRS) focus on either improving the recognition accuracy or enabling the CBRS's environment-adaption ability [11], [18], while lacking the comprehensive exploration for its security issues. In fact, the security problem of CBRS is of essence, because the recognition results are frequently related to the vital interests (*e.g.*, economic interest and life safety) of CBRS users. For instance, an adversary could manipulate certain wireless signals to mislead the decision of a fall detection system, threatening users' life safety. Even worse, in a smart home application, if an activity associated with turning the light on is falsely recognized as the activity of turning the gas on or opening the door, the user's life or property safety would be directly threatened.

Current CBRSes dominantly leverage machine learning-based methods for behavior recognition, but the emergence of adversarial samples severely threat the security of machine learning classifiers [21]. Thus, a natural concern arises: *Are these CBRSes vulnerable to practically physical adversarial samples? If so, to what extent?* Huang *et al.* [22] demonstrate that delicate cross-technology interference (CTI) could mislead the target CBRS to make wrong decision (*i.e.*, untargeted attack). However, they have not thoroughly explored the security risks in CBRS. In this paper, we also study the security issue of CBRSes under adversarial environments by designing physical online attacks. But we try to explore the possibility of causing more tremendous consequences (*i.e.*, both untargeted and targeted attacks) via simple and effective ways instead of CTI. For doing so, we first probe the feasibility of manipulating the input CSI samples of CBRSes in the real world. We find that jamming signal could induce CSI absence in normal CSI samples due to the regulation of the CSMA/CA protocol [23]. The CSMA/CA protocol is adopted by network interface cards (NICs) in CBRSes and NICs control the transmission of signals. Therefore, it is possible to perform effective attacks by emitting jamming signals (standards-compliant WiFi signals) towards the transmitter of the CBRS.

Although it is feasible to manipulate the input CSI, to achieve effective attacks is still difficult due to the following challenges: 1) *Stealthiness*: The attack should maintain the property of stealthiness so that the attack could not be easily

- *Jianwei Liu is with Zhejiang University, China, and ZJU-Hangzhou Global Scientific and Technological Innovation Center, China.*
- *Yinghui He is with Zhejiang University, China.*
- *Chaowei Xiao is with Nvidia Research and Arizona State University, USA.*
- *Jinsong Han is with School of Cyber Science and Technology, Zhejiang University, China, and Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, China.*
- *Kui Ren is with School of Cyber Science and Technology, Zhejiang University, China, Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, China, and Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, China.*

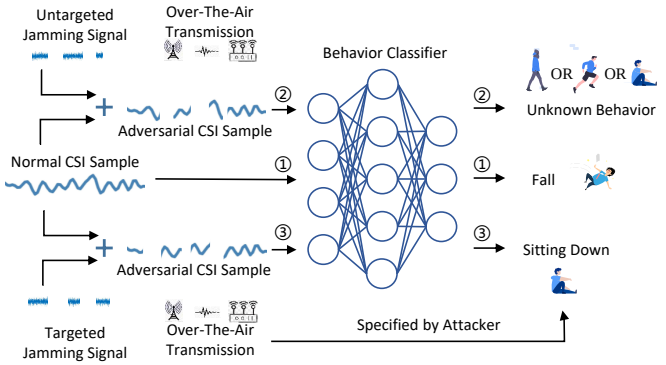


Fig. 1. Consequences of untargeted attack and targeted attack.

detected by the CBRS user; 2) *Disdifferentiability*: Existing targeted attack methods mainly rely on adding perturbations to normal samples. The process of the perturbation optimization is differentiable. However, jamming signal changes the CSI in CBRS by causing CSI absence instead of adding perturbation, and this process is non-differentiable; 3) *Robustness*: To launch effective targeted attacks, the attacker should immediately emit jamming signals as long as the user starts to perform a behavior; otherwise, the attack will not jam the specified position in the normal CSI sample, resulting in the degradation of the attack effectiveness. Nevertheless, it is difficult to synchronize the jamming signal in the physical world. Besides, the CSI sample of a specific behavior is not unique. Therefore, the jamming signal designed for a known CSI sample may be ineffective to the one collected during online attack.

By overcoming the above challenges, we propose two approaches to launch physical-world untargeted and targeted attacks against CBRSes, respectively. As shown in Fig. 1, the untargeted attack can lead the CBRS to recognize a behavior demonstrated by the user ('fall') as an unkown wrong one ('walk', 'run', or 'sit down'). The targeted attack can make the CBRS recognize the behavior ('fall') performed by the user as the one specified by the attacker ('sit down').

In detail, in order to overcome the first challenge, as our method exploits the CSI absence, we need to explore if the CSI absence also occurs in normal CSI samples. To this end, we first collect a large number of normal CSI samples and perform statistical analysis over them. We find that CSI absence exists in some normal CSI samples as well. In this case, as long as the degree of the CSI absence (*i.e.*, the number of absence times and the time length of each absence) caused by jamming signals is similar to that in normal CSI samples, the stealthiness of the attack can be guaranteed. Based on the result of this statistical analysis, we design untargeted and targeted attack approaches, in which we control the number of jamming times and the time length of each jamming to ensure sufficient stealthiness.

To address the second challenge, we first design an encoding scheme to encode jamming signals as bit sequences. With this scheme, we can leverage the genetic algorithm [24] consisting of three manipulation operations (duplication, crossover, and mutation) to optimize the jamming signal to generate targeted adversarial samples. As this optimization method does not require the differentiability, we can address the second challenge fundamentally.

To deal with the synchronization problem in the last challenge, we take the effect of the delay into account during optimization. We simulate the effect of delay by extending the fitness function in the genetic algorithm to a weight-based one. Moreover, to suppress the impact of the diversity of CSI samples, we introduce multiple CSI samples for each behavior when calculating the fitness score. Such countermeasure can help bit sequences improve their adaption abilities to the differences among different CSI samples.

In the evaluation part, we conduct comprehensive experiments on four CBRSes in real environments to study the effectiveness of our attack approaches. 17 volunteers are invited to collect both normal CSI samples and adversarial ones. The experiment results show that an attacker is able to achieve over 80% success rates in untargeted attacks. The success rate for targeted attack can reach 60%. The studies under harsh environments demonstrate that our attack approaches are still effective under non-light-of-sight (NLOS), occluded, and black-box attack scenarios.

In summary, our contributions are as follows:

- We study the security issues of existing CBRSes in the physical world. To our best knowledge, we are the first to achieve both untargeted attack and targeted attack in CBRSes physically.
- We conduct comprehensive evaluation over four CBRSes. The results demonstrate that an attacker can achieve over 80% and 60% success rates on untargeted and targeted attacks, respectively.
- We show that our attack approaches can be easily generalized to other WiFi-based sensing applications, such as user authentication. Moreover, We propose three ways to mitigate the harmfulness of the attacks.

2 Background and Attack Feasibility

We start this section by introducing some background knowledge on our attack target, *i.e.*, CBRS. Then, we describe the CSMA/CA protocol adopted by WiFi NICs for collision avoidance. The feature of this protocol enables us to change normal CSI in CBRS. Next, we formulate the adversarial environments in CBRSes. At last, we present the threat model.

2.1 CSI-based Behavior Recognition

A CBRS usually contains two modules, *i.e.*, CSI acquisition and learning-based behavior classification [10]. Below, we introduce each module elaborately.

CSI acquisition: In a CBRS, users obtain behavior information by measuring CSI from WiFi signals. Since CSI describes how the signal experiences power attenuation and phase shift caused by human behavior, it can record abundant behavior information. Taking a CBRS with a transmitter and a receiver as an example, the transmitted signal s_{tx} is reflected/absorbed by human body and becomes s_{rx} at the receiver end. Then, the CSI H is estimated using known s_{tx} and s_{rx}. Since CBRS transmits signals with a unit of packet and a behavior usually takes a period of time, a behavior is recorded by a CSI sample containing the CSI of all packets transmitted during this period [25]. Therefore, a CSI sample has t rows and f columns of CSI values, where

t is the number of packets and f is the number of used frequency. The CSI sample will be further processed in the next module.

Learning-based behavior classification: This module operates in two steps: *feature extraction* and *behavior classification*. In *feature extraction*, the CSI sample H extracted from the prior module first goes through some preprocesses (*e.g.*, low-pass filtering and interpolation [17], [19]). Then, an extraction method is applied to the preprocessed H to get a feature vector x. Without loss of generality, we use $f_{ext}(\cdot)$ to represent the whole feature extraction process: $x = f_{ext}(H)$. In the second step, a machine learning classifier $F_w(\cdot)$ parameterized by w is built to map the feature vector x to the probabilities of a set of labels. Each label corresponds to a category of behavior. The label that has the largest probability is the prediction result of $F_w(\cdot)$: $y = F_w(f_{ext}(H)) = F_w(x)$, where y is the predicted behavior label of x. To train the classifier, a batch of labeled CSI samples (*i.e.*, training set) is collected and the prediction error rate between the prediction label and ground-truth label is minimized. Once being well trained, the classifier can be used to predict the labels of unseen CSI samples, achieving the goal of behavior recognition.

2.2 CSMA/CA Protocol and CSI Absence

CSMA/CA protocol: NICs conform to the IEEE 802.11 a/b/g/n/ac/ax communication standard [26]. In these standards, CSMA/CA protocol is adopted to avoid collisions among signals at the same transmission channel but from different transmitters (each region in the world is allowed to use a specific number of channels [27] and each channel has f frequency). There are two main anti-collision mechanisms used by CSMA/CA protocol: *carrier sensing* and *collision avoidance*. In a WiFi signal transmission task, the *carrier sensing* mechanism works at first. It lets the transmitter listen to the shared medium (*e.g.*, WiFi signals in the wireless network) to determine whether another transmitter is transmitting signals at the same channel or not. If the transmitter detects that the signal power of the same channel in the shared medium is larger than a threshold, the *collision avoidance* mechanism will stop the transmitter transmitting packets and wait for a period of time. After that, the transmitter will repeat the "*carrier sensing*"-"*collision avoidance*" loop until the shared medium is detected clear, *i.e.*, the sensed power of the signal at the same channel is smaller than the threshold. In the transmission process, the *carrier sensing* mechanism keeps working to guarantee that the transmitter stops transmitting once collision occurs in the shared medium.

CSI absence: As mentioned in Section 2.1, each CSI sample is composed of CSI values of multiple packets over a period of time. In a CBRS, the time interval between any two consecutive packets approximates a constant value, *i.e.*, the transmitter sends packets at equal time intervals. In this way, each CSI sample can stably record the information of behavior. Suppose that the i_{th} transmission channel is used, the transmission rate is 100 packets per second, and each behavior continues for two seconds, then each CSI sample H should have dimensionality of (t, f). Ideally, t equals to 200 (100 packets/s $\times$ 2s). However, once we use another transmitter (attacker) to continuously emit signals (termed as jamming signals) at the i_{th} channel towards the CBRS's transmitter (victim), the aforementioned CSMA/CA protocol will stop the transmission of the victim transmitter. The victim transmitter will wait until the attacker transmitter stops the jamming. In this case, $t < 200$. That is, the number of rows in attacked CSI sample H' is less than 200, which means that some rows of the normal CSI sample are absent. This CSI absence caused by jamming signals makes the attack feasible as it manipulates H to H' ($H' \neq H$). In the remainder of this paper, the jamming signal is denoted by s_j. The impact of the jamming signal to H is denoted as $J(\cdot)$ and we have $H' = J(H, s_j)$.

2.3 Behavior Recognition in Adversarial Environments

Given a classifier $F_w(\cdot)$, a feature vector x and its label y, an adversarial attacker launches an attack by generating an adversarial sample x', so that $F_w(x') \neq y$ (untargeted attack) or $F_w(x') = y'$ (targeted attack), in which y' is a targeted label. Prior works [21], [28] have shown that the targeted attack can be achieved by generating an adversarial perturbation by optimizing the following objective function:

$$\min \quad ||x - x'||_p, \quad \text{s.t.} \quad F_w(x') = y' \text{ and } x' \in X, \qquad (1)$$

where $F_w(x') = y'$ is the attack goal and $x' \in X$ means that the generated adversarial sample x' is in a valid set. Then, an optimization algorithm is leveraged to generate the perturbation. In a CBRS, the adversarial perturbation is indeed the jamming signal s_j. The objective function can be re-written as follows:

$$\min \quad ||f_{ext}(H) - f_{ext}(J(H, s_j))||, \qquad (2a)$$
$$\text{s.t.} \quad F_w(f_{ext}(J(H, s_j))) = y' \text{ and } J(H, s_j) \in X. \qquad (2b)$$

In our attack scenario, as the jamming effect $J(H, s_j)$ is non-differentiable, we leverage the genetic algorithm to achieve the optimization objective.

2.4 Threat Model

Untargeted threat model: Untargeted attack attempts to fool the CBRS to output a false behavior label, which is not the one that the user demonstrated. In this threat model, the attacker does not need to have any prior knowledge about the CBRS. This model minimizes the constraints on the attacker.

Targeted threat model: Targeted attack aims to mislead the CBRS to output a behavior label that is specified by the attacker. For the targeted attack, we have the following assumptions: 1) The attacker can detect when the CBRS user starts to perform an activity. This can be achieved by using WiFi-based behavior detection methods [29]. Under this assumption, the attacker can immediately launch attacking signals once the victim demonstrates an activity. 2) We assume the targeted attack as a grey-box one, *i.e.*, the attacker only knows the architecture of the target behavior classifier. This is very nature as the classifier details are public in the existing literature on CBRS [15], [16], [17], [18], [19], [20]. Note that the attacker does not need to know the specific parameters of the target classifier.

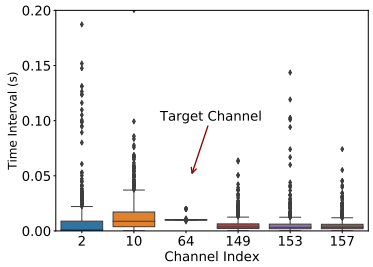


Fig. 2. Time interval distributions of CBRS and communication channels.

Threat scenario. Both untargeted and targeted attacks can be launched in any scenario the target CBRS is deployed in. Particularly, to achieve the most effective targeted attack, the attacker is required to simulate the context (including the surrounding environment and the placement of transceivers) of the target CBRS. Such a context can be easily known when the CBRS is a public application, *e.g.*, patient monitoring in hospitals and elderly monitoring in nursing homes. For some private places like personal residence, the context could also be observed through windows [30]. Since our attack approaches can be realized in both public and private places, they pose real threats to existing CBRSes.

3 Attack Preparation

Performing attacks against a CBRS should simultaneously meet the following four requirements. 1) Attacker Transmitter Selection: the adversary transmitter should be able to cover all WiFi transmission channels. 2) Target Channel Determination: the attacker knows the WiFi transmission channel (target channel) used by the CBRS. 3) Attack Power Estimation: The power of the jamming signal should be large enough to stop the victim transmitter's transmission. 4) Stealthiness: The attack should not be easily detected, *i.e.*, the jamming signal should have stealthiness. In the following, we first explain the reasons why the above requirements are necessary. Then, we introduce our designs to enable the attacker to meet these requirements.

3.1 Attacker Transmitter Selection

Since a CBRS can use any permitted WiFi channel to transmit signals, the attacker transmitter should be able to cover all permitted WiFi channels. To meet this requirement, we opt to use NICs or software defined radios (SDRs), *e.g.*, USRP. These two kinds of devices can emit jamming signals at the target channel, yet have different performance in respect to different aspects. In terms of the expense, SDRs are more expensive than NICs. For example, an USRP costs about 773 dollars yet a NIC normally costs about 20 dollars. For the propagation distance of emitted jamming signals, SDRs usually perform better because their upmost transmission power is higher than that of NICs. Thus, SDRs can attack CBRSes within a larger range, although SDRs are more costly than NICs. The attacker needs to make a trade-off between the attack range and cost.

3.2 Target Channel Determination

To jam the legitimate signal in a CBRS, the attacker needs to determine the channel used by the victim transmitter, *i.e.*, knowing the channel index (each permitted channel has a unique index). Intuitively, we can employ a NIC or SDR to collect signals around the CBRS to detect the channel index. However, there are many transmission channels used for daily communications in the ambient environments. As a result, the target channel might be overwhelmed by other irrelevant transmission channels, which confuses the attacker. Fortunately, to identify the target channel, the attacker can utilize the time interval between any two continuous packets to distinguish the target channel from other irrelevant ones. This is because such time intervals are stable in a CBRS but generally unstable in a communication system. To validate the feasibility of the above countermeasure, we first collect a batch of WiFi signals with different transmission channels around a CBRS, and then calculate the time intervals for each transmission channel. The box-plot of the time interval distributions are shown in Fig. 2. It can be observed that the time intervals of the target channel are significantly stable (with small box and a few black circles), while those of other transmission channels are unstable (with large box and lots of black circles). Therefore, the attacker can easily distinguish the target channel from other irrelevant ones according to the time interval distribution.

3.3 Attack Power Estimation

To trigger desired CSI absence, the power of jamming signals should be larger than the collision avoidance threshold of the victim transmitter when the jamming signals reach the victim transmitter. Hence, the attacker needs to guarantee that the power of jamming signals is still high enough after experiencing decay during propagation. To estimate the emitting power of jamming signals, two parameters should be known in advance: the collision avoidance threshold and the distance between the attacker transmitter and the victim transmitter. Fortunately, the threshold is fixed as -62 dBm according to the IEEE 802.11 standard [23]. The distance can be easily measured, *e.g.*, via a telemeter. Then, according to [31], the attacker can estimate the emitting power using the following power decay formula:

$$L_d = 92.3 + 20\log(d) + 20\log(f), \tag{3}$$

where L_d, d, and f are the power decay, distance, and the frequency, respectively. In physical-world attack, in consideration of the extra power decay caused by occlusion (*e.g.*, wall), multi-path effect, and antenna gain, we add 3dB to the estimated power empirically. The real emitting power should be equal to or larger than the estimated power. Recalling that an attacker can use both NICs and SDRs to emit jamming signals, the largest attack distance can be estimated based on Eq. 3. Since the largest transmission power of NICs and SDRs are 18 dBm [32] and 20 dBm [33] respectively, the largest attack range of them are 43.3 meters and 54.6 meters respectively, without regard of other power decay excluding L_d.

3.4 Stealthiness of Attack

In order to launch attacks stealthily, a sophisticated attacker should make the jamming signal effectively concealed, *i.e.*,

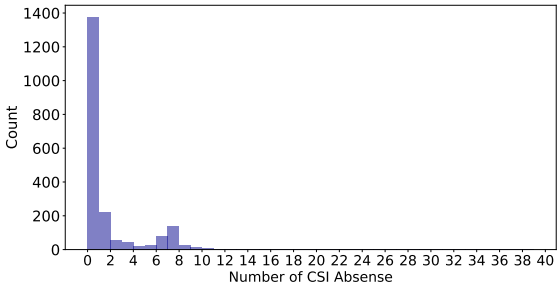


Fig. 3. Histogram of the number of CSI absences in each normal CSI sample.

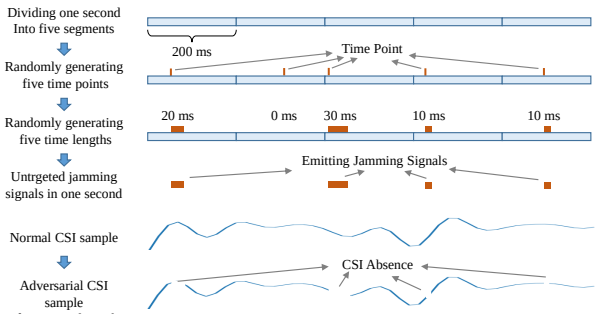


Fig. 4. Generation flow of untargeted jamming signals.

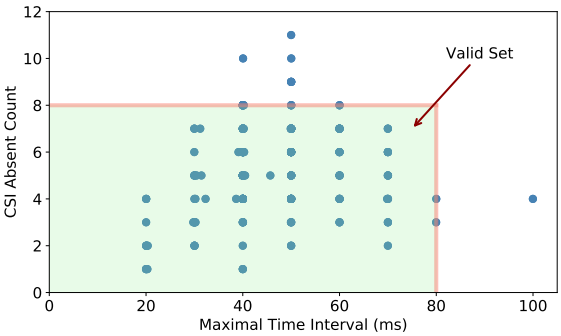


Fig. 5. Distribution of adversarial CSI samples.

the adversarial CSI samples should be difficult to distinguish from normal ones. To this end, we conduct a preliminary experiment to explore the feasibility of satisfying the stealthiness. To be specific, we first collect over 2000 normal CSI samples in a normal laboratory environment from six reproduced CBRSes [9], [10], [11], [12], [13], [14]. Then, we calculate the time intervals in each CSI samples. The experimental result shows that the CSI absence appears in over 50% normal CSI samples. The reasons causing this phenomenon are threefold: 1) The reflection/absorption/occlusion of human body may hinder the signal propagation, resulting in the CSI absence in the received signals. This kind of CSI absence is also a kind of feature of user behavior, because different behaviors cause different reflection/absorption/occlusion. 2) There are massive WiFi signals in the ambient environments. Some of them may be at the target channel, leading to the CSI absence in normal CSI samples. 3) With the hardware imperfection of the transmitter/receiver, some packets may not be successfully transmitted/received, which also induces CSI absence. Therefore, it is difficult for the CBRS to judge whether a CSI absence is induced by malicious jamming signals or other natural factors.

Afterwards, we count the number of CSI absences in every one-second in each CSI sample and show the histogram in Fig. 3. It can be found that the most frequency of CSI absence is smaller than 8. Besides, we find that most time intervals of CSI absences are less than 80 milliseconds. Therefore, similar to the X in Eq. 2, we define the *valid set*. In the valid set, each CSI sample contains no more than N_{abs} CSI absences per second, and the time length of the longest CSI absence in this sample is less than T_{abs} milliseconds. By default, we set N_{abs} and T_{abs} as 8 and 80, respectively. Accordingly, to guarantee the stealthiness, the number of jamming attempts per second and each jamming duration should be smaller than N_{abs} and T_{abs} milliseconds, respectively. In this way, the generated adversarial CSI sample will fall into the valid set with a high probability, and hence be indistinguishable from normal CSI samples.

4 Attack Approaches

In this section, we detail the untargeted and targeted attack approaches.

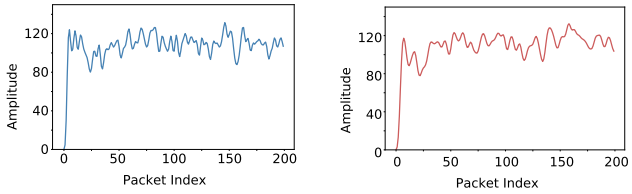
4.1 Untargeted Attack

Unlike [22] that crafts attacking signals with expert knowledge on CTI, we aim to achieve imperceptible attacks *while minimizing the requirements for attack*. Thus, we opt to generate random jamming signals for untargeted attack. In this way, the attacker does not need to have any expert knowledge and the victim cannot find the attack pattern. Based on the analysis in Section 3.4, we summarize the untargeted jamming signal generation flow to the following steps:

1) Dividing one second into N_{abs} segments in the temporal domain. Each segment is $1000/N_{abs}$ milliseconds.
2) Randomly generating N_{abs} jamming start time points (from t_s^1 to $t_s^{N_{abs}}$) for N_{abs} segments, the jamming signal will be emitted since the start time point.
3) Randomly generating N_{abs} jamming time lengths (from l_s^1 to $l_s^{N_{abs}}$) for N_{abs} segments, with each time length less than or equals to T_{abs} milliseconds.
4) In one second, the jamming signal is emitted at t_s^i for l_s^i milliseconds ($i \in [1, N_{abs}]$).

A generation flow of the untargeted jamming signal is illustrated in Fig. 4. In this example, $N_{abs} = 5$ and $T_{abs} = 30$. To continuously launch online untargeted attacks at time t_{att}, the attacker only needs to repeat the last three steps since t_{att}.

The essential goal of the above flow is jamming the target channel stealthily. To validate the stealthiness, we invite volunteers to repeat three activities ('walk', 'sit', and 'fall') introduced in [12] and perform untargeted attack.



(a) Normal CSI sample. (b) Adversarial CSI sample.

Fig. 6. Waveform of normal and adversarial CSI samples.

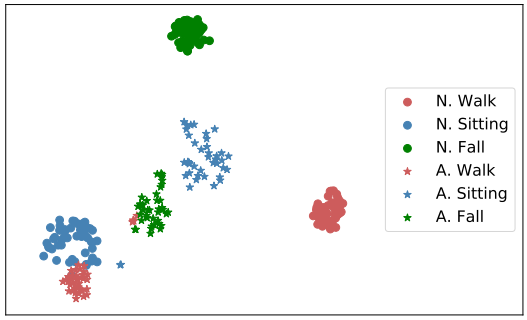


Fig. 7. Distributions of normal CSI samples and attacked CSI samples. 'N.' means 'normal' and 'A.' means 'attacked'.

Meanwhile, we collect untargeted adversarial CSI samples (*i.e.*, the signal samples that under untargeted attacks). Then we show the time interval distributions of the adversarial CSI samples in Fig. 5. It can be observed that the majority of adversarial CSI samples lie in the valid set. Moreover, we found that the waveform of normal CSI sample (Fig. 6(a)) is similar to that of the adversarial one (Fig. 6(b)). Thus, the proposed untargeted attack conceals itself well.

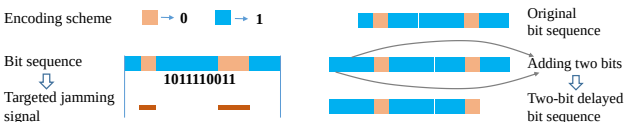
Finally, we validate the attack effectiveness of our approach. In order to visually observe the attack effectiveness, we utilize the t-SNE [34] algorithm to reduce the dimension of statistical features of both normal and attacked CSI samples, and show the result in Fig. 7. It can be found that the normal CSI samples of different activities (marked by different colors) are separated far away from each other, while that of the same activity are close to each other. More importantly, the attacked CSI samples of different activities are almost mixed together. For the 'walk' activity, the attacked CSI samples (marked by red stars) are distant from the normal ones of 'walk' yet close to the normal CSI samples of 'sit down'. In this case, a classifier trained with normal CSI samples would misclassify the attacked CSI samples. Therefore, our untargeted attack approach is simple but effective. This is reasonable because the loss of image pixels can also achieve high attack effectiveness even under black-box conditions in computer vision field [35].

4.2 Targeted Attack

In this part, we describe the targeted attack approach, in which an attacker can manually design a jamming signal $s_j^{a\to b}$, such that a CSI sample H^a of a specific behavior y_a can be classified as a target behavior y_b, *i.e.*, $y_b = F_w(f_{ext}(J(H^a, s_j^{a\to b})))$.

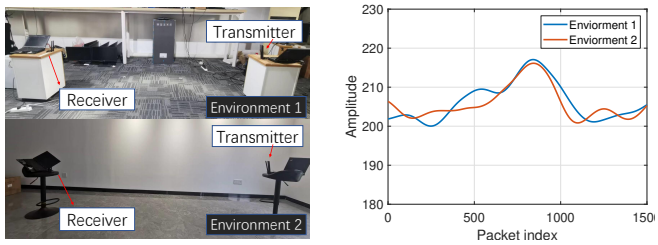
4.2.1 Methodology

To perform targeted attack, conventional approaches are to randomly generate a perturbation, and then leverage differentiable gradient descent to adjust its elements to optimize the perturbation. The perturbation can be added



(a) Encoding scheme. (b) Delay method.

Fig. 8. Jamming signal encoding scheme and delay method.



(a) Two similar CBRS contexts. (b) CSI profiles.

Fig. 9. Two CSI profiles of 'kicking' in two similar contexts are similar.

to normal samples to generate adversarial ones [21], [28]. Nevertheless, these approaches cannot be used to generate $s_j^{a\to b}$ in our attack scenario, because what an attacker can do is to cause CSI absence (*i.e.*, element loss) of H^a, rather than increasing/decreasing its element values. More importantly, this process is non-differentiable.

To tackle this challenge, we opt to use the *genetic algorithm*. The core components of the genetic algorithm are how to calculate the *fitness score*, and how to encode and decode the *jamming signal*. If the genetic algorithm is used in our attack scenario, the attacker will generate better jamming signals (the signal with higher fitness score) and encoding them to feed into a fitness function (designed to calculate fitness score), until reach the optimum. The optimum is such a jamming signal that has the highest probability to mislead the behavior classifier to output a behavior label specified by the attacker.

Encoding scheme: To feed jamming signal into fitness function to calculate fitness score, we propose an encoding scheme for transforming the jamming signal $s_j^{a\to b}$. We observed that each element in H^a is only in one of the two states, *i.e.*, either 'absent' or 'captured' during attack. In this case, the state of each element can be encoded as '0' (absent) or '1' (captured). Hence, the jamming signal can be represented by a bit sequence.

Suppose that the transmission rate of the victim transmitter is n_p packets per second and each behavior exists one second, each normal CSI sample would have n_p elements for each frequency. Moreover, since once a packet is not captured, the elements of all frequency corresponding to this packet would be absent simultaneously. Without loss of generality, we assume that only one frequency is used by the victim transmitter to ease our following explanation. Under the above assumptions, the jamming signal can be encoded as a bit sequence that contains n_p bits. As shown in Fig. 8(a), a '0' in the bit sequence means the attacker transmitter emits jamming signals (making the packet absent) and a '1' means the attacker transmitter stops jamming (making the packet captured). Accordingly, the jamming function $J(\cdot)$ can be formulated as:

$$J(H^a, s_j^{a\to b}) = H^a \circ s_j^{a\to b}, \tag{4}$$

where $\circ$ is *Bitwise AND* operation [36].

Fitness score: In the genetic algorithm, each bit sequence is assigned with a fitness score to measure how close it is to the optimum. In our attack scenario, we regard the confidence coefficient calculated by the behavior classifier $F_w(\cdot)$ as the fitness score, because such confidence coefficient measures the probability that an input CSI sample should be classified as a behavior label. Therefore, a larger confidence coefficient

means a larger probability, *i.e.*, a larger fitness score. The fitness score F^b of $s_j^{a\to b}$ can be calculated by:

$$F^b = Fit^b(F_w(f_{ext}(H^a \circ s_j^{a\to b}))) = Fit^b(H^a, s_j^{a\to b}), \quad (5)$$

where $Fit^b(F_w(\cdot))$ is the fitness function and it outputs the confidence coefficient of the behavior label y_b calculated by the behavior classifier $F_w(\cdot)$.

Nevertheless, in a realistic attack scenario, the attacker has no access to the behavior classifier of the target CBRS, not mention to the confidence coefficient. To solve this problem, the attacker can simulate a CBRS context similar to that of the target CBRS to collect CSI samples, on which a surrogate behavior classifier can be trained to obtain confidence coefficient. This is rational because the CSI samples collected under similar contexts are similar as well. As shown in Fig. 9, the CSI profile of 'kicking' collected in laboratory (Environment 1) is similar to that collected in hall (Environment 2). Thus, this countermeasure is reasonable. Note that the attacker does not need to know the architecture of the target behavior classifier. The experiment results in Sec. 5.6 demonstrate that our attack approach has good transferability.

4.2.2 Suppressing the Impact of Delay

So far, it seems that we can leverage the fitness function $Fit(\cdot)$ to optimize $s_j^{a\to b}$. However, certain delay exists in real-world attacks, *i.e.*, the normal CSI sample H^a and jamming signal $s_j^{a\to b}$ are not synchronized. This is because that even if the attacker instantly emits jamming signals once detects the beginning of a behavior, the time point that the jamming signals reach the victim transmitter would lag behind the beginning time point of the behavior. The lagging is induced by the propagation delay and hardware delay. The delay would make the received CSI sample not aligned with $H^a \circ s_j^{a\to b}$, and deteriorate the attack effectiveness.

To suppress the impact of delay, our solution is to enhance the fitness function. Specifically, the attacker can manually introduce a delay into the jamming signal during the optimization. The purpose is to improve the tolerance against the delay. As shown in Fig. 8(b), we deliberately generate a delay of n_d bits in a bit sequence through two steps:

1) Adding n_d '1' in the head of the bit sequence, so that the new bit sequence contains $n_d + n_p$ bits.
2) Removing n_d bits from the tail of the new bit sequence and a delayed bit sequence with n_p bits is finally obtained.

If we denote the function of delaying $s_j^{a\to b}$ for n_d bits as $D(s_j^{a\to b}, n_d)$, F_b is enhanced to a weighted fitness function as follow:

$$F^b = \sum_{i=0}^{n_d} \omega_i \cdot Fit^b(H^a, D(s_j^{a\to b}, i)), \quad (6)$$

where $\omega_i \in [0, 1]$ denotes the weight for the i-bit delayed bit sequence and $\omega_i \geq \omega_{i+1}$. Empirically, we set n_d to 5.

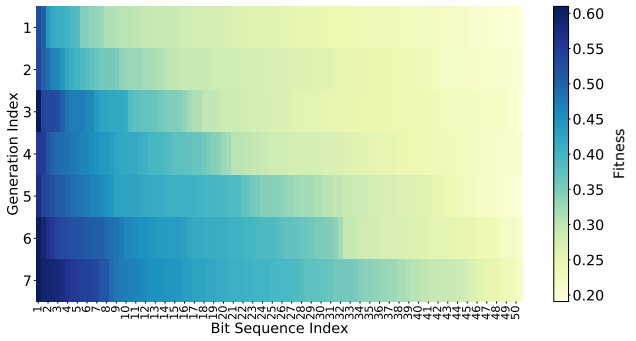


Fig. 10. Sum of the fitness scores of the generation increases as the increase of the iteration. The deeper the color is, the larger the fitness is.

4.2.3 Jamming Signal Optimization

With the fitness function, the optimal $s_j^{a\to b}$, *i.e.*, the optimization objective (which is equivalent to the objective in Eq. 2) can be formulated as:

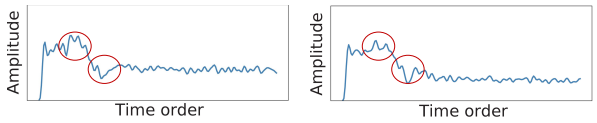
$$\max_{s_j^{a\to b}} \quad \sum_{i=0}^{n_d} \omega_i \cdot Fit^b(H^a, D(s_j^{a\to b}, i)), \quad (7a)$$

$$\text{s.t.} \quad J(H^a, s_j^{a\to b}) \in X. \quad (7b)$$

Achieving this objective requires the following operations:

1) Initial generation: The attacker randomly generates N_b bit sequences that are in the valid set as the initial generation.
2) Fitness calculation: The attacker calculates the fitness score of every bit sequence in the generation.
3) Duplication: The attacker sorts the bit sequences according to their fitness scores. The top N_{dup} bit sequences are duplicated and the N_{dup} bit sequences with lowest fitness scores are removed.
4) Crossover: N_{cro} pairs of bit sequences are randomly selected from the generation to perform crossover. In each pair of bit sequences, we exchange their last n_{cro} bits.
5) Mutation: The attacker first randomly selects N_{mut} bit sequences, and then randomly selects n_{mut} bits from each of the N_{mut} bit sequences. The *Bitwise NEGATION* process [36] is then performed on these n_{mut} bits.

The first operation only needs to be performed once at the beginning of the optimization, yet the following four operations are alternately conducted in multiple iterations. As shown in Fig. 10, a new generation will be produced in each iteration, which would be better than the previous generation. However, in practice, we find that a generation might degrade, *i.e.*, the sum of the fitness scores of current generation is smaller than that of the former generation after crossover and mutation. We term this phenomenon as degeneration. To deal with this problem, we introduce a mechanism that the crossover and mutation operations will be re-conducted once degeneration occurs. Moreover, in some generations the attacker might re-conduct the crossover or mutation operation to guarantee that the bit sequence is in the valid set. The iteration will not stop unless the degeneration continuously occurs for N_{end} times. Empirically, N_{end} is set as 10. When the iteration terminates, we regard the bit sequence that has the largest fitness score



(a) CSI sample H_1^a. (b) CSI sample H_2^a.

Fig. 11. Two normal CSI samples of 'sit down' with small differences in red circles.

as the optimum and decode it to obtain the final $\boldsymbol{s_j^{a\to b}}$. The adversarial CSI sample generated by $J(H^a, \boldsymbol{s_j^{a\to b}})$ is most likely to be classified as behavior y_b. Furthermore, the attacker can use random offspring generation [37] to prevent our approach from being trapped into local optimum. Or, if our approach outputs a local optimum, the attacker can further perform simulated annealing [38] to get the global optimum.

4.2.4 Attack Robustness Enhancement

In real-world scenarios, the H for a specific behavior is not unique. For example, two normal CSI samples of 'sit down' are presented in Fig. 11. We find that although the holistic profiles of the two curves are similar, their local profiles are different. In this case, the $\boldsymbol{s_j^{a\to b}}$ generated for CSI sample H_1^a may be ineffective in attacking CSI sample H_2^a. To solve this practical problem, we further enhance the fitness function and objective function. Specifically, an attacker can first collect a batch of CSI samples containing that of behavior y_a to train the $F_w(\cdot)$. Then, the attacker can sum the fitness scores of all CSI samples of behavior y_a to improve the robustness of the generated jamming signal $\boldsymbol{s_j^{a\to b}}$. If we denote the number of the CSI samples of y_a in the batch as n_{bat}, the enhanced fitness function can be formulated as:

$$F^b = \sum_{j=1}^{n_{bat}} \sum_{i=0}^{n_d} \omega_i \cdot Fit^b(H^{a_j}, D(s_j^{a\to b}, i)), \quad \text{s.t.} \quad \omega_i \in [0,1]. \tag{8}$$

The corresponding objective becomes:

$$\max_{s_j^{a\to b}} \quad \sum_{j=1}^{n_{bat}} \sum_{i=0}^{n_d} \omega_i \cdot Fit^b(H^{a_j}, D(s_j^{a\to b}, i)), \tag{9a}$$

$$\text{s.t.} \quad \omega_i \in [0,1] \quad \text{and} \quad J(H^a, s_j^{a\to b}) \in X. \tag{9b}$$

By using the batch and Eq. 8, the attacker can generate a more robust $\boldsymbol{s_j^{a\to b}}$ to attack both H_1^a and H_2^a.

5 Evaluation and Result

Existing CBRSes can be divided into two categories according to whether the behavior classifier is based on deep neural network or not. We select two representatives for each category and conduct experiments over them: WiFall [12], STFT [13], SignFi [11], and WiLSTM [9]. WiFall and STFT use random forest (RF) [39] and logistic regression (LR) [40] as behavior classifiers, respectively. The classifiers in SignFi and WiLSTM are most commonly used deep neural networks, *i.e.*, convolutional neural network (CNN) and long-short term memory (LSTM). These four systems can achieve high behavior recognition accuracy.

5.1 Experiment Setup and Metrics

Experiment setup: We reproduce four representative CBRSes and ensure that our implementation has comparable behavior recognition accuracy to the reference. The implementation details are summarized as follows:

- In WiFall, six statistical features are calculated as the input of the behavior classifier. WiFall leverages RF to classify four activities including fall.
- Frequency domain features are extracted as the input of classifiers in STFT. STFT can leverage LR to recognize six activities 'lie down, fall, walk, run, sit down, and stand up'.
- Focusing on hand sign recognition, SignFi utilizes a CNN to classify the features containing both CSI amplitude and phase. We reproduced SignFi to recognize ten hand signs that represents ten numbers from zero to nine.
- The WiLSTM system utilizes an LSTM classifier and CSI amplitudes to recognize six activities similar to those in STFT.

As illustrated in Fig. 12, we implement these systems under three different environments, including laboratory, home, and hall. The victim transmitter is equipped with an Intel 5300 NIC and three antennas. The transmission rate of the CBRSes is 100 packets per second and each behavior lasts two seconds. For the attacker transmitters, we use both NIC (Atheros 9380) and SDR (USRP B210) to emit jamming signals. The jamming signals are modulated by *LabVIEW* [41]. We invite 25 volunteers (18 males and 7 females) aged from 21 to 29 to collect CSI samples. In each environment, volunteers are asked to perform behaviors between the victim transmitter and receiver (with three antennas). At least 5 persons are involved in each experiment. In the default setting, the distance between the victim and attacker transmitters is about three meters. We totally collect 14772 normal CSI samples, 16479 CSI samples under untargeted attacks, and 31270 CSI samples under targeted attacks. All the experiments are carried out by adhering to the approval of our university's Institutional Review Board (IRB).

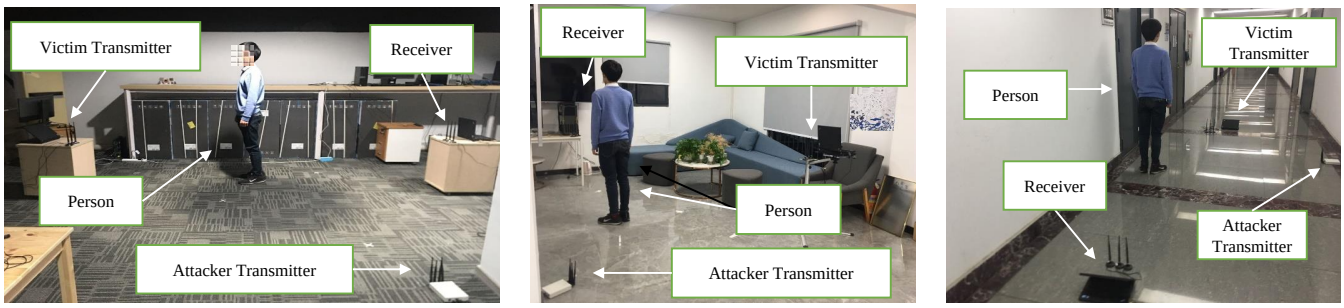
Metrics: We defined two metrics to quantitatively measure the attack effectiveness: untargeted attack success rate (UASR) and targeted attack success rate (TASR). UASR is the probability that the jamming signals mislead the CBRS to output a false behavior label. It can be calculated by:

$$UASR = Acc_{nor} - \frac{N_{unt}^{cor}}{N_{unt}^{all}}, \tag{10}$$

where Acc_{nor}, N_{unt}^{cor}, and N_{unt}^{all} are the reproduced behavior recognition accuracy of the four systems (*e.g.*, 98.5% in WiFall), the number of correctly classified untargeted adversarial CSI samples, and the number of all untargeted adversarial CSI samples, respectively. Similarly, TASR is the probability that a CSI sample of behavior y_a is classified as the behavior y_b when the victim transmitter is influenced by the targeted jamming signal $\boldsymbol{s_j^{a\to b}}$. It can be calculated by:

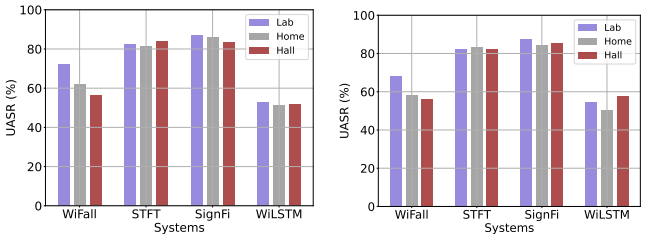
$$TASR = \frac{N_{tar}^{cor}}{N_{tar}^{all}}, \tag{11}$$

where N_{tar}^{cor} and N_{tar}^{all} are the number of targeted adversarial CSI samples that are classified as the target behavior and the



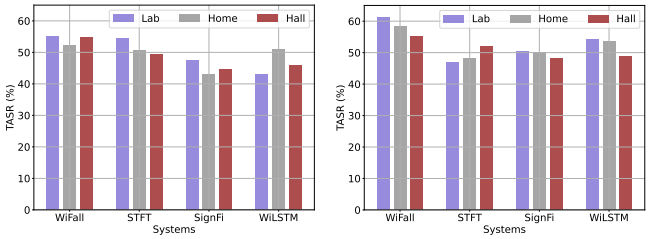
(a) Laboratory environment. (b) Home environment. (c) Hall environment.

Fig. 12. Experiment setup in three environments.



(a) NIC. (b) SDR.

Fig. 13. Effectiveness of untargeted attacks in three environments.



(a) NIC. (b) SDR.

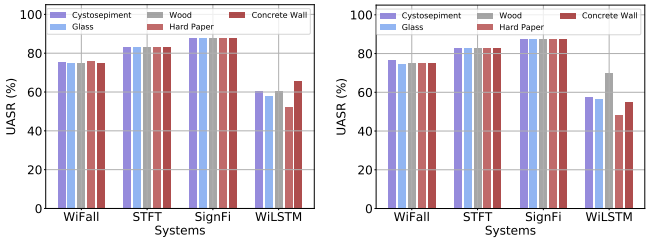
Fig. 14. Effectiveness of our targeted attack approach ('Genetic Algorithm') and a baselines ('Random').

number of all targeted adversarial CSI samples of possible (y_a, y_b) pairs.

5.2 Overall Attack Effectiveness

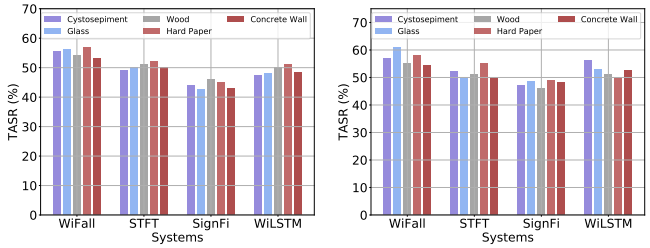
To measure the effectiveness of our attack approaches, we first calculate the UASRs and TASRs of all volunteers, and then obtain the averages as the final results. The UASRs of NIC and SDR are shown in Fig. 13. It can be observed that, with a NIC as the attacker transmitter, the highest UASRs for WiFall, STFT, SignFi, and WiLSTM can achieve 72.3%, 84.0%, 87.0%, and 52.6%, respectively. As for the SDR, the highest UASRs for the four systems are 68.2%, 83.3%, 87.5%, and 57.5%, respectively. Besides, in most cases, there is no obvious UASR difference among the three environments. The UASR of WiFall in lab is higher than that in the other environments. It is very likely to be induced by the hardware imperfection and ambient RF noise, as the lab has many WiFi access points that could emit interfering WiFI signals from time to time, while the RF environments in the home and hall are relatively clear. In short, the high UASRs on the three environments indicate that our untargeted attack approach is significantly effective.

For the targeted attack, we average the TASRs over three environments and compare the targeted attack approach with a baseline, *i.e.*, the random jamming signal generation method in the untargeted attack approach. The results of



(a) NIC. (b) SDR.

Fig. 15. Effectiveness of untargeted attack under different occlusion objects.



(a) NIC. (b) SDR.

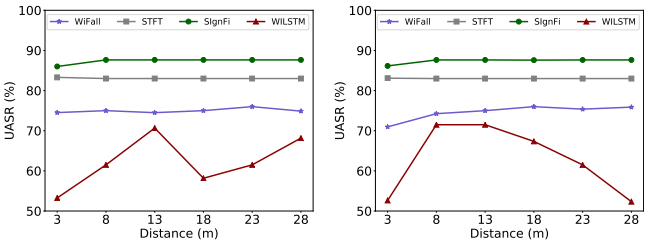
Fig. 16. Effectiveness of targeted attack under different occlusion objects.

NIC and SDR are shown in Fig. 14. 'Random' means the baseline and 'Genetic Algorithm' represents our targeted attack approach. It can be found that our targeted attack approach outperforms the baseline in all systems. The highest TASRs of NIC for these four systems are 55.2%, 54.5%, 47.5%, and 51.0% respectively. For the SDR, the highest TASRs for these systems are 61.2%, 52.0%, 50.5%, and 54.2%, respectively.

We also evaluate the time cost used to launch an attack. The time cost can be divided into two parts: signal generation and signal emission. For doing the former, our program needs to spend a few microseconds. Moreover, the hardware can achieve the latter within 16 microseconds. Thus, the attack can be launched in real time.

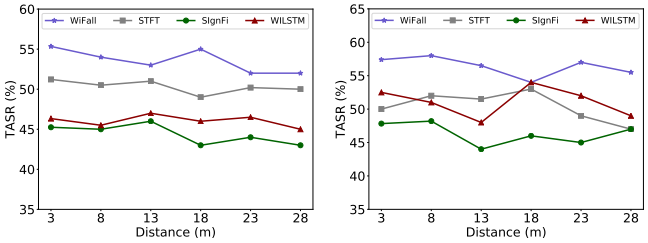
5.3 Non-Line-Of-Sight Attack

In real-world attack scenarios, the main propagation path of signals between the victim transmitter and the attacker transmitter may be occluded by some objects. This attack scenario is called NLOS attack. The power of jamming signals under this scenario would be reduced by the occlusion object. We also evaluate our approach in this extreme case. Specifically, we place the attacker transmitter eight meters away from the victim transmitter and test with five types of materials contained by the objects in our daily



(a) NIC. (b) SDR.

Fig. 17. Effectiveness of untargeted attack under different attack distances.



(a) NIC. (b) SDR.

Fig. 18. Effectiveness of targeted attack under different attack distances.

lives: cystosepiment, glass, wood, hard paper, and concrete wall. The thickness of them is 10.8, 0.5, 1.0, 2.5, and 28.0 centimeters respectively. In the setting of the concrete wall, the CBRS and the attacker transmitter are in different rooms. The untargeted attack results of NIC and SDR are shown in Fig. 15. We can observe that the UASRs of different occlusion objects are similar, no matter we use NIC or SDR to emit jamming signals. The reason behind is that the jamming signal is utilized to stop the victim transmitter emitting signals, rather than change the values of normal CSI elements. As long as the power of the jamming signal is larger than the collision avoidance threshold of the victim transmitter, the attack can be successful. We also show the targeted attack results in Fig. 16, in which we can find that there is no obvious difference among different occlusion objects as well. Therefore, our attack approaches are still effective under occlusion conditions.

5.4 Impact of Distance

To explore the impact of the distance between the victim transmitter and the attacker transmitter, we change the distance from 3 meters to 28 meters with a step of 5 meters. The UASRs of the NIC and SDR are shown in Fig. 17. Similar to the results of NLOS attack experiments, the distance (within 28 meters) has negligible effects on the attack effectiveness in WiFall, STFT, and SignFi. However, the UASRs of the WiLSTM system are unstable and jittering within the range from 52.3% to 71.5% randomly. This randomness is not induced by the distance variation, but the randomness in our untargeted jamming signal generation approach. The targeted attack results under different distances are shown in Fig. 18. Likewise, the distance does not affect the targeted attack effectiveness much. Therefore, an attacker is able to effectively launch long-distance untargeted and targeted attacks, while being hardly detected by CBRS users.

5.5 Impact of Transmission Rate

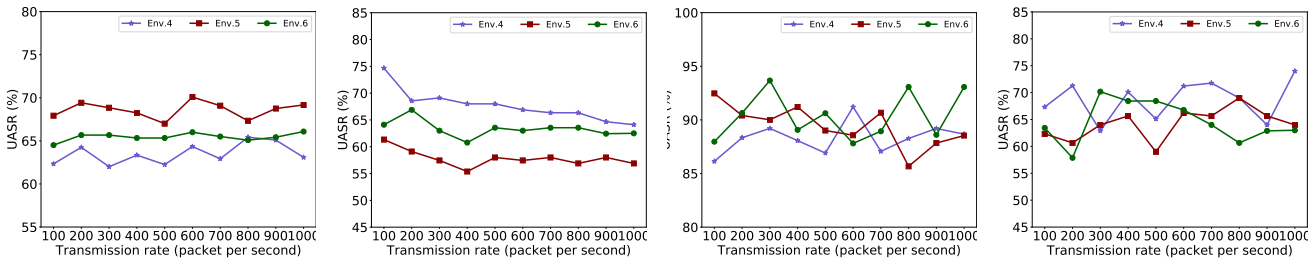
In our default setting, the transmission rate of the victim transmitter is 100 packets per second. This transmission rate is adopted by many CBRSes [12]. However, 100 is not the only choice. CBRS users can use any transmission rate (*e.g.*, 500) when they train the behavior classifier. In this part, we explore the impact of transmission rate by varying it from 100 to 1000 in step of 100. The effectiveness of NIC of the four systems in three environments are shown in Fig. 19. It can be observed that the variation of the transmission rate does not affect the attack effectiveness too much when we use NIC as the attacker transmitter. The attack effectiveness of SDR of the four systems are shown in Fig. 20(a), (b), (c), and (d), respectively. Similar to the results in Fig. 19, we do not find apparent UASR variation when the transmission rate increases. Hence, the transmission rate hardly impacts the effectiveness of targeted attack.

5.6 Transferability Study

We also evaluate the transferability of our adversarial samples (*i.e.*, a black-box setting). We train new classifiers with different architecture parameters among different tasks by following the standard setting [42], and then feed the previous adversarial samples into the new classifier. Specifically, we respectively used an RF classifier with 100 trees, a LR classifier with 'one vs. rest' strategy, a four-layer CNN, and a Bi-LSTM to design jamming signals in WiFall, STFT, SignFi, and WiLSTM, while testing the attack effectiveness with an RF classifier with 50 trees, a LR with multinomial loss, a five-layer CNN, and an LSTM, respectively. The results show that the TASRs for NIC are 50.6%, 42.0%, 30.8%, and 37.2% in WiFall, STFT, SignFi, and WiLSTM, respectively. Meanwhile, the TASRs of SDR for these four systems are 53.0%, 45.6%, 32.5%, and 42.0% respectively. It can be found that the TASRs for WiFall, STFT, and WiLSTM only drop about 7%, which means that our targeted attack approach has decent transferability. Although the TASR of SignFi decreases a lot, it is still higher than 30.0%, which is also impactful in CBRS attacking. Hence, our attack approaches are effective under black-box conditions.

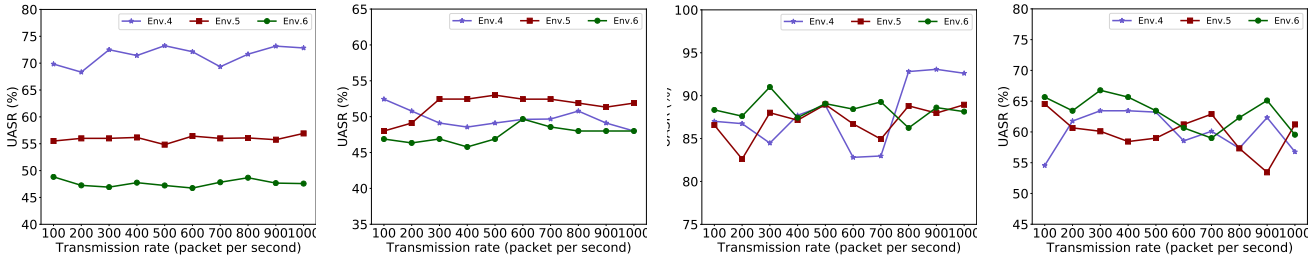
5.7 Universality of the Attack Approach

In addition to behavior recognition, our attack approaches can be generalized to other WiFi-based sensing applications, such as user authentication [43] and localization [44]. This is because: 1) The transceivers in these applications also comply with the CSMA/CA protocol; 2) Many of them leverage machine learning techniques to achieve the sensing goals. To show the feasibility of such attacks. we reproduce a WiFi-based user authentication system WiPIN [43] with a false accept rate of 2.4%. As shown in Fig. 12, WiPIN also utilizes a pair of transceivers to probe the identity information of the user in between. To launch untargeted attacks, we also use the attack transmitter (NIC/SDR) in Fig. 12 to emit jamming signals designed by our untargeted attack approach. As a result, 10.0% CSI samples of illegitimate users are falsely accepted as legitimate users. Intuitively, a better attack effectiveness can be achieved by using our targeted attack approach. For doing so, the attacker can first stealthily record the identity information of the victim with his/er own transceivers. Since WiFi signals



(a) WiFall. (b) STFT. (c) SignFi. (d) WiLSTM.

Fig. 19. Effectiveness of untargeted attack of NIC under different transmission rates.



(a) WiFall. (b) STFT. (c) SignFi. (d) WiLSTM.

Fig. 20. Effectiveness of untargeted attack of SDR under different transmission rates.

are imperceptible to humans, the victim will not realize that s/he is scanned by malicious WiFi signals. Then, the attacker can use our targeted attack approach to optimize jamming signals based on recorded identity information and launch targeted attacks. The attack scenario is similar to Fig. 12 as well. Therefore, our attack approaches are effective in compromising other WiFi-based sensing applications.

6 Mitigation and Discussion

6.1 Geofencing WiFi Signals

Geofencing stops jamming signals from reaching the victim transmitter. A necessity of our attack approaches is that the power of jamming signals around the victim transmitter is larger than the collision avoidance threshold. Thus, geofencing, such as building walls with metal and painting walls with electromagnetic shielding paints, is an effective mitigation solution. However, it is undesirable to adopt geofencing as: 1) Geofencing also blocks legitimate WiFi signals, which affects the normal use of WiFi signals for communication. 2) Geofencing usually is costly. Strategic geofencing remains challenging.

6.2 Adversarial Sample Identification

With this mitigation method, we can determine whether a CSI sample is adversarial or not. To be specific, we first extract five statistical features reflecting the degree of CSI absence from CSI samples. These features include the number of delay times, maximal delay, minimal delay, average delay, and median of delay. Then, we train a one-class classifier (isolation forest [45]) with the features of normal CSI samples. The experiment results show that our classifier can identify all the adversarial samples introduced by NIC and 98.8% adversarial samples caused by SDR. Thus, this mitigation can help CBRSes defend against our attacks effectively. Nevertheless, 23% normal samples are misclassified as adversarial ones. Therefore, this mitigation also sacrifices a little usability of CBRSes. It is difficult to balance the usability and security while using this mitigation method.

6.3 Adversarial Training

To mitigate the impacts of adversarial CSI samples, users can improve the robustness of the behavior classifier by adding some adversarial CSI samples to the classifier's training set. These adversarial samples can be collected by the user with his/er own devices by simulating the attack scenario. In this way, the classification accuracy of adversarial CSI samples in WiFall, STFT, SignFi, and WiLSTM can achieve 65.0%, 68.8.0%, 75%, and 62.5%, respectively. However, adopting this mitigation method has to deal with a trade-off between the usability and security due to the following reasons: 1) Adding adversarial CSI samples into the training set brings massive extra overhead since users need to simulate the attack to collect adversarial CSI samples; 2) This mitigation method induces the degradation of normal CSI samples' classification accuracy, *e.g.*, a 13% decrease in STFT system. Therefore, this mitigation should be further improved in defending against the proposed attacks.

6.4 Discussion

As aforementioned, we present three mitigations to defend against our attack approaches. Indeed, these solutions require the user to make a trade-off between the usability and security. However, it is this point that suggests that researchers should pay more attention to the security of WiFi sensing rather than always focusing on improving the sensing precision. Here, we give two potential solutions that could completely solve the security problem. 1) Training a stronger adversarial sample discriminator. In Sec. 6.2, we feed some statistical features into the isolation forest to distinguish adversarial samples from normal ones. It is possible to extract more representative features that can better characterize the adversarial sample. With these enhanced features, the user can employ a stronger classification algorithm like deep neural network to accurately identify adversarial samples without misjudgment. 2) Improving behavior classifier's robustness. In Sec. 6.3, we propose to add adversarial samples to the training set to improve the

robustness of the behavior classifier. In fact, such robustness could be further improved by more advanced adversarial training techniques like FreeAT [46]. Specifically, the user can train the behavior classifier in multiple epochs. In each epoch, the user updates not only the parameters but also the jamming signals. With the evolution of the jamming signal, the behavior classifier will become more robust as the number of iterations increases. In this way, it is possible to obtain a behavior classifier that can accurately recognize both adversarial and normal CSI samples.

7 Related work

Behavior recognition systems have been widely deployed in many human-computer interaction applications. Traditional behavior recognition system usually is camera-, wearable-, phone-, or sonar-based [1], [2], [3], [4], [5], [6], [7], [7], [8], [47]. For example, Guan *et al.* [8] proposed to use ensemble LSTM to improve the gesture recognition accuracy of individual LSTM on wearables. To enable non-intrusive and device-free human behavior recognition, WiFi-based solutions [48] were proposed and developed rapidly. For instance, Guo *et al.* [16] have shown the feasibility of utilizing CSI amplitude and DT/RF/CNN/LSTM to accurately recognize activities. Nevertheless, previous works rarely paid attention to the security of the CBRS. In this paper, we explore the security of CBRS mainly from the perspective of an attacker.

WiFi-based attack techniques can be divided into active and passive ones according to whether the attack signal is emitted by the attacker or not. In the active attack, an attacker emits WiFi signals to sense physical-layer privacy of victims [49], [50], [51]. For example, Ali *et al.* [50] propose WiKey to sense a victim's keystroke. WiKey first emits WiFi signals towards the victim's keyboard, and then analyzes the signals reflected off the keyboard to infer the keystroke. In the passive attack, an attacker eavesdrops the WiFi signals emitted by victims and mine private information from these signals [29], [52], [53]. For instance, Cheng *et al.* [53] extract features from public WiFi signals to obtain WiFi users' privacy, such as identity, location, and financial privacy. To our knowledge, we are the first to achieve physical attacks towards CBRSes.

8 Conclusion

In this paper, we noticed that there are security threats in WiFi-based behavior recognition systems. By inducing CSI absence in collected CSI sample, we can manipulate the decision of the CBRS. We proposed two approaches to achieve untargeted attack and targeted attack, respectively. The experiment results on four real-world CBRSes demonstrated the high success rates of our attack approaches. Moreover, our attack approaches can be easily generalized to other WiFi-based sensing applications, *e.g.*, user authentication. At last, we introduced three methods to suppress the hazard of the attacks.

Acknowledgement

This paper is partially supported by the National Key R&D Program of China (2021QY0703), National Natural Science Foundation of China under grant U21A20462 and 62032021, and Research Institute of Cyberspace Governance in Zhejiang University.

References

[1] G. Gkioxari, R. B. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018, pp. 8359–8367.

[2] T. Li, Q. Liu, and X. Zhou, "Practical human sensing in the light," in *ACM International Conference on Mobile Systems, Applications, and Services, MobiSys*, 2016.

[3] M. Wang, B. Ni, and X. Yang, "Recurrent modeling of interaction context for collective activity recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017.

[4] K. Yatani and K. N. Truong, "Bodyscope: a wearable acoustic sensor for activity recognition," in *ACM Conference on Ubiquitous Computing Ubicomp*, A. K. Dey, H. Chu, and G. R. Hayes, Eds., 2012.

[5] R. Nandakumar, A. Takakuwa, T. Kohno, and S. Gollakota, "Covertband: Activity information leakage using music," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, IMWUT*, vol. 1, no. 3, pp. 87:1–87:24, 2017.

[6] K. Kalgaonkar and B. Raj, "One-handed gesture recognition using ultrasonic doppler sonar," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2009.

[7] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys*, vol. 46, no. 3, pp. 33:1–33:33, 2014.

[8] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, IMWUT*, vol. 1, no. 2, pp. 11:1–11:28, 2017.

[9] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, "Wifi CSI based passive human activity recognition using attention based BLSTM," *IEEE Transactions on Mobile Computing, TMC*, vol. 18, no. 11, pp. 2714–2724, 2019.

[10] J. Ma, H. Wang, D. Zhang, Y. Wang, and Y. Wang, "A survey on wi-fi based contactless activity recognition," in *IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, 2016.

[11] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, IMWUT*, 2018.

[12] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing, TMC*, vol. 16, no. 2, pp. 581–594, 2017.

[13] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.

[14] Q. Gao, J. Wang, X. Ma, X. Feng, and H. Wang, "Csi-based device-free wireless localization and activity recognition using radio image features," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 346–10 356, 2017.

[15] Y. Lu, S. Lv, and X. Wang, "Towards location independent gesture recognition with commodity wifi devices," *MDPI AG Electronics*, vol. 8, no. 10, p. 1069, Sep 2019.

[16] L. Guo, S. Guo, L. Wang, C. Lin, J. Liu, B. Lu, J. Fang, Z. Liu, Z. Shan, and J. Yang, "Wiar: A public dataset for wifi-based activity recognition," *IEEE Access*, vol. 7, pp. 154 935–154 945, 2019.

[17] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, W. Xu, and L. Su, "Towards environment independent device free human activity recognition," in *ACM International Conference on Mobile Computing and Networking, MobiCom*, 2018.

[18] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *ACM International Conference on Mobile Systems, Applications, and Services, MobiSys*, 2019.

[19] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *ACM International Conference on Mobile Computing and Networking, MobiCom*, 2015.

[20] Y. Tian, G. Lee, H. He, C. Hsu, and D. Katabi, "Rf-based fall monitoring using convolutional neural networks," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, IMWUT*, vol. 2, no. 3, pp. 137:1–137:24, 2018.
[21] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial sensor attack on lidar-based perception in autonomous driving," in *ACM SIGSAC Conference on Computer and Communications Security, CCS*, 2019.
[22] P. Huang, X. Zhang, S. Yu, and L. Guo, "Is-wars: Intelligent and stealthy adversarial attack to wi-fi-based human activity recognition systems," *IEEE Transactions on Dependable and Secure Computing*, 2021.
[23] IEEE, "IEEE standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks," *IEEE Standard 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016.
[24] TutorialsPoint, "Genetic algorithm," https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm, 2020.
[25] F. Wang, S. Zhou, S. Panev, J. Han, and D. Huang, "Person-in-wifi: Fine-grained person perception using wifi," in *IEEE/CVF International Conference on Computer Vision, ICCV*, 2019.
[26] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "802.11 with multiple antennas for dummies," *Computer Communication Review*, vol. 40, no. 1, pp. 19–25, 2010.
[27] N. Mostahinic and H. H. Refai, "Spectrum occupancy for 802.11a/n/ac homogeneous and heterogeneous networks," in *IEEE International Wireless Communications & Mobile Computing Conference, IWCMC*, 2019.
[28] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy, S&P*, 2017.
[29] Y. Zhu, Z. Xiao, Y. Chen, Z. Li, M. Liu, B. Y. Zhao, and H. Zheng, "Et tu alexa? when commodity wifi devices turn into adversarial motion sensors," in *Annual Network and Distributed System Security Symposium, NDSS*, 2020.
[30] T. Sugawara, B. Cyr, S. Rampazzi, D. Genkin, and K. Fu, "Light commands: Laser-based audio injection attacks on voice-controllable systems," in *Proceedings of the USENIX Security Symposium*, 2020.
[31] A. Goldsmith, *Wireless Communication*. Cambridge University Press, 2005.
[32] E. Specifier, "Sparklan introduces first 3t3r 450mbps wi-fi mini pcie cards with atheros ar9300 family," https://www.electronicspecifier.com/industries/wireless/sparklan-introduces-first-3t3r-450mbps-wi-fi-mini-pcie-cards-with-atheros-ar9300-family, 2020.
[33] U. S. D. R. D. Manual, "Usrp-2901 specifications," https://www.ni.com/documentation/en/usrp-software-defined-radio-device/latest/specs-usrp-2901/specs/, 2020.
[34] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.
[35] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.
[36] Programiz, "Bitwiase operation," https://www.programiz.com/c-programming/bitwise-operators, 2020.
[37] M. Rocha and J. Neves, "Preventing premature convergence to local optima in genetic algorithms via random offspring generation," in *Proceedings of the International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE)*, 1999.
[38] ResearchGate, "How to overcome strong local minima in genetic algorithm?" https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm, 2023.
[39] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[40] R. M. C. R. de Souza, D. C. F. Queiroz, and F. J. de A. Cysneiros, "Logistic regression-based pattern classifiers for symbolic interval data," *PATTERN ANALYSIS AND APPLICATIONS*, vol. 14, no. 3, pp. 273–282, 2011.
[41] LabVIEW, "The introduction of labview," https://www.ni.com/en-us/shop/labview.html, 2020.
[42] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *CoRR*, vol. abs/1611.02770, 2016.
[43] F. Wang, J. Han, F. Lin, and K. Ren, "Wipin: Operation-free passive person identification using wi-fi signals," in *IEEE Global Communications Conference, GLOBECOM*, 2019.
[44] Z. Chen, H. Zou, J. Yang, H. Jiang, and L. Xie, "Wifi fingerprinting indoor localization using local feature-based deep LSTM," *IEEE Syst. J.*, vol. 14, no. 2, pp. 3001–3010, 2020.
[45] A. Vidhya, "Anomaly detection using isolation forest – a complete guide," https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/, 2021.
[46] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. P. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
[47] S. Shen, H. Wang, and R. R. Choudhury, "I am a smartwatch and I can track my user's arm," in *ACM International Conference on Mobile Systems, Applications, and Services, MobiSys*, 2016.
[48] Z. Shi, J. A. Zhang, R. Y. Xu, and Q. Cheng, "Environment-robust device-free human activity recognition with channel-state-information enhancement and one-shot learning," *IEEE Transactions on Mobile Computing*, vol. 21, no. 2, pp. 540–554.
[49] K. Chetty, G. E. Smith, and K. Woodbridge, "Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1218–1226, 2012.
[50] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *ACM International Conference on Mobile Computing and Networking, MobiCom*, 2015.
[51] B. Chen, V. Yenamandra, and K. Srinivasan, "Tracking keystrokes using wireless signals," in *ACM International Conference on Mobile Systems, Applications, and Services, MobiSys*, 2015.
[52] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When CSI meets public wifi: Inferring your mobile phone password via wifi signals," in *ACM SIGSAC Conference on Computer and Communications Security, CCS*, 2016.
[53] N. Cheng, X. O. Wang, W. Cheng, P. Mohapatra, and A. Seneviratne, "Characterizing privacy leakage of public wifi networks for users on travel," in *IEEE International Conference on Computer Communications, INFOCOM*, 2013.

Jianwei Liu received the BS degree from Northwestern Polytechnical University in 2018. He received his Master degree from Xi'an Jiaotong University. He is working toward the Ph.D. degree at Zhejiang University. His research interests include RFID, mobile computing, and smart sensing. He is student member of the IEEE.

Yinghui He received the B.S.E. degree in information engineering from Zhejiang University, Hangzhou, China, in 2018. He is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests mainly include mobile edge computing, wireless edge intelligence, and integrated sensing and communication.

Chaowei Xiao received his Ph.D. degree in computer science and engineering from University of Michigan, Ann Arbor in 2020. He is now research scientist at Nvidia Research and an assistant professor at the Arizona State University. His research interests focus on Trustworthy Machine Learning.

Jinsong Han received his Ph.D. degree from Hong Kong University of Science and Technology in 2007. He is now a professor at Zhejiang University. He is a senior member of the ACM and IEEE. His research interests focus on IoT security, smart sensing, wireless and mobile computing.

Kui Ren received the Ph.D. degree from the Worcester Polytechnic Institute, Worcester, MA, USA. He is currently a Professor of computer science and technology and the Director of the Institute of Cyberspace Research, Zhejiang University, Hangzhou, Zhejiang, China. His current research interests include cloud and outsourcing security, wireless and wearable system security, and artificial intelligence security. Dr. Ren is also a Distinguished Scientist and Fellow of the ACM. He was a recipient of the IEEE CISTC Technical Recognition Award 2017 and the NSF CAREER Award in 2011.