Evasion Attacks and Countermeasures in Deep Learning-Based Wi-Fi Gesture Recognition

Guolin Yin, Junqing Zhang, *Senior Member, IEEE*, Xinping Yi, and Xuyu Wang

Abstract—Deep learning-based Wi-Fi sensing has received massive interest thanks to the prevalence of Wi-Fi technology. While deep learning techniques provide promising results in Wi-Fi sensing, there are only very few studies on the vulnerabilities against Wi-Fi ensing. In this paper, we studied evasion attacks against deep learning-based Wi-Fi sensing and the countermeasure and conducted an extensive experimental evaluation using two publicly available datasets, namely SignFi and Widar. Accordingly, we proposed three white-box and two black-box attacks and revealed that even with an undetectable power change, evasion attacks can achieve a remarkable attack success rate (ASR) of 97.0% and 95.6% in white-box and black-box settings, respectively. These results highlight the urgent need for countermeasures against evasion attacks in Wi-Fi sensing systems. We introduced adversarial training and randomised smoothing, which notably improved the robustness of the Wi-Fi sensing model. The ASRs for white-box and black-box attacks were reduced to a minimum of around 6% and 2%, respectively. Moreover, randomised smoothing also introduced certifiable robustness, achieving 70.1% of samples certified for our model. The certification method provides an additional layer of reliability, ensuring that the model's performance remains consistent and predictable even under adversarial conditions.

Index Terms—Wi-Fi sensing, deep learning, adversarial attacks, adversarial training, randomised smoothing

1 Introduction

Wi-Fi sensing has attracted considerable research interest from academia and industry because Wi-Fi is widespread and integrated into many consumer products, including computers, smartphones, tablets, Fitbits, and smart home appliances, to name but a few. Wi-Fi sensing encompasses a broad range of applications, including large-scale movements like human activity recognition [1], fall detection [2], and person identification based on posture [3] and gait [4] as well as small-scale motions like gesture recognition [5], [6] and sign language recognition [7]. Gesture recognition has emerged as an important application in Wi-Fi sensing [8].

Wi-Fi sensing provides distinct advantages over camera-based and wearable-based methods due to its non-intrusive and privacy-preserving nature. Unlike cameras, which may be unsuitable for private spaces [9] like bedrooms, Wi-Fi sensing enables applications such as gesture recognition, fall detection, and activity monitoring while safeguarding user privacy, even in low-light conditions where cameras fail. Furthermore, compared to wearable-based systems, Wi-Fi sensing eliminates the need for users to carry or wear devices [9], making it particularly suitable for elder care scenarios where continuous monitoring is essential but compliance with wearable use may be inconsistent or impractical [10].

Deep Neural Networks (DNNs) have been widely used for Wi-Fi sensing gesture recognition, thanks to their excellent feature extraction capability [5], [8], [11]. While DNN has shown promising performance for Wi-Fi sensing research, the security vulnerabilities of DNN models have become a concern in recent years. In particular, evasion attacks can mislead the DNN model to make a wrong prediction in the inference phase with adversarial perturbations that are applied to the model's input [12], [13]. Initially studied in the area of image classification [12], evasion attacks are extensively studied in various computer vision areas, such as object detection [14] and semantic segmentation [15]. Furthermore, the work in [16] highlights the application and impact of these adversarial methods in broader domains, including cybersecurity.

Evasion attacks have also been applied to Wi-Fi sensing, although research in this domain is still in its early stages. The latest works, such as [17] and [18], explored the evasion attack with fast gradient sign method (FGSM) and projected gradient descent (PGD). Their work primarily focused on the impact of the attack on joint communication and attack performance. However, they need to access the model input and true label for adversarial sample generation. This may not be practical in real-world scenarios. The authors in [19] investigated the adversarial robustness of Wi-Fi sensing model under evasion attacks and introduced adversarial

Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. The work of J. Zhang was in part supported by the UK Engineering and Physical Sciences Research Council (EPSRC) New Investigator Award under grant ID EP/V027697/1. The work of X. Yi was supported by the National Natural Science Foundation of China under Grant 62471129. *The review of this paper was coordinated by xxx. (Corresponding author: Junqing Zhang, Xinping Yi)*

- *G. Yin and J. Zhang are with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom. (email: {Guolin.Yin, Junqing.Zhang}@liverpool.ac.uk)*
- *X. Yi is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China. (email: xyi@seu.edu.cn)*
- *X. Wang is with the Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL 33199, US. (email: xuywang@fiu.edu)*

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier xxx

training as countermeasures. However, adversarial training may fail against attack methods that are never seen during adversarial training.

In this paper, we systematically evaluate the impact of evasion attacks on deep learning-based Wi-Fi gesture recognition models as well as countermeasures. We conduct a comprehensive analysis of various attack methods and the adversarial robustness of Wi-Fi sensing models. Experimental evaluations using SignFi [7] and Widar [5] datasets demonstrate the effectiveness of our methods, offering a robustness benchmark for Wi-Fi sensing models under evasion attacks. In particular, in terms of the attack methods, we explored both white-box and black-box attacks, using different perturbation generation approaches.

- White-box attack: We utilised perturbation generation methods like FGSM [20], PGD [21], and DeepFool [22]. We implemented both non-targeted (FGSM, PGD, and DeepFool) and targeted (FGSM and PGD) strategies, which notably achieved an attack success rate (ASR) of up to 97% and 82%, respectively, with low perturbation-to-signal ratio (PSR).
- Black-box attack: We employed universal adversarial perturbation (UAP) [23] and exploited the transferability of adversarial samples. We focused on the class-invariant characteristics of Wi-Fi sensing samples and achieved a peak ASR of 95.6% across various environments and models.
- Black-box attack: Additionally, we considered a more practical scenario where the attacker is unaware of the victim system's information but can eavesdrop on all sensing signals. The attacker uses an unsupervised k-means clustering algorithm to construct pseudo-labels to train a surrogate model and achieved an average ASR of 80% in this challenging scenario.

Because evasion attacks show strong potential in attacking deep learning-based gesture recognition systems, it is essential to design countermeasures. In this paper, we adopted adversarial training [20] and randomised smoothing [24] to bolster model robustness against these attacks.

- Defence by adversarial training: Adversarial training significantly mitigated the impact of these attacks, reducing the ASR to as low as 6% and 15.9% under FGSM and PGD attacks, respectively. For the DeepFool attack, the ASR dropped to a minimum of 45.6%. Additionally, our model exhibited strong defences in black-box scenarios, reducing the ASR to around 2%.
- Defence by randomised smoothing: The smoothed classifier reduced the ASR to 7% for FGSM, 29.1% for PGD, and 13.8% for Deepfool attacks, illustrating a significant enhancement in robustness. Randomised smoothing is particularly effective against black-box attacks utilising UAP, where the ASR was maintained below 2%.
- Certification by randomised smoothing: Besides the defence, randomised smoothing can certify the robustness of Wi-Fi sensing models. This method provides a certified defence against adversarial attacks, offering a quantifiable and certified radius within which the model's robustness is guaranteed. This certification enhances the model's reliability against unknown threats and facilitates comparative analysis with other Wi-Fi sensing models. Our experiment shows the randomised smooth method can improve the approximate certified test set accuracy (ACTS) of the sensing model to 70.1% when the PSR threshold was set to 0.5×10^{-4}.

The code can be accessed online.[1]

The structure of this paper is organised as follows: Section 9 presente the related works. Section 2 introduces Wi-Fi sensing, followed by the system model in Section 3. We discuss white-box and black-box attack methods in Sections 4 and 5, and outline defense strategies like adversarial training and randomised smoothing in Sections 6 and 7. Section 8 details the experimental evaluation. We conclude the paper in Section 10.

2 Wi-Fi Sensing Primer

Wi-Fi sensing has received massive research interest in recent years. This paper uses gesture recognition as an example. Performing a gesture involves hand movements, which will affect the propagation paths of Wi-Fi signals. OFDM is commonly adopted by IEEE 802.11 a/g/n/ac/ax, which can provide fine-grained channel state information (CSI). During the course of performing a gesture, a Wi-Fi transmitter will continuously send packets to a receiver which can estimate a series of CSI over time, representing variations caused by gesture movements.

Different gestures involve unique hand movement patterns, which will cause different effects on CSI variations. Gesture recognition can be designed by constructing and extracting unique features for each gesture by analysing the specific hand and body movements. Such approaches are named feature engineering-based approaches [25]. The challenge for the feature engineering-based method lies in the concurrent and unpredictable movements of various body parts, which introduce complex impacts in the sensing signals. Such intricacies hinder their development to accurately represent these behaviours [25]. In contrast, DNN can automatically learn the feature mapping between the gesture and the CSI variations, which can eliminate the need for hand-crafted features. Therefore, DNN has been widely used in gesture recognition [8].

As shown in Fig. 1, DNN-based Wi-Fi sensing involves two stages, namely training and inference. The training dataset is denoted as $\{\mathcal{X}_{tra}, \mathcal{Y}_{tra}\}$ with $\mathcal{X}_{tra} = \{x_{tra,1}, x_{tra,2}, \ldots, x_{tra,N}\}$ and its corresponding labels $\mathcal{Y}_{tra} = \{y_{tra,1}, y_{tra,2}, \ldots, y_{tra,N}\}$, N is the size of the dataset. Each sample in $\mathcal{X}_{tra}$ is a time series of CSI amplitude, i.e., $x_{tra,i} = \{H_1(i), H_2(i), \ldots, H_T(i)\}$. Each CSI in a sample has a dimension of $S \times M$, where S is the number of subcarriers, and $M = N_{rx} \times N_{tx}$, where N_{rx} and N_{rx} represent the numbers of receiver and transmitter antennas, respectively. Therefore, each sample is a tensor $x \in \mathbb{R}^{T \times S \times M}$. The DNN model, $f(\cdot)$, can be trained using $\{\mathcal{X}_{tra}, \mathcal{Y}_{tra}\}$ in a supervised learning manner, given as

$$\min \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(x_{tra,i}), y_{tra,i}), \tag{1}$$

1. https://github.com/Guolin-Yin/Attack_WiFi_Sensing. The code will be made available upon acceptance of this paper.

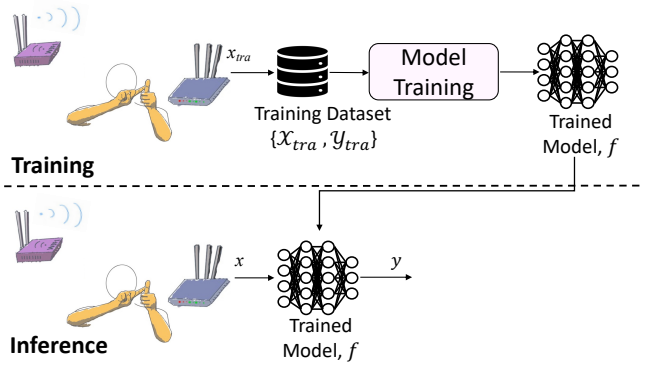


Fig. 1: Deep learning-based Wi-Fi sensing.

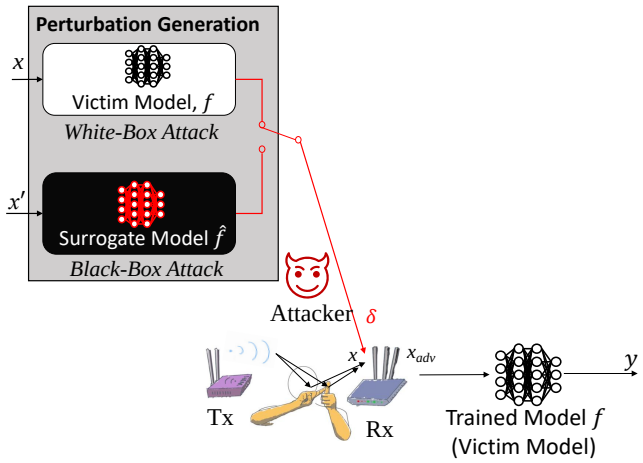


Fig. 2: Evasion attacks to the inference stage of Wi-Fi sensing.

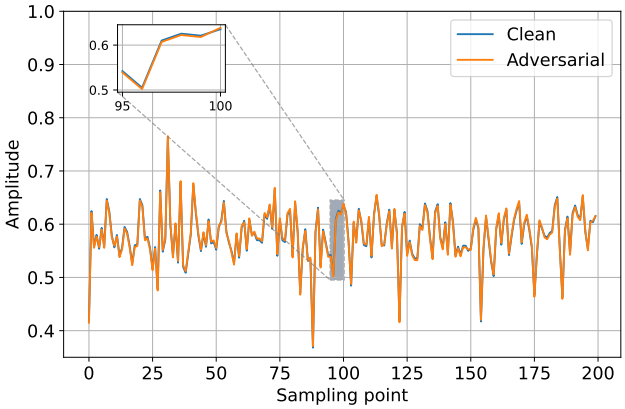


Fig. 3: Comparison of the clean and adversarial waveforms.

where $\mathcal{L}(\cdot,\cdot)$ is the chosen loss function, $f(x_{tra,i})$ is the prediction for sample i in the training dataset by the DNN model, and $y_{tra,i}$ is its corresponding true label.

During the inference phase, the receiver records the time series of CSI sample x during the gesture performing. x has the same dimension as the training samples, i.e., $x \in \mathbb{R}^{T\times S\times M}$. The pre-trained deep learning model f predicts the label y as

$$y = \arg\max_t f^t(x), \tag{2}$$

where $f^t(x)$ is the t^{th} output of the model f.

3 System Model

3.1 Evasion Attacks

The DNN model provides an effective way to learn complicated features from a complex data structure and tackle the complex classification problem. However, DNN models are subject to evasion attacks, which take place at the inference stage of deep learning to manipulate the neural network prediction. The attacker crafts a perturbation, termed an "adversarial sample" [12], aimed at deceiving a deep learning model to make inaccurate predictions. Under the attack, the DNN's output probability distribution is skewed, heavily favouring the wrong outcome.

In this paper, we will apply evasion attacks to a Wi-Fi gesture recognition system. As shown in Fig. 2, the system consists of a transmitter, a receiver, and an attacker. The legitimate receiver has a pre-trained DNN model, also referred to as the victim model in this paper.

When an attack occurs, the received signal, x_{adv}, referred to as the adversarial sample in the attack context, becomes

$$x_{adv} = x + \delta, \tag{3}$$

where x is the signal from the legitimate transmitter and δ is the received perturbation.

To constrain the intensity of the perturbation, we define PSR as the ratio of the power of the perturbation, P_δ, to the power of the signal, P_x, given as

$$\xi = \frac{P_\delta}{P_x}. \tag{4}$$

The PSR should be kept as low as possible, i.e., $\xi \ll 1$, to ensure undetectability and minimal impact on normal Wi-Fi communication functionality [17]. Research in [18] shows that higher PSR increases Bit Error Rate (BER), making adversarial attacks more conspicuous. Sudden BER changes could potentially serve as a indicator as detection. Thus, we would like to maintain low PSR during attack a Wi-Fi sensing model. As exemplified in Fig. 3, the clean signal and adversarial samples ($PSR = 1 \times 10^{-4}$) almost overlap. Though small, such carefully crafted perturbations can mislead the model into making incorrect predictions.

This paper considers the digital attack [26], which is the worst-case scenario. In digital attacks, an attacker can directly manipulate the model's input, which can allow attackers to bypass the natural barriers of over-the-air transmission, such as interference and multipath fading. Thus, direct manipulation poses the greatest threat, as it enables precise and covert alterations to the system's perception.

Evasion attacks can pose a huge security risk to sensing systems [27].In smart home settings, Wi-Fi-based gesture recognition can be utilised for functions such as lighting, entertainment, locking/unlocking doors, etc. These applications offer user convenience but also create potential security loopholes. Adversaries could exploit these vulnerabilities, particularly in the security domain, by altering gesture recognition models. For instance, they might reconfigure gesture commands to unlock doors without triggering

alarms. If such evasion attacks succeed, they could facilitate unauthorised access, leading to theft, surveillance, or even broader network security breaches in the home. These risks highlight the need for robust defence mechanisms in smart home technology to safeguard user privacy and safety, emphasising the importance of continued research and development in securing these systems against adversarial threats.

Evasion attacks are categorised into targeted and non-targeted attacks based on the attacker's goal over the target class, as discussed in Section 3.2. They can also be classified into white and black-box attacks based on the attacker's knowledge, which will be explained in Section 3.3.

3.2 Non-targeted and Targeted Attacks

3.2.1 Non-targeted Attack

The goal of non-targeted attacks is to find a perturbation, δ, that maximises the loss function, expressed by

$$\begin{aligned} \max_{\delta} \quad & \mathcal{L}(f(x+\delta), y^{true}), \\ & \text{s.t. } P_\delta \leq P_{max}. \end{aligned} \tag{5}$$

Here, we have no control over the class to which the input CSI will be classified. In other words, the output can be any other class, which can be mathematically given as [28]

$$y^{true} \neq \arg\max_t f^t(x+\delta). \tag{6}$$

3.2.2 Targeted Attack

The targeted attack aims to mislead the victim model to a specific target label y^{target}. It is achieved by solving the following optimisation problem:

$$\begin{aligned} \min_{\delta} \quad & \mathcal{L}(f(x+\delta), y^{target}), \\ & \text{s.t. } P_\delta \leq P_{max}. \end{aligned} \tag{7}$$

Here, we have control over the class to which the input CSI will be classified, mathematically given as

$$y^{target} = \arg\max_t f^t(x+\delta). \tag{8}$$

3.3 White-Box and Black-Box Attacks

As shown in Fig. 2, the attacker attacks the victim sensing model by generating a perturbation sample δ using the perturbation generation module. The purpose of this research is to develop adversarial sample generation algorithms capable of fooling Wi-Fi sensing systems in a variety of settings. The perturbation generation module requires a DNN model, denoted as $\hat{f}$. Depending on the knowledge that the attacker has about the victim system, evasion attacks can be categorised as white-box and black-box attacks.

3.3.1 White-Box Attack

In a white-box setting, the attacker has complete knowledge of the victim model f, model input x and its label y. The perturbation generation model of the attacker is exactly the same as the victim model, i.e., $\hat{f} = f$. This attack is presented in Section 4.

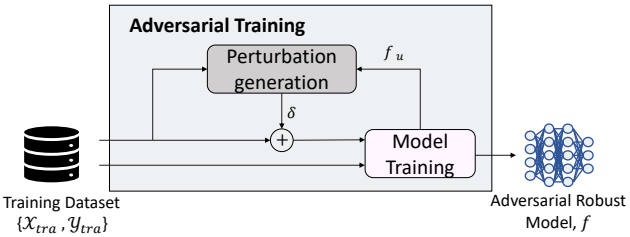


Fig. 4: Block diagram of adversarial training.

3.3.2 Black-Box Attack

In a black-box setting, the attacker lacks knowledge about the victim system. Firstly, the attacker needs to obtain the perturbation generation model ($\hat{f}$), referred to as the surrogate model, independently. Secondly, the attacker does not have access to the victim system's training dataset.

Depending on the knowledge of the label space of the victim system, we propose two black-box attacks such that the attacker can gather a surrogate dataset ($\mathcal{X}'_{tra}$) and then train a surrogate model for perturbation generation.

- **Surrogate dataset with true labels (Section 5.2).** The attacker is aware of the true labels, $\mathcal{Y}_{tra}$, i.e., $\mathcal{Y}'_{tra}$ and $\mathcal{Y}_{tra}$ share same label space.
- **Surrogate dataset with pseudo labels (Section 5.3).** The attacker is unaware of the tasks on which the victim model is trained, i.e., $\mathcal{Y}_{tra}$ is unknown to the attacker. The victim sensing system is a complete black-box to the attacker. In this case, an unsupervised clustering technique is used to construct a pseudo-label $\hat{\mathcal{Y}}'_{tra}$.

3.4 Defence by Adversarial Training

Adversarial training is a technique proposed for deep learning models to enhance their robustness against evasion attacks. As shown in Fig. 4, adversarial examples are first generated from the training data, and then the model is trained on both clean and adversarial examples. The updated weight version of the model f_u will be used for generating new perturbations for the next round of training until the model is converged.

3.5 Defence by Randomised Smoothing

Randomised smoothing enhances the robustness of machine learning models, particularly DNNs, against adversarial perturbations. Randomness is introduced into the model's input during both training and inference, which leads to smoother decision boundaries and improved robustness.

During the training process, the base classifier f is trained on noisy variants of the original clean input data, which enables the model to learn a smooth decision boundary. In other words, the base classifier training resembles adversarial training; however, rather than augmenting the dataset with dedicated adversarial perturbations, it incorporates Gaussian noise to augment the training dataset.

Subsequently, randomised smoothing produces a smoothed classifier g from the base classifier f. For an input x, it creates multiple noisy instances of x by adding Gaussian noise. The base classifier then processes each noisy variant. The smoothed classifier will output the most frequently

predicted class. We denote the number of noisy variants as K. Formally, for the base classifier f, the corresponding smoothed classifier g is constructed as [24]

$$g(x) = \arg\max_{y \in \mathcal{Y}} \mathbb{P}(f(x+\varepsilon) = y), \tag{9}$$

where ε is Gaussian noise and $\varepsilon \sim \mathcal{N}\left(0, \sigma^2 I\right)$.

3.6 Certification by Randomised Smoothing

The certification is a process of quantifying the robustness of a model with a certified radius r. The term "certified radius" refers to a measure of robustness against adversarial perturbations. Specifically, for a given input x, the certified radius denotes the maximum amount by which the input can be perturbed (within that radius) while still ensuring a consistent model prediction. In simpler terms, assume there is a data point x and a classifier that makes a certain prediction for x, a certified radius r guarantees that any point within the L_2 ball of radius r centred at x will receive the same prediction from the classifier, regardless of adversarial manipulations. Mathematically, the certified radius r ensures that for any perturbation δ with $\|\delta\|_2 < r$, the prediction for $x + \delta$ remains y:

$$g(x+\delta) = y \text{ for all } \delta \quad \text{such that} \quad \|\delta\|_2 < r. \tag{10}$$

3.7 Threat Model

In this work, we consider an evasion attack where the adversary's end goal is to manipulate the input of the sensing model into making erroneous decisions. In other words, the adversary maximises the error rate of the sensing model. To simulate the worst-case adversarial scenario, we assume the adversary has knowledge of the victim model's input size and can manipulate the input of the model directly.

We consider two different scenarios based on the knowledge of the adversary. In the white-box approach, the adversary has full knowledge of the sensing model, including its structure, and parameters. Additionally, in the black-box approach, the adversary has limited information. Specifically, the adversary only has access to the label space $\mathcal{Y}$ that the victim model was trained on, but lacks details about the model's parameters, the training dataset, and live model inputs during the testing phase. Finally, we even remove the assumption of known label space of adversaries.

4 White-box Attacks in Wi-Fi Sensing

4.1 Overview

During the inference phase of a Wi-Fi sensing system, the attacker obtains the input x_i to craft the adversarial perturbation δ_i to fool the model in the receiver. For a set of inference input $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$, a corresponding set of perturbation $\Delta = \{\delta_1, \delta_2, \ldots, \delta_N\}$ is crafted for each input, which is referred to as the input-dependent perturbation. The perturbations are small enough so that they are imperceptible to a detector, but they can precisely lead the classifier to make a wrong decision. There are different approaches for perturbation generation. This paper focuses on FGSM, PGD, and DeepFool.

4.2 Fast Gradient Sign Method (FGSM)

FGSM [20] is a "one step at a time" strategy that creates an adversarial example by pushing the clean example into a gradient-ascent direction. The attacker calculates the gradient of the loss function with respect to the input x. For a given input x and a pre-trained DNN model f, the perturbation is obtained by computing the gradient with respect to the input x and then taking the sign of the gradient. Mathematically given as

$$g := (-1)^a \text{sign}(\nabla_x \mathcal{L}(f(x), y)), \tag{11}$$

where the variable y represents either the true label (y^{true}) for non-targeted attacks or the target label (y^{target}) for targeted attacks. The choice is determined by the value of a. Specifically, $a = 0$ implies the use of y^{true} for non-target attack, while $a = 1$ implies the use of y^{target} for targeted attack. The gradient is calculated with respect to the input x, and the sign function subsequently extracts the direction of change for each input feature.

To maintain the PSR of the attack system, the perturbation can be constructed as

$$\delta = \sqrt{\frac{\xi \cdot P_x}{P_g}} \cdot g. \tag{12}$$

We can then have control over the intensity of the attack to the original signal.

4.3 Projected Gradient Descent (PGD)

The PGD [21] is an iterative variation of the FGSM, shown in Algorithm 1. Different from FGSM, PGD optimises the objective function more carefully and adjusts the perturbation δ in the direction gradient iteratively with a smaller step size instead of one large step.

In each iteration, the gradient with respect to the input x was calculated depending on different attacker's goals, i.e., non-targeted and targeted attacks, and they correspond to the objective functions (5) and (7), respectively. Unlike FGSM, which directly utilises the sign of the computed gradient, the PGD attack constructs the perturbation $\hat{g}$ in each step without taking the sign of the gradient. Different from the PGD in [21], we use PSR of $\frac{\xi}{N_I}$ in each iteration to constrain the power of $\hat{g}$, i.e., step 7 in Algorithm 1. We obtain the perturbation g accumulatively with N_I iterations. The final perturbation δ is obtained by constraining the power of g to the given PSR ξ (step 8).

4.4 DeepFool

DeepFool [22] is a hyperplane-based non-targeted adversarial attack method. The hyperplane is the basis for achieving the classification. To change the classification label of an input x, DeepFool searches for the nearest decision boundary. The minimum perturbation for the sample is the distance between this sample and the orthogonal projection point on the linear approximation of the decision boundary. By computing the distance that pushes the input to the decision boundary, the minimal perturbation is determined. The procedure of DeepFool is given in Algorithm 2.

Algorithm 1: PGD Algorithm

Input: CSI x, label y^{true} or y^{target}, model f, PSR ξ, Number of iteration N_I
Output: Perturbation δ

$g = 0$
for i *in range* N_I **do**
  if *Non-targeted attack* **then**
    $\hat{g} = \nabla_x \mathcal{L}\left(f(x+g), y^{true}\right)$
  else if *Targeted attack* **then**
    $\hat{g} = -\nabla_x \mathcal{L}\left(f(x+g), y^{target}\right)$
  $g = g + \sqrt{\frac{\xi \cdot P_x}{N_I P_{\hat{g}}}}\hat{g}$
$\delta = \sqrt{\frac{\xi \cdot P_x}{P_g}} \cdot g$
return δ

Algorithm 2: DeepFool Algorithm

Input: CSI x, model f
Output: Perturbation δ

$x_0 = x$
$i = 0$
$p = \infty$
while $\hat{p}(x_0) = \hat{p}(x_i)$ **do**
  for $t \neq \hat{t}(x_0)$ **do**
    $r = f^t(x_i) - f^{\hat{t}(x_i)}(x_i)$
    $g' = \nabla f^t(x_i) - \nabla f^{\hat{t}(x_i)}(x_i)$
    $p' = \frac{r}{g'}$
    if $p' < p$ **then**
      $p = p'$
      $g = g'$
  $\delta_i = \frac{p}{\|g\|_2^2} g$
  $x_{i+1} = x_i + \delta_i$
  $i = i + 1$
return $\delta = \sum_i \delta_i$

The classification is done by following mapping from input x and output label prediction $\hat{t}(x)$:

$$\hat{t}(x) = \arg\max_t f^t(x). \tag{13}$$

Let $\hat{p}(x_0)$ be the index of the closest hyperplane, found as

$$\hat{p}(x_0) = \arg\min_{t \neq \hat{t}(x_0)} \frac{\left| f^t(x_0) - f^{\hat{t}(x_0)}(x_0) \right|}{\left\| \nabla f^t(x_0) - \nabla f^{\hat{t}(x_0)}(x_0) \right\|_2}. \tag{14}$$

The minimum perturbation is found by computing the projection of the input x to the closest hyperplane, given as

$$\delta = \frac{\left| f^{\hat{p}(x_0)}(x_0) - f^{\hat{t}(x_0)}(x_0) \right|}{\left\| \nabla f^{\hat{p}(x_0)}(x_0) - \nabla f^{\hat{t}(x_0)}(x_0) \right\|_2^2} \times (\nabla f^{\hat{p}(x_0)}(x_0) - \nabla f^{\hat{t}(x_0)}(x_0)). \tag{15}$$

5 Black-box Attack in Wi-Fi Sensing

5.1 Overview

The black-box evasion attack is more practical for attackers, as it does not require them to have full knowledge of the victim system, which is a strong assumption in white-box attacks. Notably, Wi-Fi data is highly sensitive to environmental variations due to the impact of multipath propagation. The primary goal of this section is to develop evasion attack techniques that are capable of deceiving the victim model, independent of the environments, as well as the DNN inputs and architectures.

In contrast to white-box attacks, a black-box attack assumes that the attacker does not have access to the victim model f or its input x. Therefore, the attacker cannot use the victim model to compute the adversarial perturbation δ. However, in practice, the attacker can conduct their own experiments to collect data and train a surrogate model for perturbation generation.

Furthermore, the adversarial perturbation needs to be independent of the input in a black-box attack, as the attacker has no access to the model's input, which distinguishes it from white-box attacks. Let x_i be a set of all possible inputs to the victim model with their corresponding true labels y_i^{true}. A universal perturbation δ_U must be generated to deceive the victim model with any input, given as

$$y_i^{true} \neq f(x_i + \delta_U). \tag{16}$$

The surrogate model training and perturbation generation are elaborated in Section 5.2 when the true label space is available. Section 5.3 relaxes the assumption even further, in which, the attacker has no access to the true label space of the victim sensing system. We propose a pseudo-label generation method to train the surrogate model with pseudo-labels.

5.2 Black-box Attack Using True Labels

5.2.1 Surrogate Model and Dataset

We assumed that the attacker possesses prior knowledge regarding the specific gestures that the victim model has been trained on, denoted as the label space. The attacker is then able to carry out independent experiments by performing these gestures to obtain a surrogate dataset. We symbolised the surrogate dataset as $\{\mathcal{X}'_{tra}, \mathcal{Y}'_{tra}\}$, where the $\mathcal{X}'_{tra} = \{x'_{tra,1}, x'_{tra,2}, \ldots, x'_{tra,N}\}$ and $\mathcal{Y}'_{tra} = \{y'_{tra,1}, y'_{tra,2}, \ldots, y'_{tra,N}\}$. Because the adversary has no information about the victim's deep learning model, it will create its own deep learning model, i.e., the surrogate model. Finally, the adversary will train the surrogate model using the surrogate dataset.

5.2.2 Perturbation Generation

To obtain an adversarial perturbation, an attacker can use a trained surrogate model as shown in Algorithm 3. The perturbation δ is initialised as 0 and incrementally built over each sample in the surrogate dataset $\mathcal{X}'_{tra}$. For each data sample, we assess whether the perturbation δ leads to a change in the prediction of the classifier $\hat{f}$. If it does not, we seek the smallest possible perturbation that would move the data sample across the decision boundary. This minimal

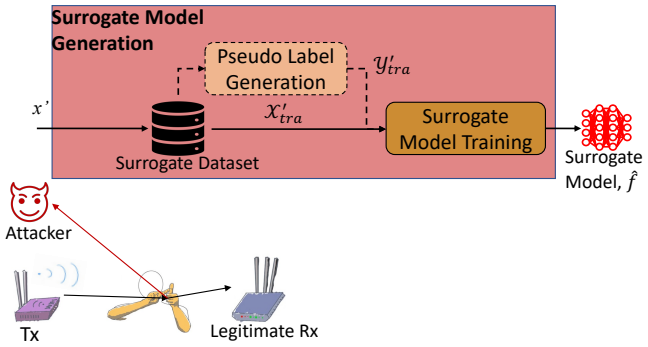


Fig. 5: Training the surrogate model for black-box attacks.

Algorithm 3: UAP Algorithm

Input: Dataset $\mathcal{X}'_{tra}$, model $\hat{f}$, desired ASR $\mathcal{D}$, PSR ξ
Output: UAP δ_U

$\delta = 0$
while $ASR < \mathcal{D}$ **do**
  for *each value* $x'_{tra,i}$ *in* $\mathcal{X}'_{tra}$ **do**
    if $\hat{f}\left(x'_{tra,i} + \delta\right) == \hat{f}\left(x'_{tra,i}\right)$ **then**
      Calculate the smallest perturbation that will cause $x'_{tra,i} + \delta$ to approach the decision boundary:
      $\Delta\delta_i = DeepFool(x'_{tra,i} + \delta, \hat{f})$
      Update the perturbation: $\delta = \delta + \Delta\delta_i$
$\delta_U = \sqrt{\frac{\xi \bar{P}_{\mathcal{X}'_{tra}}}{P_\delta}} \cdot \delta$
return δ_U

perturbation is denoted as $\Delta\delta_i$. After each iteration, $\Delta\delta_i$ is incorporated into the overall perturbation δ. This process is mathematically represented as follows:

$$\delta := \delta + \Delta\delta_i. \tag{17}$$

In this paper, we use DeepFool from Section 4.4 to estimate the minimal perturbation $\Delta\delta_i$. The algorithm will be terminated when the ASR on the dataset $\mathcal{X}'_{tra}$ reaches the threshold $\mathcal{D}$. The final perturbation δ_U is obtained by projecting it to the desired PSR, and it is given as

$$\delta_U = \sqrt{\frac{\xi \bar{P}_{\mathcal{X}'_{tra}}}{P_\delta}} \cdot \delta, \tag{18}$$

where $\bar{P}_{\mathcal{X}'_{tra}} = \frac{1}{N}\sum_{i=1}^{N} P_{x'_{tra,i}}$ and $P_{x'_{tra,i}}$ is the power of the sample $x'_{tra,i}$. UAP is expected to be effective across different environments and models. Notably, our approach only considers non-target attacks as DeepFool is primarily used for non-target attacks in prior studies [29], [30].

5.3 Black-Box Attack Using Pseudo Labels

5.3.1 Training Surrogate Model with Pseudo-labels

In this section, we relax the assumption that the attacker possesses knowledge of the classes used to train the victim model. and consider a more challenging scenario where the attacker lacks information about the classes on which the victim model is trained.

Due to the broadcast nature of wireless propagation, an attacker within range can receive all the signals of the Wi-Fi sensing system, as shown in Fig. 5. During the victim system's inference phase (the working stage), the attacker can eavesdrop on the reflected sensing signal, and construct $\mathcal{X}'_{tra} = \{x'_{tra,1}, x'_{tra,2}, \ldots, x'_{tra,N}\}$ by locating a small Wi-Fi receiver within the sensing area of the victim system. However, it may not have the label information. Pseudo-labels of the dataset can be generated. We denote the pseudo-label set as $\hat{\mathcal{Y}}'_{tra} = \{\hat{y}'_{tra,1}, \hat{y}'_{tra,2}, \ldots, \hat{y}'_{tra,N}\}$. The surrogate model will be trained on the surrogate dataset $\{\mathcal{X}'_{tra}, \hat{\mathcal{Y}}'_{tra}\}$. The perturbation generation method will be the same as in Section 5.2.

5.3.2 Pseudo Label Generation

Given a set of unlabelled eavesdropped datasets, we propose to create pseudo-labels using k-means clustering, which clusters similar data points together based on their characteristics, and then we assign the pseudo-label to each cluster for surrogate model training. We choose k-means due to its proven effectiveness in conjunction with DNN for discovering data patterns. k-means-generated pseudo-labels train DNN models through self-supervision [31], guiding feature learning in neural networks by creating clusters from unlabelled data, eliminating the need for manual annotations. Besides k-means clustering, alternative advanced unsupervised learning methods could also be employed, such as k-means++ [32], hierarchical clustering, and spectral clustering [33].

The k-means clustering technique organises data into R independent groups based on the feature distribution of each sample. The process involves the following steps.

(a) Define the number of clusters as R.
(b) R cluster centroids initialised by dividing all objects into R clusters randomly.
(c) The distance between centroids of all clusters and each sample will be computed using the Euclidean distance. The centroid-sample distance is computed as follows:

$$d(x'_{tra,i}, \mathcal{C}_r) = \sqrt{\sum_{j=1}^{D}\left(x'_{tra,i,j} - \mathcal{C}_{r,j}\right)^2}, \tag{19}$$

where i denotes the i^{th} sample in the set $\hat{\mathcal{X}}'_{tra}$, and j refers to the j^{th} element in $x'_{tra,i}$, where the total number of elements is $D = T \times S \times M$. $\mathcal{C}_r$ symbolises the r^{th} centroid.
(d) Each sample will be assigned to the cluster with the closest centroid.
(e) Update the centroids of clusters.

Steps (c), (d) and (e) will be repeated until the centroids stop changing. Each data sample $x'_{tra,i}$ will be assigned with the closest cluster $\mathcal{C}_r$. All the samples within the r^{th} cluster will be assigned the same label. Please note that the pseudo labels in this section may not be mapped to the real gestures, but are the index of the clusters.

6 Defence by Adversarial Training

To reduce the impact of the attack, an adversarial training mechanism was employed as a countermeasure against evasion attacks.

In each training epoch, a perturbation generation algorithm will generate the adversarial samples that increase model loss based on the current model's parameters. The training process will minimise the overall loss on top of adversarial and clean samples. The process repeats until the DNN-based model converges. The mathematical equation of adversarial training is given as

$$\min \frac{1}{N}\sum_{i=1}^{N}\{\mathcal{L}(f(x_{tra,i}), y_{tra,i}) + \mathcal{L}(f(x_{tra,i}+\delta_i), y_{tra,i})\}$$
$$\text{s.t. } P_\delta \leq P_{max}, \tag{20}$$

where $x_{tra,i} \in \mathcal{X}_{tra}$ and $y_{tra,i} \in \mathcal{Y}_{tra}$. The adversarial training aims to solve the minimisation problem and obtain a DNN model f over both perturbed and clean training datasets. The training PSR $\xi_{training}$ is the relative power of the perturbation with respect to the original signal during the adversarial training phase, the adversarial samples can be generated using various methods such as FGSM and PGD. By using adversarial samples during the training process, the model is compelled to learn not just the features inherent in clean input data but also the patterns existent in adversarial samples.

7 Defence by Randomised Smoothing

Despite bolstering adversarial robustness, adversarial training falls short in providing concrete robustness guarantees and the defence may fail when encountering a stronger or unknown attack that has not been seen during training. Randomised smoothing [24] is adopted to fix this issue. Specifically, randomised smoothing is a certified defence approach, which brings quantifiable robustness (certified radius). Within this certified radius, corresponding to a specific PSR value, the system's predictions remain constant, unperturbed by adversarial interference.

Randomised smoothing showcases remarkable generalisation capabilities for novel attacks. Owing to its inherent statistical foundation, randomised smoothing exhibits reduced vulnerability to overfitting on particular adversarial perturbation generation techniques. This trait equips the system with an enhanced capability to counteract not just known, but also emergent and novel adversarial attacks.

7.1 Training Base Classifier with Gaussian Noise

Similar to the aforementioned adversarial training, the training dataset can be also augmented with Gaussian noise instead of adversarial perturbation (δ_i). The training process can be mathematically given as

$$\min \frac{1}{N}\sum_{i=1}^{N}\{\mathcal{L}(f(x_{tra,i}), y_{tra,i}) + \mathcal{L}(f(x_{tra,i}+\varepsilon_i), y_{tra,i})\}$$
$$\text{s.t. } P_\varepsilon \leq P_{max}, \tag{21}$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I)$. The relative power of the noise with respect to the signal during training can be also characterised using $\xi_{training}$. By introducing such noise, the training data is perturbed in various directions, ensuring the deep learning model is exposed to a diverse range of input variations. By training the classifier on this augmented dataset, it inherently learns to recognise and correctly classify inputs even in the presence of perturbation. Ultimately, this noise-augmented training strategy enhances the model's ability to generalise across a broader spectrum of input perturbations, leading to increased robustness in multiple directions.

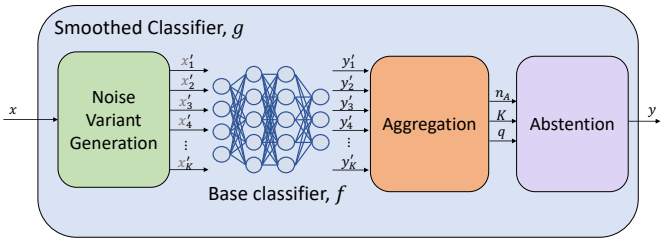


Fig. 6: Inference process of smoothed classifier

7.2 Inference

The overall inference process is shown in Fig. 6, where the smoothed classifier g assigns a label to the input x based on the most frequent label that f predicts among K noisy samples.

- **Noise Variant Generation**: the smoothed classifier generates K noisy samples $x'_i = x + \varepsilon_i$, where each ε_i is drawn from a normal distribution $\mathcal{N}(0, \sigma^2 I)$.
- **Base Classifier Prediction**: the classifier f makes predictions for each of these perturbed inputs, denoted by $y'_i = f(x'_i)$ for $i = 1, 2, ..., K$.
- **Aggregation**: the most frequent label y_A among y'_i will be determined as the predicted label for the input x.
- **Abstention** will determine if the probability of y_A is statistically significant. The classifier g will output the label y_A if p_A is statistically significant, otherwise, the algorithm will abstain.

Abstention in randomised smoothing refers to a technique where a model, instead of making a prediction, chooses to abstain from a response when it is not sufficiently confident about the result by two-sided hypothesis test [24]. According to (10), the smoothed classifier g draws K predictions from the noisy variants of input x. The probability of the most frequent predicted class is denoted as p_A with occurrences n_A. A two-sided hypothesis test evaluates if p_A is statistically significant, with n_A assumed to follow a Binomial distribution with parameters K and success probability q. Let $\texttt{BinomPValue}(\cdot)$ be the function to compute p-value of the two-sided hypothesis. If the p-value from $\texttt{BinomPValue}(n_A, K, q)$ is less than or equal to α (incorrect probability), the classifier confidently predicts the label of the input x. Otherwise, it abstains from the prediction, since it is not statistically significant. The abstention mechanism ensures predictions are statistically significant and reliable, thus enhancing the system's robustness.

7.3 Certification by Randomised Smoothing

In evaluating and certifying the robustness of g around an input x, the most frequent predicted class y_A is first

estimated using $g(x)$ with a limited set of noise variants, i.e., small K. This initial estimation serves as a preliminary assessment. Subsequently, a more extensive set of noise variants is employed to refine the estimation of the lower bound probability $\underline{p_A}$. The upper bound probability of any other class, denoted as $\overline{p_B}$, will be simply deduced as $\overline{p_B} = 1 - \underline{p_A}$. Subsequently, the same abstain mechanism introduced in Section 7.2 will be used to determine if p_A is statistically significant. If a statistically significant determination is made, indicating a robust and confident classification, the procedure proceeds to calculate the certified radius, which is calculated as

$$r = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})), \tag{22}$$

where $\Phi^{-1}(\cdot)$ denotes the inverse of the cumulative distribution function of the standard normal distribution.

To better assess adversarial robustness in Wi-Fi sensing systems, we transitioned from using certified radius, r, to certified PSR, providing a relative measure of perturbation strength against the signal, given as

$$\xi = (\frac{r}{||x||_2})^2. \tag{23}$$

This adjustment ensures a more precise evaluation of the system's robustness to adversarial attacks.

8 Experimental Evaluation

In this section, we first introduced the setup and evaluation metrics in Sections 8.1 and 8.2, respectively. The white-box algorithms were evaluated in Section 8.3. Black-box attacks with true and pseudo labels were evaluated in Sections 8.4 and 8.5, respectively. Before dig into the defence performance, we firstly evaluated the impact of different defence strategy in Section 8.6 following by certification experiment in Section 8.7. The defence performance of adversarial training against white-box, black-box attacks and corresponding complexity analysis was shown in Section 8.8.1, Section 8.8.2, and Section 8.8.3, respectively.

8.1 Setup

A PC was used, equipped with an Intel i7-8700K 3.7 GHz processor, 16 GB of memory, and an NVIDIA GeForce GTX 2080Ti graphics card. We used TensorFlow and Keras for deep learning.

8.1.1 DNN Models

Victim Model: A CNN model is used, denoted as C_D. The architecture is shown in Fig. 7(a), which is revised from the classic AlexNet architecture.

Model for White-Box Attacks: As the attacker is assumed to have access to the victim model, the same model as the victim model is used for white-box attacks.

Surrogate Models for Black-Box Attacks: Three different surrogate models were used. Specifically, the model C_{D_1} is constructed based on the C_D, with different numbers of units in dense layers. The model C_{D_2} is constructed based on the C_{D_1}, with different numbers of filters in all convolutional layers and different filter sizes, maxpooling size. The details of those models are given in Figs. 7(b) and 7(c). A standard VGG19 architecture with revised input and output layers is also used but is not shown due to the space limit.

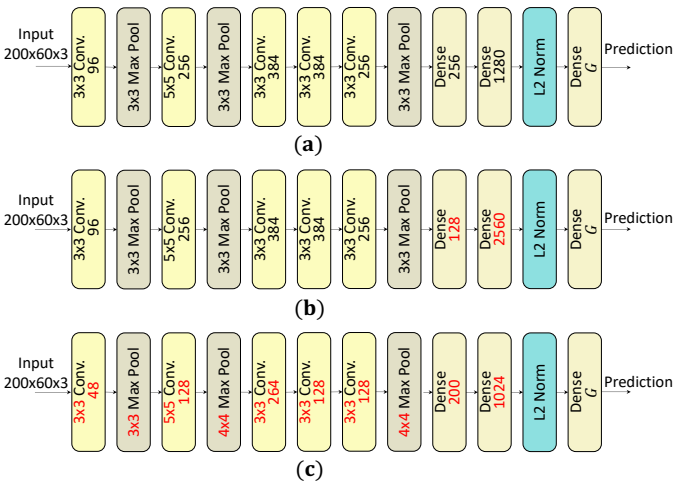


Fig. 7: The architecture of DNN models used. G in the last dense layer is the number of prediction classes. (a), (b), and (c) are model C_D, C_{D_1}, and C_{D_2}, respectively. The different parameters are highlighted in red.

8.1.2 Datasets

This paper used public Wi-Fi sensing datasets, i.e., SignFi [7] and Widar [5]. They are chosen for their relevance to gesture recognition research and practical applications.

- The Widar dataset includes widely recognised gesture classes essential for academic study. It was collected in a classroom environment by three different users, and each user performed six gesture classes ($push\&pull$, $Sweep$, $Clap$, $DrawO$, $Zigzag$, and $DrawN$) 20 times.
- The SignFi dataset offers a diverse range, with 276 classes of sign language, providing a detailed and varied set of gestures for a comprehensive system evaluation. The dataset was collected in both lab and home environments, with 20 and 10 samples per class, respectively.

Both datasets were collected using Intel 5300 Wi-Fi cards. The card has $N_{tx} = 1$ transmitter antenna and $N_{rx} = 3$ receiver antennas, i.e., $M = 3$. The 802.11 CSI tool [34] was used, which reports the CSI of 30 selected subcarrier groups, i.e., $S = 30$. And $T = 200$ sampling points was used. In summary, each input CSI tensor is $x \in \mathbb{R}^{T \times S \times M}$. All datasets were randomly divided into 80% of data for training and 20% of data for testing, respectively.

8.2 Evaluation Metrics

8.2.1 Attack Success Rate

To quantify the effectiveness of the adversarial attack, ASR was used, which is the proportion of the number of successfully attacked samples (N_S) among the number of attacked samples (N_T), defined as

$$ASR = \frac{N_S}{N_T}. \tag{24}$$

8.2.2 Approximate Certified Test Set Accuracy

ACTS is proposed to quantify the robustness of models in [24]. In the original ACTS definition, a certified radius, r, is used to compute the ACTS. In this paper, to better characterise the relationship between the certified range and signal strength, we proposed to use certified PSR instead, defined in (23).

ACTS represents the proportion of test samples that the model accurately classifies without abstaining and certifies with a certified PSR larger than the predefined PSR threshold (ξ_τ). The ACTS is mathematically defined as

$$\text{ACTS} = \frac{1}{N_c} \sum_{i=1}^{N_c} C_i, \tag{25}$$

where N_c is the total number of certified samples in the dataset, C_i is a binary variable defined as

$$C_i = \begin{cases} 1 & \text{if } \xi_i \geq \xi_\tau \\ 0 & \text{if } \xi_i < \xi_\tau. \end{cases} \tag{26}$$

8.3 White-Box Attack

8.3.1 Non-Targeted White-Box Attacks

In this section, the performance of FGSM, PGD, and DeepFool was evaluated on the SignFi dataset. As a baseline, we also considered a Gaussian noise attack [27], and the Gaussian noise perturbations were randomly generated according to a Gaussian distribution. To demonstrate the effectiveness of the evasion attacks, we examined them under low PSR levels in the range of PSR$=0$ to 5×10^{-4}. Clean samples were used when PSR$=0$.

In Fig. 8, all the attack algorithms reduce the victim system's performance significantly at a low level of PSR. When the PSR was set to 5×10^{-4}, the ASR of FGSM was found to be 61.4%; PGD with one, two, and three iterations achieved ASRs of 86.3%, 86.3%, and 89.5%, respectively. The notation "PGD-N_I" represents PGD with N_I iterations. These results confirm that PGD outperforms FGSM, regardless of the number of iterations, due to the more meticulous crafting of perturbations by PGD. Notably, DeepFool achieved an ASR of 91.2% under the same conditions. These results demonstrate that increasing the number of iterations of PGD leads to stronger attacks, with the performance at two iterations comparable to that of DeepFool. The Gaussian noise attack, has no negative impact on the victim model, as the power of the perturbation is extremely limited. In comparison to Gaussian noise attacks, evasion attacks are more power-efficient and can cause a significant performance decrease in the victim system.

8.3.2 Targeted White-Box Attacks

This subsection examined targeted FGSM and PGD using the Widar datasets. As shown in Fig. 9, the PGD attack yielded superior results compared to the FGSM attack. Conversely, the efficacy of the Gaussian noise attack had a negligible impact on the performance of the victim model, with an ASR of nearly zero.

To demonstrate the targeted attack's effect, $PSR = 6.7 \times 10^{-4}$ and $y^{target} = 4$ was chosen, which corresponds to the class "$DrawO$" as an example. The ASR of the FGSM and PGD are 21% and 78.8%, respectively. The confusion matrix is depicted in Fig. 10. The PGD approach causes the model to incorrectly classify the majority of examples to the targeted class, and the FGSM performs poorly at such a low PSR. It is worth noting that targeted attacks require higher perturbation levels than non-targeted attacks since the targeted attack involves manipulating the input in a specific way to cause the model to misclassify it as a particular class. In contrast, a non-targeted attack only requires causing the model to make any kind of misclassification.

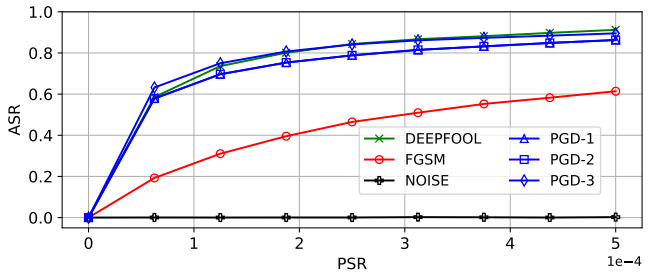


Fig. 8: Performance comparison of non-targeted white-box attacks. FGSM, PGD, and DeepFool are studied.

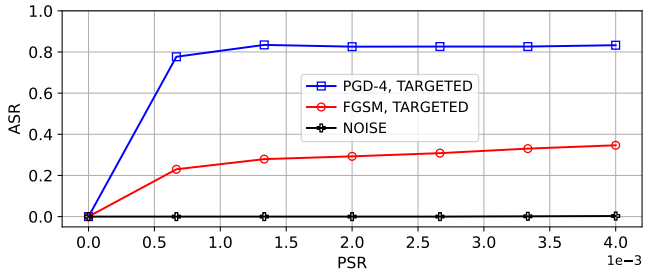
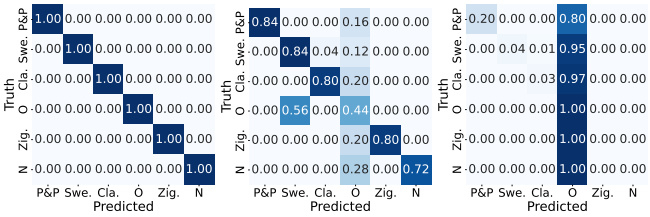


Fig. 9: Performance comparison of targeted white-box attacks. FGSM and PGD are studied.



(a) (b) (c)

Fig. 10: The confusion matrix of attack-free and different white-box target-attack approaches. (a) Attack-free. Overall accuracy: 100%. (b) Targeted FGSM attack with PSR = 6.7×10^{-4}. Overall ASR: 21%. (c) Targeted PGD attack with PSR = 6.7×10^{-4}. Overall ASR: 78%.

8.4 Black-box Attack with True Labels

In this section, we evaluated the performance of the UAP under two different scenarios, i.e., cross-environment scenario and cross-environment & cross-model scenarios. SignFi dataset was used.

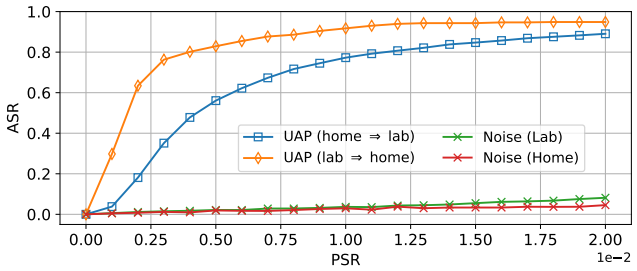


Fig. 11: Performance of UAP in the scenario of cross-environment.

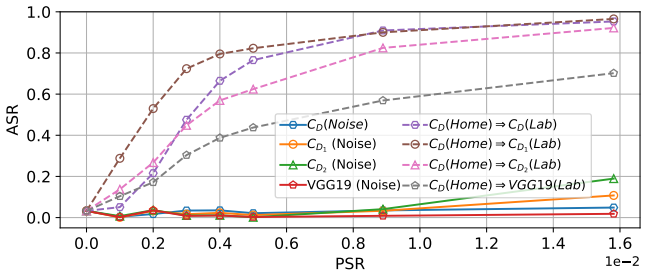


Fig. 12: Performance of UAP in the scenario of cross-environment and models.

8.4.1 *Cross-Environment Scenario*

The evaluation in this section was conducted under the assumption that the attacker is aware of the victim model's classes and the victim's deep learning architecture. However, the attacker is not aware of the model parameters f and the train/test data x.

The surrogate dataset was collected in a different environment from the victim system. In order to show the difference between the two environments, we compared a DNN's performance on 'home' data versus 'lab' data. The home-trained model achieved an accuracy of 9% on lab data as test data, while the lab-trained model achieved an accuracy of 7.8% on home-collected test data. The significant performance drop confirms the distinct nature of these two environments in terms of data characteristics.

We trained two distinct models utilising the C_D architecture shown in Fig. 7(a) on the SignFi datasets that were collected from home and lab environments, respectively. To examine UAP's cross-environment attack performance, the UAP produced in one environment will be utilised to attack the model trained in another environment. For instance, if the model of the home environment is used as a surrogate model to generate the UAP perturbation, then the model of the lab environment would be used as the victim model to test the UAP performance. The attack performance of Gaussian noise was utilised as a baseline for both models.

In Fig. 11, the blue and orange lines show the performance of UAPs that were created in the home and lab environment, respectively. They impair victim models' performance significantly. Specifically, when the PSR $= 2 \times 10^{-2}$, the UAP created from home and lab environment, achieved 89.0% and 94.8% of ASR, respectively. This indicates that surrogate models trained for the same task as the victim model make UAPs effective even with significant environmental changes. Compared to Gaussian noise attacks, UAPs compromise the victim system more effectively with low PSR. Despite variations in the CSI pattern across different environments, UAPs work across environments because the same class retains some common characteristics.

8.4.2 *Cross-Environment & Cross-Model Scenario*

In this subsection, the assumption was further released. The attacker only has the knowledge of classes on which the victim model was trained.

Both victim's and attacker's models employed four different architectures, i.e., C_D, C_{D_1}, C_{D_2}, and VGG19. The models from the home environment were utilised to generate the UAP, while the models from the lab served as victims. Fig. 12 shows the cross-environment and cross-model attack performance. The UAP performs much better than the Gaussian noise attack at the same level of PSR.

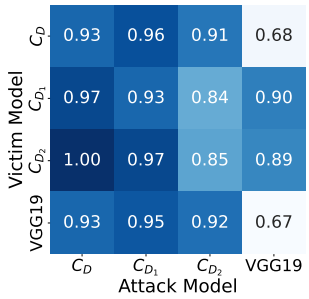


Fig. 13: Cross-environment and model test result of UAP. The attacker's environment was 'home', and the victim's environment was 'lab'. PSR = 1.58×10^{-2}.

Fig. 13 demonstrates cross-model attack capabilities using $PSR = 1.58 \times 10^{-2}$. The x-axis shows the victim model trained on the SignFi dataset in the lab environment, while the y-axis represents surrogate models trained in the home environment of the SignFi dataset. The average ASR achieved in this scenario is 89.4%. The UAP generated from model C_D to VGG19 showed the lowest ASR in the cross-environment and model scenario. The results revealed that an effective UAP can be computed even if the surrogate model does not match the victim model.

Although the training environments for the surrogate and victim models differ, their shared classes allow for learning common features. This enables the surrogate model to create a robust UAP targeting these consistent features, effectively causing misclassifications.

8.5 Black-Box Attack with Pseudo Labels

In this evaluation, Widar was utilised, involving the simultaneous recording of each sample by multiple receivers in different locations. To simulate a practical attack scenario with an eavesdropping attacker capturing Wi-Fi transmissions during gesture sensing, a different receiver was designated as the attacker. This enables the attacker to construct a CSI dataset of various gestures without corresponding labels. The k-means clustering algorithm outlined in Section 5.3 was applied to assign pseudo-labels.

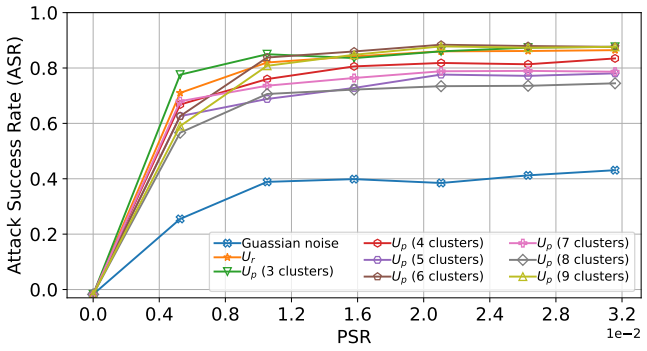


Fig. 14: Black-box attacks using pseudo-labelling.

As the attacker is unaware of the task on which the victim model was trained, we evaluated the performance of the UAP generated from surrogate models trained on varying numbers of clusters. As shown in Fig. 14, the surrogate model trained on the pseudo-label, U_p, has a similar performance to the one trained on the true label U_r. The average ASR of the UAPs produced by pseudo-label was 80%, which is much higher than the ASR of the Gaussian noise attack (40%).

Furthermore, the variability introduced by k-means clustering does not significantly affect the effectiveness of UAPs generated from DNN models. ASRs do not vary significantly across clusters from 3 to 9, and are similar to UAP generated by the surrogate model trained with real labels, U_r in Fig. 14. The performance curves overlap, indicating that UAPs maintain their ability to induce misclassifications, regardless of the number of clusters used.

Transferability [35] of adversarial examples, where those crafted for one model are effective on another with similar tasks, is essential for establishing a generalised surrogate model. In addition, the results in Section 8.4.2 also show that a UAP can be effectively transferred and applied across different models and environments, enhancing its practical applicability. This characteristic suggests that UAP creation can be considered a one-time investment. Therefore, the threat posed by this form of attack should not be underestimated, given its potential impact and feasibility.

8.6 Impact of Defence Methods on Clean Samples

During adversarial training, the victim has the ability to choose two key parameters: the PSR of each adversarial sample and the perturbation generation method. These two parameters must be carefully balanced to address the trade-off between the accuracy of the DNN model on clean samples and its robustness against various perturbations such as FGSM and PGD with a different number of iterations. It is worth noting that the DeepFool algorithm was not selected for adversarial training due to its high computational complexity. Table 1 presents the accuracy of the adversarial-trained model on clean samples, the models were trained with different PSR and perturbation generation methods. The accuracy of the adversarially robust model on clean samples varies based on both PSR and perturbation generation methods. For instance, when FGSM was used as a perturbation generation algorithm during adversarial training, the model's accuracy decreased from 93.10% to 71.7% when the PSR was increased from 1×10^{-3} to 5×10^{-3}. Additionally, a stronger attack method led to a further decrease in the model's accuracy on clean samples. Specifically, when the number of PGD iterations increased from 4 to 16, the accuracy of the model decreased from 86.70% to 16.40%. The reason for this is that an increase in the PSR or the utilisation of stronger attack methods leads to an increase in model loss, thereby impeding the convergence of the overall loss minimisation process, as represented by (20).

TABLE 1: Accuracies of adversarial trained models on clean samples

$\xi_{training}$	0.001	0.003	0.005
FGSM	93.10%	91.40%	71.70%
PGD-4	86.70%	74.20%	0.22%
PGD-8	76.70%	71.30%	0.09%
PGD-10	69.70%	74.80%	0.02%
PGD-16	16.40%	2.00%	0.90%

TABLE 2: Accuracy of smoothed model on clean samples

$\xi_{training}$	0.2	0.3	0.4	0.5	1.0	1.5
Accuracy	97.9%	98.5%	98.1%	98.8%	92.5%	85.9%

The results of randomised smoothing on clean samples are shown in Table 2. Training with Gaussian noise had a considerably smaller negative effect on model accuracy compared to adversarial training. When $\xi_{training} \leq 0.5$, the smoothed classifier retains the same performance level as the regularly trained model, maintaining an accuracy of 98.6%. This is because the Gaussian noise is inherently random and does not intentionally target the model's vulnerabilities. The Gaussian noise acts as a form of regularisation, encouraging the model to learn more robust features that could be helpful for accuracy on clean data when a proper level of noise strength is selected, while still providing some level of robustness against perturbations. When $\xi_{training} > 0.5$, the noise becomes dominant and overwhelms, which affects the classification accuracy.

8.7 Impact of Training PSR on Certified Robustness of Randomised Smoothing

To explore how training PSRs ($\xi_{training}$) influence defensive capabilities, we trained six distinct models, each augmented with noise at varying $\xi_{training}$. The certified radius was computed by setting K to 10^2, and 10^5 for initial guess and certification, respectively. Subsequently, we evaluated ACTS as we increased the PSR threshold (ξ_τ) at the testing phase. The results are shown in Fig. 15. The ACTS of all the models decrease as the increase of the ξ_τ, because the higher ξ_τ, the more strict requirements are needed. The higher ACTS at a certain ξ_τ, the more robust the model is.

As $\xi_{training}$ increases in models trained with varying PSRs, robustness initially rises, peaking at a PSR of 0.5. Notably, at $\xi_\tau = 0.5\times10^{-4}$, the model with a $\xi_{training} = 0.5$ certifies 70.1% of samples, surpassing those with $\xi_{training} = 0.2$, 0.3, 0.4, 1.0, and 1.5, which certify 27%, 40%, 57%, 48%, and 26% of samples, respectively. This is because a higher PSR during training ensures that the model does not overfit to clean data or specific types of noise. Instead, the model learns to generalise across a broader set of perturbations,

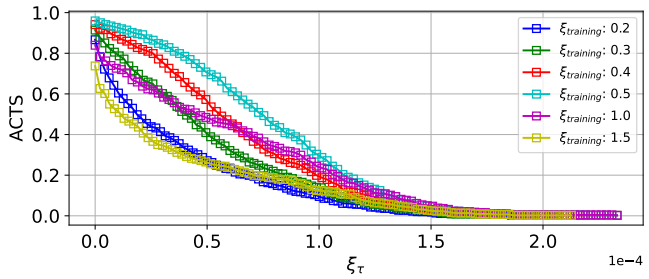


Fig. 15: The fraction of samples get certified as the increase of the PSR threshold.

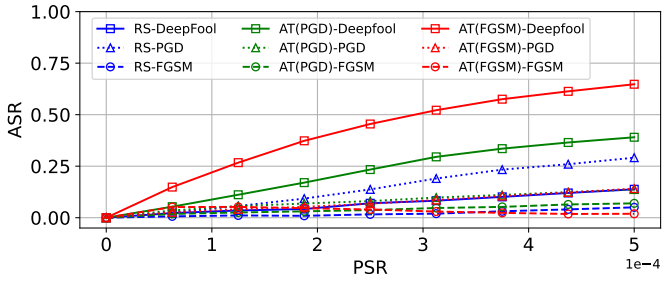


Fig. 16: Comparative defence performance of adversarial training using FGSM, PGD, and randomised smoothing techniques. The legend denoted as 'Defence method-Attack method". Randomised smoothing denoted as "RS" and adversarial training noted as "AT".

refining the decision boundaries in a manner that is less sensitive to small changes in the input.

However, a further increase of $\xi_{training}$ would reduce the robustness of the model. As the PSR continues to increase and the noise power becomes more dominant, it begins to overwhelm the genuine features of the signal during the training. At this point, while the adversarial patterns may still be disrupted, the true and meaningful patterns of the data are also masked by noise. As a result, the model may struggle to correctly classify even benign inputs, leading to a decrease in its overall robustness.

8.8 Defence Performance of Adversarial Training and Randomised Smoothing

8.8.1 Defence Against White-Box Attacks

In this subsection, we examined the defence performance of both adversarially trained models and randomised smoothed models.

For the adversarially trained model, considering the adversarial trained model's accuracy on clean samples, we only tested the models that are adversarially trained with FGSM and PGD-4 with PSR of 1×10^{-3}. For the smoothed model, we choose the model trained with a PSR of 0.5. Overall, the models trained with adversarial perturbations obtained similar defence performance compared with the smoothed classifier. However, the randomised smoothed model showed less variability in defence performance under different attacks. The results are shown in Fig. 16. Upon analysing the results, it becomes evident that the adversarially trained and smoothed models demonstrated

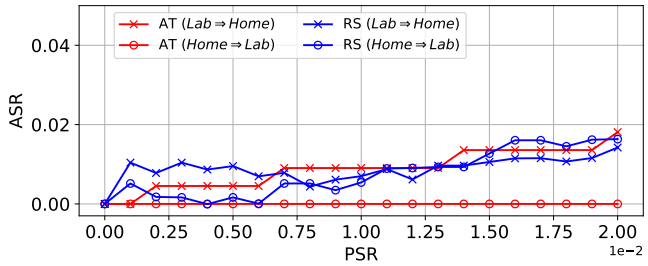


Fig. 17: Defence performance of adversarial training and randomised smoothing in the scenario of cross-environment.

enhanced robustness against evasion attacks compared to the regularly trained models (Fig. 8).

The adversarially trained models showed increased robustness against attacks. When subjected to the FGSM attack with a PSR of 5×10^{-4}, the model trained with FGSM and PGD perturbations exhibited an ASR of 6% and 8%, respectively, a stark contrast to the 63.6% ASR of the regular model. Under the more rigorous PGD attack at the same PSR, the FGSM-trained model experienced an ASR of 14.8%, while the PGD-trained model demonstrated an ASR of 13.6%. Moreover, during the DeepFool attack, the FGSM-trained model endured a high ASR of 74.9%, but the PGD-trained model had a significantly lower ASR of 45.6%. Adversarially trained models exhibit enhanced robustness against adversarial examples they encountered during training. This is because adversarially trained models have been attuned to recognise the genuine feature and motion pattern in the adversarially perturbed input. However, one limitation of adversarial training is the potential for the model to become overly specialised, developing robustness that is overly tailored to the specific types of adversarial attacks used during training. For example, the adversarial trained model demonstrated good defence capability when it faced the gradient-based attacks that it had seen during training, but failed to achieve similar defence performance when it dealt with the adversarial samples generated from the DeepFool technique.

The smoothed classifier displayed competent defensive capabilities. When challenged by the FGSM attack at a PSR of 5×10^{-4}, the smoothed model achieved an ASR of 7%. This robustness was slightly less effective against the PGD attack, where the smoothed model's ASR rose to 29.1%. Notably, at the same PSR level, the smoothed classifier recorded a more favourable ASR of 13.8% under Deepfool attacks. Unlike adversarial training specifically sharpening the model's robustness against known adversarial patterns, randomised smoothing aids in elevating the model's overall robustness, especially against unforeseen or new adversarial perturbations. As the blue lines shown in Fig. 16, they are close together, compared with the adversarial trained models, indicating that the smoothed model can maintain relatively stable defence performance.

8.8.2 Defence Against Cross-environment Black-Box Attack

This section presents an evaluation of the capacity of adversarially robust models to withstand black-box attacks in cross-environment scenarios. To do so, we conducted experiments using models that were adversarially trained with the PGD-4 (PSR = 1×10^{-3}) and ramdonised smoothing. In these experiments, we examined the defence effectiveness of different adversarially robust victim models in the scenario of cross-environment black-box attacks. Specifically, we trained surrogate models regularly and generated UAP to attack victim models that were either adversarially trained or randomised smoothed in a different environment. For example, we trained a surrogate model regularly in a home environment and used it to attack victim models that were adversarially trained in a lab environment. The experiment set-up is similar to the experiment in Section 8.4.1, except the victim models were trained with adversarial samples.

The results of these experiments are shown in Fig. 17, where adversarial training and randomised smoothing yielded comparable performance. In comparison to the regular trained model (depicted in Fig. 11), both adversarially trained models and randomised smoothed models in both scenarios demonstrate a substantial ability to decrease the impact of UAPs, with the UAP generated from the lab environment exhibiting the strongest ASR of below 2%.

During adversarial training, the model is exposed to both clean and adversarial examples, allowing it to learn the intrinsic features of the input data that are relevant to the classification task during adversarial training, rather than relying on irrelevant or spurious features that are susceptible to adversarial perturbations. Therefore, the adversarially trained model is less sensitive to small perturbations around the input data, making it more difficult for attackers to generate small adversarial perturbations.

As for randomised smoothed models, their inherent design introduces a layer of variability during training. By incorporating Gaussian noise, these models develop a more generalised understanding of the decision boundary, making them less sensitive to perturbations.

It is worth noting that the focus of our evaluation is exclusively on this particular scenario, i.e., cross-environment black-box scenario, given that it provides the attacker with the most significant advantage, compared to the other two scenarios, i.e., cross both environment and model attack and pseudo-label attack. An additional constraint was put on the attack's abilities. Therefore, if the adversarially robust model were robust in this scenario, it would have a greater likelihood of performing well in more restricted scenarios. This would indicate that the model has a high level of robustness to a variety of black-box attack circumstances. Therefore, the evaluation of the model's performance in this scenario is critical in assessing its overall robustness and effectiveness.

8.8.3 Complexity Analysis

In the evaluation of defence mechanisms against evasion attacks, the complexity of each training epoch varied significantly across different methods. Adversarial training using the FGSM clocked in at 34.58 seconds per epoch. A more intensive approach, adversarial training with PGD-4, required a higher time investment of 46.58 seconds per epoch. In contrast, randomised smoothing, which entails training with Gaussian noise, stood out for its efficiency, costing only 4 seconds per epoch.

Regarding the convergence speed of these methods, adversarial training with FGSM and PGD-4 showed similar requirements, taking 323 and 322 epochs, respectively, to converge. Randomised smoothing again demonstrated its efficiency, requiring only 128 epochs to achieve convergence.

Summarising the overall training time, randomised smoothing emerges as the more time-efficient defence mechanism. In terms of performance, randomised smoothing achieved comparable defence performance with adversarial training against gradient-based attacks, but the defence capability remains valid over unseen attacks.

9 Related Works

9.1 Evasion Attacks in Wireless Domain

The evasion attack has been studied for wireless communications such as spectrum sensing [26], [36], modulation classification [37]–[39], power allocation [40] and 5G networks [41]. For example, the work in [41] presented a "myopic threat model" to simulate realistic adversarial machine learning (ML) attacks in 5G networks. This threat model was proactively tested on six ML applications within 5G, five out of six applications were broken by the proposed attacks. Compared to conventional jamming attacks [42], evasion attacks are often more stealthy and energy-efficient.

Evasion attacks against deep learning-based Wi-Fi sensing is still in the nascent stage. An initial exploration was presented in [43], which considers white-box approaches and assumes the attacker can access the model's parameters. In contrast, [17] proposed a model-independent attack, but it lacked consideration of environmental diversity, potentially weakening the attack's effectiveness. The study in [27] presented a black-box attack method using random Gaussian noise, but this was not optimised for deep learning models, leading to inefficiency. The authors in [44], [45] demonstrated vulnerabilities in Wi-Fi-based behaviour and gesture recognition systems by developing attacks that target the inference stage - using signal jamming and adversarial perturbations, respectively. However, These methods often require large jamming signals that can interrupt data transmissions. Moreover, [45] relied on frequent victim model queries for specific perturbations, which may not be practical due to limited access to victim model responses. The latest work [18] studied adversarial attacks in Wi-Fi sensing using PGD and FGSM, which focused on the impact of the attack on joint communication and attack performance.

9.2 Countermeasures

In order to address the threat of evasion attacks, defence mechanisms have been designed, such as adversarial training and randomised smoothing. Initially proposed by Goodfellow et al. in [20], adversarial training trains neural networks on adversarial examples. This method's importance is underscored by [46], which highlights the need for adaptable defences against simple but effective evasion attacks.

In cybersecurity, as [47] notes, such training is crucial for countering threats to machine-learning classifiers. Additionally, [48] emphasises the need for this training method to address learning algorithm vulnerabilities against sophisticated attacks. Extending this concept, the authors in [40] applied adversarial training to power allocation in massive MIMO systems in wireless communication, demonstrating improved adversarial robustness against gradient-based attacks. Similarly, in the field of wearable device-based human activity sensing, the authors in [49] studied adversarial training against white-box attacks.

In the area of Wi-Fi sensing, the authors in [19] investigated the vulnerability of the Wi-Fi sensing model, leveraged FGSM and PGD, and proposed an adversarial training method as countermeasures to improve the robustness of the sensing model. Nevertheless, the study exclusively examined adversarial attack types encountered during the training phase, thereby failing to ensure robustness against unknown attacks.

Different from adversarial training, randomised smoothing can not only address the challenge of unknown evasion attacks but also bring advantages like scalability and certified robustness [24], [50]. Randomised smoothing introduces controlled noise and creates models less sensitive to small perturbations, which is more resilient to a variety of attacks. Its scalability ensures effectiveness across diverse model sizes and types, while the aspect of certified robustness provides a quantifiable measure of security against adversarial threats. However, adopting randomised smoothing for WiFi sensing is missing and their performance is unknown.

10 Conclusion and Future Work

This paper not only extensively studied evasion attacks against deep learning-based Wi-Fi sensing systems, revealing their vulnerabilities to even minimal power changes, but also introduced effective defence methods to enhance system robustness and security. Our experiments demonstrate the significant vulnerability of WiFi sensing model to evasion attacks which achieved ASR as high as 97.0% in white-box scenarios and 95.6% in black-box scenarios. Notably, our pseudo-labelling strategy, which can be launched easily by eavesdropping the sensing signal of victim Wi-Fi sensing system, achieved an average ASR of 80%. To address these vulnerabilities, we implemented adversarial training and randomised smoothing as defences. These strategies considerably improved the robustness of the Wi-Fi sensing model by reducing ASR to about 6% for white-box and 2% for black-box scenarios. Additionally, randomised smoothing provided certifiable robustness, with 70.1% of samples certified in our most robust model, enhancing predictability and security in Wi-Fi sensing systems.

In a real-world Wi-Fi sensing system, the scope extends beyond the sensing DNN model to encompass various preprocessing steps, including noise filtering, signal normalisation, and feature extraction. These steps are crucial to the system's functionality, as they can significantly modify raw Wi-Fi signals before their analysis by the deep learning model. The extent to which these preprocessing steps impact the effectiveness of adversarial examples remains an open question. Furthermore, when adversarial perturbations are transmitted over the air, factors like multipath fading and synchronisation between the perturbation and the sensing signal introduce additional complexities. Therefore, our future work aims to broaden our research scope to not only focus on the DNN model but also consider the entire Wi-Fi sensing system. This expansion will involve examining various preprocessing schemes in over-the-air scenarios, providing a more comprehensive understanding of the system's vulnerabilities and robustness.

References

[1] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained Wi-Fi signatures," in *Proc. Annual Int. Conf. Mobile Computing and Networking (MobiCom)*, Maui, Hawaii, Sep. 2014, pp. 617–628.

[2] L. Zhang, Z. Wang, and L. Yang, "Commercial Wi-Fi based fall detection with environment influence mitigation," in *Proc. Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, Massachusetts, USA, Jun. 2019, pp. 1–9.

[3] Z. Wu, X. Xiao, C. Lin, S. Gong, and L. Fang, "Widff-id: Device-free fast person identification using commodity wifi," *IEEE Trans. on Cogn. Commun. Netw.*, Nov. 2022.

[4] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using Wi-Fi signals," in *Proc. ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, New York, NY, USA, Sep. 2016, p. 363–373.

[5] Y. Zhang, Y. Zheng, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Widar3.0: Zero-effort cross-domain gesture recognition with Wi-Fi," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, Aug. 2021.

[6] G. Yin, J. Zhang, G. Shen, and Y. Chen, "FewSense, towards a scalable and cross-domain Wi-Fi sensing system using few-shot learning," *IEEE Trans. Mobile Comput.*, 2022.

[7] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "SignFi: Sign language recognition using Wi-Fi," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–21, Mar. 2018.

[8] Y. Ma, G. Zhou, and S. Wang, "Wi-Fi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–36, 2019.

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

[10] B. Tan, Q. Chen, K. Chetty, K. Woodbridge, W. Li, and R. Piechocki, "Exploiting WiFi channel state information for residential healthcare informatics," *IEEE Communi. Mag.*, vol. 56, no. 5, pp. 130–137, 2018.

[11] D. Wang, J. Yang, W. Cui, L. Xie, and S. Sun, "Airfi: empowering wifi-based passive human gesture recognition to unseen environment via domain generalization," *IEEE Trans. Mob. Comput.*, Dec. 2022.

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

[13] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial machine learning in wireless communications using RF data: A review," *IEEE Communi. Surv.s & Tutor.*, 2022.

[14] H. Zhang and J. Wang, "Towards adversarially robust object detection," in *Proc. Inter. Conf. Comput. Vis.*, Seoul, Korea, Oct. 2019, pp. 421–430.

[15] A. Arnab, O. Miksik, and P. H. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 888–897.

[16] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial machine learning attacks and defense methods in the cyber security domain," *ACM Comput. Sur. (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021.

[17] L. Xu, X. Zheng, X. Li, Y. Zhang, L. Liu, and H. Ma, "Wicam: Imperceptible adversarial attack on deep learning based wifi sensing," in *Proc. Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, Stockholm, Sweden, Oct. 2022, pp. 10–18.

[18] L. Xu, X. Zheng, Y. Zhang, L. Liu, and H. Ma, "WiCAM 2.0: Imperceptible and targeted attack on deep learning based WiFi sensing," *ACM Trans. on Sens. Netw.*, vol. 20, no. 6, pp. 1–22, 2024.

[19] Y. Pan, Z. Zhou, W. Gong, and Y. Fang, "SAT: A selective adversarial training approach for wifi-based human activity recognition," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 12 706–12 716, 2024.

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

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[22] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, US, Jun. 2016, pp. 2574–2582.

[23] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[24] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. internat. conf. on mach. learn.*, California, USA, 2019, pp. 1310–1320.

[25] Z. Wang, B. Guo, Z. Yu, and X. Zhou, "Wi-fi csi-based behavior recognition: From signals and actions to activities," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 109–115, 2018.

[26] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Trans. on Infor. For. and Secur.*, vol. 15, pp. 1102–1113, Aug. 2019.

[27] J. Yang, H. Zou, and L. Xie, "SecureSense: Defending adversarial attack for secure device-free human activity recognition," *IEEE Trans. Mobile Comput.*, pp. 1–11, 2022.

[28] L. Wu, Z. Zhu, C. Tai *et al.*, "Understanding and enhancing the transferability of adversarial examples," *arXiv preprint arXiv:1802.09707*, 2018.

[29] W. Jiang, Z. He, J. Zhan, W. Pan, and D. Adhikari, "Research progress and challenges on application-driven adversarial examples: a survey," *ACM Trans. on Cyb. Phys. Sys. (TCPS)*, vol. 5, no. 4, pp. 1–25, 2021.

[30] A. Goel, A. Singh, A. Agarwal, M. Vatsa, and R. Singh, "Smartbox: Benchmarking adversarial detection and mitigation algorithms for face recognition," in *Proc. Internat. Conf. Bio. Theo., Appli. and Sys. (BTAS)*, Redondo Beach, CA, USA, Oct. 2018, pp. 1–7.

[31] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Eur. Conf. Comput. Vis. (ECCV)*, Munich Germany, Sep. 2018, pp. 132–149.

[32] D. Arthur and S. Vassilvitskii, "K-means++ the advantages of careful seeding," in *Proceedings of the eighteenth ann. ACM-SIAM sym. on Dis. algori.*, 2007, pp. 1027–1035.

[33] A. I. Károly, R. Fullér, and P. Galambos, "Unsupervised clustering for deep learning: A tutorial survey," *Acta Polytechnica Hungarica*, vol. 15, no. 8, pp. 29–53, 2018.

[34] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM. SIGCOMM. CCR.*, vol. 41, no. 1, p. 53, Jan. 2011.

[35] A. G. Matachana, K. T. Co, L. Muñoz-González, D. Martinez, and E. C. Lupu, "Robustness and transferability of universal attacks on compressed models," *arXiv preprint arXiv:2012.06024*, 2020.

[36] Y. E. Sagduyu, Y. Shi, and T. Erpek, "Adversarial deep learning for over-the-air spectrum poisoning attacks," *IEEE Trans. Mob. Comput.*, vol. 20, no. 2, pp. 306–319, Oct. 2021.

[37] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 213–216, Aug. 2018.

[38] K. W. McClintick, J. Harer, B. Flowers, W. C. Headley, and A. M. Wyglinski, "Countering physical eavesdropper evasion with adversarial training," *IEEE open j. Commun. Soc.*, vol. 3, pp. 1820–1833, Oct. 2022.

[39] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels," in *Proc. Annu. Conf. Inf. Sci. and Sys. (CISS)*, Princeton, NJ , USA, Jan. 2020, pp. 1–6.

[40] B. Manoj, M. Sadeghi, and E. G. Larsson, "Downlink power allocation in massive mimo via deep learning: Adversarial attacks and training," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 707–719, 2022.

[41] G. Apruzzese, R. Vladimirov, A. Tastemirova, and P. Laskov, "Wild networks: Exposure of 5G network infrastructures to adversarial examples," *IEEE Trans. Netw. Serv. Manag*, vol. 19, pp. 5312–5332, Dec. 2022.

[42] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, no. 1, pp. 2–14, 2018.

[43] H. Ambalkar, X. Wang, and S. Mao, "Adversarial human activity recognition using Wi-Fi csi," in *Can. J. Electr. Comput. Eng.*, May 2021, pp. 1–5.

[44] J. Liu, Y. He, C. Xiao, J. Han, L. Cheng, and K. Ren, "Physical-world attack towards Wi-Fi-based behavior recognition," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Jun. 2022, pp. 400–409.

[45] Y. Zhou, H. Chen, C. Huang, and Q. Zhang, "WiAdv: Practical and robust adversarial attack against Wi-Fi-based gesture recognition system," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 2, pp. 1–25, 2022.

[46] G. Apruzzese, H. S. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. Roundy, ""real attackers don't compute gradients": Bridging the gap between adversarial ml research and practice," in *IEEE Conf. on Sec. and Trus.y Mach. Learn. (SaTML)*, 2023, pp. 339–364.

[47] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, "Addressing adversarial attacks against security systems based on machine learning," in *Inter. conf. on cyb. confl. (CyCon)*, vol. 900, May 2019, pp. 1–18.

[48] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," in *Proc. ACM SIGSAC Conf. on Comput. and Comm. Sec.*, New York, US, Oct. 2018, pp. 2154–2156.

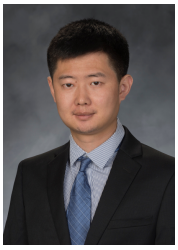
[49] R. K. Sah and H. Ghasemzadeh, "Adar: Adversarial activity recognition in wearables," in *IEEE/ACM Int. Conf. Comput.-Aided Des. Dig. Tech. Pap. (ICCAD)*, Westminster, Colorado, USA, Dec. 2019, pp. 1–8.

[50] L. Li, T. Xie, and B. Li, "Sok: Certified robustness for deep neural networks," in *IEEE sympos. on secur. and priva. (SP)*, May 2023, pp. 1289–1310.

Guolin Yin received a Ph.D degree in Electronics and Electrical Engineering from the University of Liverpool, UK in 2024. He is currently a research associate at the University of Liverpool. Before that, he received the B.Eng in Electrical and Electronic Engineering from Coventry University, UK, in 2018 and M.Eng degrees from the University of Manchester, UK, in 2019. His research interests include wireless sensing, radio frequency fingerprint identification, and deep learning.

Junqing Zhang received a Ph.D. degree in Electronics and Electrical Engineering from Queen's University Belfast, UK in 2016. From Feb. 2016 to Jan. 2018, he was a Postdoctoral Research Fellow at Queen's University Belfast. From Feb. 2018 to Oct. 2022, he was a Tenure Track Fellow and then a Lecturer (Assistant Professor) at the University of Liverpool, UK. Since Oct. 2022, he has been a Senior Lecturer (Associate Professor) at the University of Liverpool. His research interests include the Internet of Things, wireless security, physical layer security, key generation, radio frequency fingerprint identification, and wireless sensing. He was a co-recipient of the IEEE WCNC 2025 Best Workshop Paper Award. He is a Senior Area Editor of IEEE Transactions on Information Forensics and Security and an Associate Editor of IEEE Transactions on Mobile Computing.

Xinping Yi (S'12-M'15) received his Ph.D. degree in Electronics and Communications in 2015 from Télécom ParisTech, Paris, France. He is currently a Professor at the National Mobile Communications Research Laboratory, Southeast University, China. Prior to that, he has been an Assistant Professor at University of Liverpool, United Kingdom, a Postdoctoral Research Associate at Technische Universität Berlin, Germany, and a Research Assistant at EURECOM, France. His main research interests include network information theory, statistical learning theory, graph machine learning, and their applications in wireless communications and trustworthy artificial intelligence.



Xuyu Wang [S'13-M'18] received the M.S. in Signal and Information Processing in 2012 and B.S. in Electronic Information Engineering in 2009, both from Xidian University, Xi'an, China. He received a Ph.D. in Electrical and Computer Engineering from Auburn University, Auburn, AL, USA in Aug. 2018. He is currently an Assistant Professor in the Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL, USA. His research interests include wireless sensing, Internet of Things, wireless localization, smart health, wireless networks, and deep learning. He received the NSF CRII Award in 2021. He was a co-recipient of the ACM FAcct 2023 Best Paper Award, the 2022 Best Journal Paper Award of IEEE ComSoc eHealth Technical Committee, the IEEE INFOCOM 2022 Best Demo Award, the IEEE ICC 2022 Best Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the IEEE GLOBECOM 2019 Best Paper Award, the IEEE ComSoc MMTC Best Journal Paper Award in 2018, the IEEE PIMRC 2017 Best Student Paper Award, and the IEEE SECON 2017 Best Demo Award. He is an associate editor for the IEEE Transactions on Mobile Computing.