Noise, Why Can't You Bend? Detecting Adversarial Perturbations in Wireless Sensing via Structural Fragility

Md Hasan Shahriar
Virginia Tech
Blacksburg, Virginia, USA
hshahriar@vt.edu

Ning Wang
University of South Florida
Tampa, Florida, USA
ningw@usf.edu

Amit Kumar Sikder
Iowa State University
Ames, Iowa, USA
aksikder@iastate.edu

Naren Ramakrishnan
Virginia Tech
Blacksburg, Virginia, USA
naren@vt.edu

Y. Thomas Hou
Virginia Tech
Blacksburg, Virginia, USA
thou@vt.edu

Wenjing Lou
Virginia Tech
Blacksburg, Virginia, USA
wjlou@vt.edu

Abstract

Wireless sensing enables safety- and privacy-critical applications, such as activity recognition, user authentication, and vital-sign monitoring, by interpreting WiFi's channel state information (CSI) with deep learning (DL) models. However, this reliance on DL exposes wireless sensing systems to adversarial perturbations: subtle, human-imperceptible modifications to CSI that can mislead models without affecting wireless communication. Unlike natural noise, which is inherently flexible, adversarial noise is rigid and fragile—the effectiveness diminishes when its carefully crafted structure is distorted, revealing its malicious intent. With that observation, we present NoiFi, a lightweight, online, and practical defense that detects adversarial CSI manipulations by exploiting this structural fragility. NoiFi samples the inherent noise from each test input and generates randomized noise variants, which are then mapped through the target classifier to construct a randomized noise manifold. For benign noise, its randomized manifold forms around its feature vector, whereas for adversarial noise, due to its fragility, the manifold deviates far away, revealing the rigid and malicious structure of the perturbations. By targeting such fundamental discrepancy between bendable natural noise and fragile adversarial perturbations, NoiFi provides an attack-agnostic, interpretable, and near real-time defense for WiFi sensing systems. Evaluations on multiple WiFi sensing datasets demonstrate high detection performance (AUROC/AUPRC of 0.93–1.00) and strong generalization—across datasets, noise distribution, and model architectures—both in white-box, black-box, and adaptive attack settings. The code and artifacts are available at: https://github.com/shahriar0651/NoiFi.

CCS Concepts

• **Security and privacy → Mobile and wireless security**; • **Computing methodologies → Neural networks**.

Keywords

Wireless Sensing, Machine Learning, Adversarial Attacks, Detection

ASIA CCS '26, June 1–5, 2026, Bangalore, India

ACM ISBN 979-8-4007-2356-8/26/06
https://doi.org/10.1145/3779208.3806083

ACM Reference Format:
Md Hasan Shahriar, Ning Wang, Amit Kumar Sikder, Naren Ramakrishnan, Y. Thomas Hou, and Wenjing Lou. 2026. Noise, Why Can't You Bend? Detecting Adversarial Perturbations in Wireless Sensing via Structural Fragility. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '26), June 1–5, 2026, Bangalore, India.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3779208.3806083

1 Introduction

Wireless sensing leverages variations in wireless signals, such as WiFi, to detect motion, gestures, and even certain biometric traits [32]. It is effectively turning networking devices into motion sensors without extra hardware, enabling applications in home security, healthcare, and smart buildings [38, 44, 50]. Modern WiFi devices extract channel state information (CSI)—fine-grained measurements of multipath channel gains—that encode how signals interact with objects while propagating. The integration of deep learning (DL) has further enabled high-accuracy wireless sensing, fueling growing interest in this non-intrusive sensing modality [40–42, 50]. However, while DL models have become the workhorse of WiFi sensing, enabling a wide range of applications, their integration also introduces critical security vulnerabilities. In particular, *adversarial attacks*—carefully crafted perturbations that are imperceptible to humans but induce misclassification—have emerged as a serious and insidious threat against DL-based applications [10, 15, 24, 26, 28, 35].

In the context of WiFi sensing, such attacks translate to minute, targeted manipulations of the CSI estimation using a variety of techniques, such as modifying pilot tones [17] or injecting carefully timed jamming signals [19], to mislead a broad range of DL–based WiFi sensing systems [17, 45, 49, 55]. For instance, as illustrated in Fig. 1, an attacker could manipulate input signals so that an arbitrary gesture is falsely recognized as an unlock-the-door command, enabling unauthorized access. As the potential impact of such attacks is profound, developing robust defense mechanisms is of paramount importance to ensuring the integrity and trustworthiness of WiFi sensing systems amid growing adversarial threats.

Defenses against adversarial attacks broadly fall into two categories. The first type aims to *increase model robustness* by modifying training procedures or input representations—common approaches include adversarial training [10, 24], and input filtering or denoising signal [49]. While these techniques can enhance resilience to attacks, they often come at the cost of reduced benign accuracy,

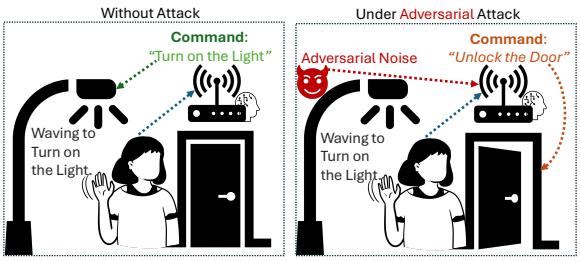


Figure 1: Illustration of adversarial attacks on WiFi sensing. Without attack (left), the system correctly interprets the user's gesture as the command *"Turn on the Light"*. Under adversarial perturbation (right), malicious noise manipulates the sensed signal, causing the system to misinterpret the same gesture as *"Open the Door"*.

increased training time, and poor generalization to unseen attack types. In contrast, the second type of defense focuses on *detecting adversarial inputs* at test time [4, 21, 47], which are compelling as they can act as a runtime safety net without requiring model retraining and generalize better across different attack strategies.

Adversarial input detection techniques generally fall into two further subcategories. The first analyzes the *test input* as a whole and analyzes the input representation or activations to identify inconsistencies or statistical anomalies—approaches include feature-space monitoring [46], manifold consistency checks [7, 39], reconstruction discrepancy [25], etc. However, these methods often rely on implicit assumptions about attack patterns or perturbation magnitudes and can fail under stealthy attacks [33]. A second, and a more effective strategy, is to sample the *non-semantic noise*, possibly resulting from an attack-induced perturbation, and examine hidden properties to reveal adversarial footprints [25]. Although existing noise-based detectors—particularly, NoiSec [33]—propose effective methods for profiling adversarial noise, they rely on strong assumptions about benign noise distributions (e.g., Gaussian), which limit their robustness in dynamic, real-time systems like wireless sensing. These limitations reveal the critical need for more robust noise-based defense *that is not built upon any specific noise distribution but still provides generalized adversarial detection* in practical and dynamic WiFi sensing systems.

To design such a generalized detector, we draw on the observation that adversarial perturbations are inherently aligned with the gradients of the target model, whereas benign noise is not. This fundamental difference causes the two to manifest distinct feature responses within the model. Thus, if we randomize the adversarial noise to break the structure—hence the alignment—it creates a totally different feature response (and behaves like a benign noise). On the other hand, randomizing a benign noise does not create any significant feature deviation, as randomization creates just another benign noise, which is indifferent to the target model. Based on these two distinct characteristics, we propose NoiFi, a noise-based detection framework that does not rely on a static pre-trained anomaly detection model trained on Gaussian noise, but rather takes a *dynamic, model-free, on-the-fly* approach.

Here are the key contributions of this work:

- We introduce NoiFi, a dynamic online detection framework that models input-conditioned perturbations in the classifier's feature space. By generating representative noise distributions on the fly, NoiFi distinguishes natural (on-manifold) from adversarial (off-manifold) perturbations, enabling interpretable and robust detection while mitigating domain shift issues common in other state-of-the-art methods that use static pre-trained outlier detection methods.
- We comprehensively evaluate NoiFi on two WiFi sensing datasets across diverse adversarial attacks, noise samplers, and outlier detectors. NoiFi achieves AUROC and AUPRC scores of 0.96–1.00 with near-zero false positives and 10 ms inference latency, demonstrating both high effectiveness and suitability for real-time deployment. It remains robust in both white-box and black-box scenarios, outperforming existing baselines with average recall improvements of 97% and 147%, respectively.
- Beyond adversarial robustness, NoiFi maintains consistent performance under diverse benign noise distributions—including statistically isotropic (Gaussian), sparse (impulse), and localized (burst). Furthermore, it generalizes across modalities, achieving strong performance on medical imaging and speech datasets, confirming its generalization and domain-agnostic applicability.

2 Background and Threat Model

This section presents the background of DL-based wireless sensing that uses CSI. Further, it presents the threat model, defense assumption, and a motivating example on which NoiFi is built.

2.1 Channel State Information in WiFi Sensing

CSI characterizes the frequency-domain behavior of a wireless channel by capturing how transmitted signals are attenuated and phase-shifted due to multipath propagation [22]. In WiFi systems employing OFDM and MIMO technologies, such as those defined by the IEEE 802.11n/ac standards [11], CSI provides a per-subcarrier estimation of the complex channel gain between each transmit–receive antenna pair. Let $s_{k,p} \in \mathbb{C}$ be the transmitted symbol on the subcarrier k during a packet p, and let $r_{k,p} \in \mathbb{C}$ denote the corresponding received symbol. Under a narrowband flat-fading assumption on each subcarrier, the received signal is modeled as:

$$r_{k,p} = h_{k,p} s_{k,p} + n_{k,p}, \tag{1}$$

where $h_{k,p} \in \mathbb{C}$ is the complex channel gain (i.e., the CSI), and $n_{k,p} \sim \mathcal{CN}(0, \sigma^2)$ is additive white Gaussian noise. When pilot symbols $s_{k,p}$ are known to the receiver, the channel can be estimated using a Least Squares (LS) approach in vectorized form. Let $\mathbf{r}_p \in \mathbb{C}^{K\times 1}$ be received pilot symbols during packet p, corresponding to the known transmitted pilot vector $\mathbf{s}_p \in \mathbb{C}^{K\times 1}$. The input-output relationship across all K subcarriers can be written as:

$$\mathbf{r}_p = \mathbf{s}_p \mathbf{h}_p + \mathbf{n}_p, \tag{2}$$

where $\mathbf{s}_p = \mathrm{diag}(\mathbf{s}_p) \in \mathbb{C}^{K\times K}$ is a diagonal matrix of transmitted pilot symbols, $\mathbf{h}_p \in \mathbb{C}^{K\times 1}$ is the subcarrier-wise channel vector to be estimated, and $\mathbf{n}_p$ is the noise vector. The LS estimate of $\mathbf{h}_p$ is given by:

$$\hat{\mathbf{h}}_p = (\mathbf{s}_p^H \mathbf{s}_p)^{-1} \mathbf{s}_p^H \mathbf{r}_p. \tag{3}$$

For each packet p, the estimated channel vector $\hat{\mathbf{h}}_p \in \mathbb{C}^{K\times 1}$ contains the subcarrier-wise complex channel gains. By aggregating

estimates over P packets, the full CSI matrix is formed as:

$$\hat{\mathbf{H}} = \begin{bmatrix} \hat{h}_{1,1} & \hat{h}_{1,2} & \cdots & \hat{h}_{1,P} \\ \hat{h}_{2,1} & \hat{h}_{2,2} & \cdots & \hat{h}_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{h}_{K,1} & \hat{h}_{K,2} & \cdots & \hat{h}_{K,P} \end{bmatrix} \in \mathbb{C}^{K\times P}, \tag{4}$$

where each column corresponds to the channel frequency response for a given packet, and each row captures the evolution of a particular subcarrier over time. The estimated CSI serves as a fundamental input to DL applications such as activity recognition, gesture classification, and device-free localization.

2.2 CSI Classification Using Deep Learning

Once the CSI matrix $\hat{\mathbf{H}} \in \mathbb{C}^{K\times P}$ is estimated across K subcarriers and P packets, it serves as a spatiotemporal representation of the wireless channel. Changes in this matrix, caused by human motion or environmental dynamics, can be exploited for classification tasks such as activity recognition or gesture detection. To apply DL, the complex CSI matrix is typically converted into a real-valued representation, such as taking the magnitude matrix $x = |\hat{\mathbf{H}}| \in \mathbb{R}^{K\times P}$, which is suitable for DL input.

Let $\mathcal{M}_\theta : \mathbb{R}^{K\times P} \rightarrow \mathbb{R}^{|\mathcal{Y}|}$, where $\mathcal{M}_\theta$ is a DL classifier parameterized by θ. The input x represents a sample with K channels and P features, and the output lies in $\mathbb{R}^{|\mathcal{Y}|}$, corresponding to the prediction scores over the label set $\mathcal{Y}$ (e.g., activities or gestures). During training, the parameters θ are optimized to minimize a loss function over a labeled dataset of CSI samples. Once trained, the model can perform real-time classification by directly processing new CSI matrices as they are captured from the wireless channel.

2.3 Attacker's Objective and Capabilities

This part presents the threat model from the perspective of the attacker's objective and capabilities.

Attacker's Objective. The adversary's primary objective is to undermine the reliability of WiFi sensing applications that rely on DL models for interpreting wireless signals, particularly CSI. Instead of broadly disrupting the wireless channel, the attacker focuses on injecting subtle, targeted perturbations that interfere with the model's inference process. Despite the success of DL models in classifying CSI matrices, recent research has demonstrated their vulnerability to adversarial attacks. In such attacks, an adversary subtly modifies the input CSI data to mislead the classifier $\mathcal{M}_\theta$, while keeping the perturbation imperceptible or physically plausible. These perturbations can cause the model to output incorrect labels, thereby undermining the reliability of CSI-based sensing systems. Hence, we assume that the adversarial attack on the original input x_{org} aims to generate a perturbed input x_{adv} that satisfies the following properties:

$$\begin{aligned} & x_{\text{adv}} = x_{\text{org}} + \delta_{\text{adv}}, \\ \text{s.t.} \quad & \|\delta_{\text{adv}}\|_p \le \epsilon, \quad \arg\max \mathcal{M}_\theta(x_{\text{adv}}) \neq \arg\max \mathcal{M}_\theta(x_{\text{org}}) \end{aligned} \tag{5}$$

where $\delta_{\text{adv}} \in \mathbb{R}^{K\times P}$ is the adversarial perturbation bounded in ℓ_p norm by a small constant $\epsilon > 0$.

The goal of the adversary is to ensure the prediction changes while keeping the perturbation small enough to avoid detection. As (5) is hard to solve, there are different methods for the approximation of such perturbations δ_{adv}. A common method is the fast gradient sign method (FGSM) [10], which computes the perturbation as:

$$\delta_{\text{adv}} = \epsilon \, \text{sign}\Big(\nabla_{x_{\text{org}}} \mathcal{L}(\theta, x_{\text{org}}, y_{\text{org}})\Big), \tag{6}$$

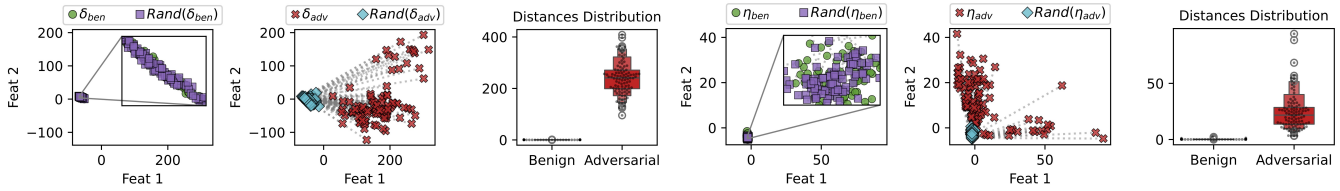
where $\mathcal{L}$ is the loss function (e.g., cross-entropy), and y_{org} is the true class label. More sophisticated attacks, such as projected gradient descent (PGD) [24], basic iterative method (BIM) [15], etc., iteratively apply small FGSM-like updates and project the result back into the ϵ-bounded ball. In contrast, universal adversarial perturbation (UAP) [26] exhibits greater potency, as it is generated offline by optimizing a perturbation over a small, representative dataset. This process yields an input-agnostic perturbation that can be applied to arbitrary test inputs to consistently mislead the model. While the end goal is to generate an effective δ_{adv}, each of these methods solves (5) in different ways, which are elaborated in Appendix A.

Attacker's Capabilities. We assume an attacker can employ gradient-based adversarial methods (e.g., [10, 15, 24, 26]), which query the model for a small number of iterations to generate the perturbation. Regardless of the method, the attacker's ultimate objective is to manipulate the CSI such that the perturbed input misleads the target classifier $\mathcal{M}_{\text{target}}$ (we omit the parameters θ for brevity, without loss of generality). We adopt two standard threat models regarding model access. **White-box Attack:** the attacker has full access to the target model $\mathcal{M}_{\text{target}}$, including its architecture and parameters. This enables precise crafting of adversarial perturbations tailored to the model's weights and gradients, often resulting in high attack success rates. **Black-box Attack:** where the attacker does not have direct access to $\mathcal{M}_{\text{target}}$ and instead relies on a surrogate model $\mathcal{M}_{\text{surrogate}}$ that approximates the behavior/architecture of the target. Adversarial examples are generated on the surrogate and transferred to the target, typically with reduced effectiveness compared to white-box attacks.

After generating the adversarial perturbation, the attacker can deploy it in either the digital or physical domain. In the digital domain, the adversary injects perturbations directly into the CSI matrix after it has been extracted at the receiver, assuming access to the CSI data stream through software-level compromises, malicious firmware, or unauthorized access to stored sensing data. In the physical domain, the attacker instead alters the wireless environment or transmission characteristics to influence the CSI as it is estimated by the receiver, without requiring access to the internal CSI pipeline, by perturbing the physical propagation path or signal behavior. Either way, the attacker's end goal remains the same: to manipulate the CSI estimation x to x_{adv} by injecting adversarial perturbations δ_{adv} that cause $\mathcal{M}_{\text{target}}$ to make incorrect predictions.

2.4 Defense Objective and Assumptions

The central challenge addressed in this work is the robust detection of adversarial perturbations embedded in CSI data. Unlike traditional anomaly detection methods that approach this as an adversarial input classification task, we formulate the problem as a *noise sampling and analysis task*: given an observed CSI input, the goal is to separate the true channel representation from additive noise and determine whether the noise is benign or adversarial.



(a) Feature representation of original noise: benign and adversarial.

(b) Feature representation of reconstructed noise: benign and adversarial.

Figure 2: Feature representations of benign and adversarial noise, along with their randomized variants at two stages: (a) original and (b) reconstructed. In (a), benign noise (δ_{ben}) and its randomized versions ($Rand(\delta_{\text{ben}})$) cluster tightly, indicating high representational consistency. Adversarial noise (δ_{adv}), however, appears more dispersed, while $Rand(\delta_{\text{adv}})$ also forms tighter, shifted clusters. Here, the dotted lines highlight the distance traversed in feature space after randomization. The distribution shows that these distances are clearly separable and can be an indication of malicious intent. In (b), reconstructed noise (η_{ben} and η_{adv}) exhibits similar trends, suggesting that reconstruction-based noise sampling preserves structural distinctions of both benign and adversarial noises.

We model each test CSI input $x_{\text{test}} \in \mathbb{R}^{K\times P}$ as a superposition of the underlying true CSI signal $x_{\text{org}} \in \mathbb{R}^{K\times P}$ and an additive noise term $\delta_{\text{test}} \in \mathbb{R}^{K\times P}$:

$$x_{\text{test}} = x_{\text{org}} + \delta_{\text{test}}. \tag{7}$$

The nature of the noise δ_{test} governs the classification of x_{test} as either benign or adversarial. We formalize this distinction as follows.

Definition 1 (Benign Noise & Benign Input). *Benign noise* $\delta_{\text{ben}} \in \mathbb{R}^{K\times P}$ consists of unintentional distortions in CSI measurements resulting from typical operational conditions and environmental variability. A *benign input*, $x_{\text{ben}} = x_{\text{org}} + \delta_{\text{ben}}$ is an observed CSI sample in which the additive noise δ_{ben} which does not impact the prediction tasks.

Definition 2 (Adversarial Noise & Adversarial Input). *Adversarial noise* $\delta_{\text{adv}} \in \mathbb{R}^{K\times P}$ refers to carefully crafted perturbations aimed at degrading model performance or causing misclassification. An *adversarial input*, $x_{\text{adv}} = x_{\text{org}} + \delta_{\text{adv}}$, is an intentionally perturbed version of the true CSI signal x_{org}, where the additive noise δ_{adv} is crafted to induce incorrect predictions.

Defense Objective. The defense primary task is to detect whether x_{test} corresponds to a benign input ($x_{\text{test}} \stackrel{?}{=} x_{\text{ben}}$) or an adversarial input ($x_{\text{test}} \stackrel{?}{=} x_{\text{adv}}$). However, directly inferring this from x_{test} is challenging, as the noise component is entangled within the measurement. To address this, we closely follow [33] and reformulate the problem by focusing on the *nature of the noise* δ_{test}. Specifically, we aim to determine whether δ_{adv} is benign ($\delta_{\text{test}} \stackrel{?}{=} \delta_{\text{ben}}$) or adversarial ($\delta_{\text{test}} \stackrel{?}{=} \delta_{\text{adv}}$). By isolating and analyzing δ_{test}, we can infer whether the input was naturally perturbed or maliciously manipulated. This decomposition-based perspective offers a more tractable and interpretable framework for adversarial detection.

Defense Assumptions. NoiFi operates under a realistic and constrained defense setting. The defender has no prior knowledge of the specific adversarial attack algorithm, strategy, or perturbation characteristics employed by the adversary. However, the defender possesses a small, trusted validation dataset $\mathcal{D}_{\text{valid}}$ consisting of clean samples that span all target classes. This dataset is used to train a denoising autoencoder (DAE) $\mathcal{A}$ that can disentangle the noise from the test data. Moreover, the defender is assumed to have full access to the target classifier $\mathcal{M}_{\text{target}}$, including the ability to extract both the final prediction and internal feature representations of any input. Importantly, we assume the DAE and the validation dataset remain uncompromised—secured against tampering, poisoning, or unauthorized access by the attacker.

2.5 A Motivating Example

This motivating example illustrates the inherent fragility of adversarial perturbations compared to benign noise. We qualitatively examine the feature representations of both types of noise before and after randomization. Specifically, we consider BIM [15] attacks on the *RoboFiSense* [51] dataset. To do so, we generate 100 benign noise δ_{ben} and 100 adversarial perturbations δ_{adv}, which are added to clean test samples. The resulting perturbed samples are then passed through an autoencoder to obtain their reconstructed representations, denoted η_{ben} and η_{adv}, respectively. To examine noise behavior after randomization, we apply a randomization operator (e.g., pixel-level permutation) $Rand(\cdot)$ to both the original and reconstructed noises, resulting in four categories of noise pairs: (i) original and randomized benign noise ($\delta_{ben}, Rand(\delta_{ben})$), (ii) original and randomized adversarial noise ($\delta_{\text{adv}}, Rand(\delta_{\text{adv}})$), (iii) reconstructed and randomized benign noise ($\eta_{ben}, Rand(\eta_{ben})$), and (iv) reconstructed and randomized adversarial noise ($\eta_{\text{adv}}, Rand(\eta_{\text{adv}})$).

Fig. 2 visualizes these four pairs in 2D using PCA applied to their 512-dimensional feature embeddings through the target classifier. Fig. 2(a) presents the feature space of the original noise and its randomized variants, while Fig. 2(b) shows the same for the reconstructed samples. The left-most panel of Fig. 2(a), benign perturbations (δ_{ben}) and their randomized variants $Rand(\delta_{ben})$ cluster tightly, suggesting strong representational consistency. Conversely, as shown in the middle panel of Fig. 2(a), adversarial perturbations (δ_{adv}) exhibit broader dispersion, while their randomized versions $Rand(\delta_{\text{adv}})$ tend to form a more compact, yet distinctly shifted manifold. The right-most panel of Fig. 2(a) depicts the distribution of distances traversed by benign and adversarial noises after randomization. It is evident that the benign noises remain tightly clustered, whereas the adversarial perturbations shift substantially—mostly into the same benign cluster—resulting in significantly higher deviations.

Fig. 2(b) also reveals a consistent pattern in the reconstructed noises: benign noises (η_{ben}) preserve tight clustering with their randomized forms, while adversarial perturbations (η_{adv}) remain largely deviated. Overall, this analysis demonstrates that adversarial perturbations carry persistent representational footprints—both in their original and reconstructed forms—highlighting the potential of reconstruction-based noise sampling and target model-based representations in detecting and analyzing adversarial behavior. With this noise-centric approach, it is prominent (from the right-most panels) that the benign and the adversarial noises are nearly linearly separable in 2D projections, and even more markedly so in the full 512-dimensional feature space. Motivated by these observations, we introduce NoiFi in the following section.

3 Our Proposed Defense: NoiFi

This section presents our proposed defense NoiFi. We start with a working overview of NoiFi, which is followed by the details of individual components.

3.1 NoiFi Overview

NoiFi is designed to identify adversarially manipulated CSI by analyzing the latent structure of the accompanying noise. As illustrated in Fig. 3, NoiFi consists of three sequential modules: the noise sampling module (NSM), the noise representation module (NRM), and the noise discrepancy module (NDM), which we describe in the following.

The goal of the NSM is to effectively sample the accompanying noise from the raw test CSI data. For that, NSM utilizes a reconstruction-based approach and uses a DAE [37] to disentangle the noise from the clean CSI signal component. We assume that the defender has a CSI dataset that only contains benign samples. In the training phase, NSM trains the DAE on such benign CSI samples so it learns to reconstruct the clean input CSI from a noisy one. In the testing phase, NSM utilizes the pretrained DAE to reconstruct the noise from the test input. The *reconstructed noise*—defined as the difference between the input and its reconstruction—serves as the primary input for subsequent modules. For adversarial inputs, the reconstructed noise captures the structure of the adversarial perturbation, whereas for benign inputs, it exhibits no meaningful structure.

The objective of the NRM is to derive an effective representation of reconstructed noise that exposes the intrinsic differences between benign and adversarial perturbations. To this end, NRM incorporates a feature extractor (FE), instantiated by reusing the target model itself, thereby embedding domain knowledge into the representation process without requiring additional training. NRM operates in two stages. First, it constructs a sample-specific set of randomized noise variants by applying pixel permutations to the reconstructed noise. These permutations disrupt structural dependencies while preserving the overall magnitude statistics. By doing so, the randomized set highlights the inherent fragility of adversarial perturbations compared to benign noise, making the differences more distinguishable in the feature space. Second, both the test noise and its randomized variants are passed through the FE to obtain feature embeddings. The module thus produces (i) a feature vector corresponding to the test noise and (ii) a feature distribution derived from its randomized variants.

Finally, the goal of NDM is to detect if the test noise is benign or malicious. In doing so, it estimates the distance between the noise feature vector and the manifold formed by randomized noise features. NDM employs a lightweight outlier detection (OD) model to measure such deviation. For every test noise, NDM trains the OD model on randomized noise features and assigns an anomaly score to the test noise feature. The randomized noises—whether from benign or adversarial noise—consistently span a compact, low-variance manifold in latent space. Since benign noise lacks structured semantics, its representation typically aligns with this manifold, resulting in a low anomaly score. In contrast, adversarial noise preserves subtle structural patterns that align with the gradients of the feature extractor. This alignment produces a distinctive representation, resulting in a substantial deviation–hence higher anomaly score—from the randomized noise manifold. During training, these scores are computed on a validation dataset, and a threshold is determined to serve as the decision boundary during the testing phase. Algorithm 1 in Appendix B summarizes the overall steps of NoiFi, and in the following parts, we provide technical descriptions of each modules.

3.2 Noise Sampling Module

The purpose of the NSM is to sample the additive-noise component from received CSI measurements so that downstream detection modules can analyze it. Separating noise from CSI is challenging because the structured benign signal and random perturbations—whether from natural variation or adversarial manipulation—are tightly superimposed in the same spectral and spatial domains. Classical filtering methods can serve as a crude proxy but typically fail to disentangle these components since they do not exploit domain structure, motivating a data-driven approach. Accordingly, as the noise sampler, we adopt a reconstruction-based strategy and utilize a DAE $\mathcal{A}$. We train $\mathcal{A}$ on a CSI dataset $\mathcal{D}_{\text{val}} = \{x_i, y_i\}$ that contains only benign samples ($y_i = 0$). Let $x_{\text{org}} \in \mathbb{R}^{K\times P}$ denote an original CSI sample from $\mathcal{D}_{\text{val}}$. During training, we form noisy inputs $x_{\text{noi}} = x_{\text{org}} + \delta_{\text{noi}}$ where $\delta_{\text{noi}} \sim \mathcal{N}(0, \sigma^2)$, and optimize $\mathcal{A}$ to reconstruct x_{org} from x_{noi}. This encourages $\mathcal{A}$ to learn a compact latent representation that captures the clean-signal manifold while suppressing additive noise. (Equivalently, one may minimize the reconstruction loss $\mathcal{L} = \|\mathcal{A}(x_{\text{noi}}) - x_{\text{org}}\|^2$).

Our testing objective is to determine whether a CSI sample x_{test} is benign (x_{ben}) or adversarial (x_{adv}) (see Section 2.4). To this end, NSM utilize the trained $\mathcal{A}$, to reconstruct the test input $\hat{x}_{\text{test}}$, and explicitly sample the additive perturbation η_{test}, as follows: $\hat{x}_{\text{test}} = \mathcal{A}(x_{\text{test}}) \approx x_{\text{org}}$ and $\eta_{\text{test}} = x_{\text{test}} - \hat{x}_{\text{test}} \approx x_{\text{test}} - x_{\text{org}} = \delta_{\text{test}}$. When $\mathcal{A}$ is well trained on benign CSI, the same approximation holds for benign and adversarial inputs: $\hat{x}_{\text{ben}} = \mathcal{A}(x_{\text{ben}}) \approx x_{\text{org}}$ and $\hat{x}_{\text{adv}} = \mathcal{A}(x_{\text{adv}}) \approx x_{\text{org}}$, yielding residuals: $\eta_{\text{ben}} = x_{\text{ben}} - \hat{x}_{\text{ben}} \approx \delta_{\text{ben}}$ and $\eta_{\text{adv}} = x_{\text{adv}} - \hat{x}_{\text{adv}} \approx \delta_{\text{adv}}$. Thus, these reconstructed noises approximate the added noise—either benign or adversarial, offering a fertile ground for discrimination. By framing adversarial *input* detection as the task of adversarial *noise* detection, we shift the emphasis from discriminating between the inputs to discriminating between the two reconstructed noises (η_{ben} vs η_{adv}).

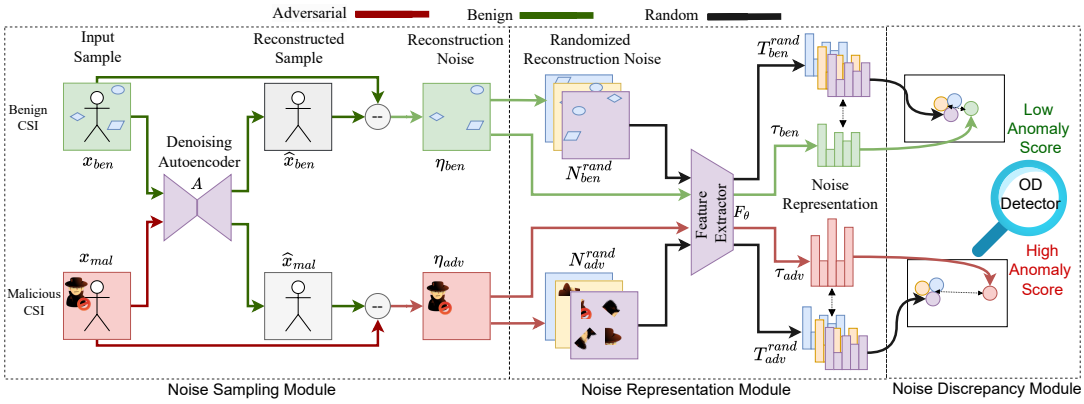


Figure 3: Overview of the inference process in NoiFi across three modules. The upper example shows a benign CSI input that results in a low anomaly score, while the lower example shows an adversarial input that yields a higher anomaly score.

3.3 Noise Representation Module

While the NSM can effectively isolate the noise from the CSI signal, the extracted noise alone does not directly reveal meaningful indicators for adversarial characterization. Two major challenges arise in leveraging this raw noise for effective detection. First, the extracted noise retains the same *high dimensionality* as the CSI matrix, which is inherently large. Second, in such a high-dimensional space, benign and adversarial noise often exhibit superficially similar structures, leading to *ineffective representation* for further discrimination for OD methods. Dimensionality reduction techniques such as PCA [8], t-SNE [23], etc., can compress the data, but they operate by maximizing variance, which is ineffective when the input is noise-like. To address these limitations, we design NRM where its core component is a *feature extractor* $\mathcal{F}_\theta$, which maps any given noise input into a low-dimensional feature space tailored for downstream adversarial detection. To incorporate *domain-specific knowledge*, $\mathcal{F}_\theta$ is instantiated by reusing a subset of the parameters from the target model $\mathcal{M}_\theta$. This ensures that the extracted features are sensitive to the same input patterns that the target model itself has learned, thereby enabling the detection of perturbations designed to exploit those learned representations. Hence, we consider splitting $\mathcal{M}_\theta$ through its penultimate layer, and consider the first part as the $\mathcal{F}_\theta : \mathbb{R}^{K\times P} \rightarrow \mathbb{R}^F$.

Let us consider that NRM takes the test reconstructed noise $\eta_{\text{test}} \in \mathbb{R}^{K\times P}$ where $test \in \{ben, adv\}$ and executes two key tasks on it. *First*, regardless of the noise type, NRM performs stochastic randomization on η_{test} to generate multiple randomized variants. Let $\mathcal{R}and$ be a randomization operator (e.g., pixel level permutation) that destroys structured content and creates N randomized instances. Therefore, we can express this as:

$$\mathcal{N}_{\text{test}}^{\text{rand}} = \{\eta_{\text{test}}^{(1)}, \eta_{\text{test}}^{(2)}, \ldots, \eta^{(N)}\} = \mathcal{R}and(\eta_{\text{test}}).$$

Although η_{test} is a single noise sample, $\mathcal{N}_{\text{test}}^{\text{rand}} \in \mathbb{R}^{N\times K\times P}$ represents a distribution of randomized noise of η_{test}. *Second*, NRM utilizes the $\mathcal{F}_\theta$ to generate feature representation(s) of both η_{test} and $\mathcal{N}_{\text{test}}^{\text{rand}}$, denoted by $\tau_{\text{test}} = \mathcal{F}_\theta(\eta_{\text{test}}) \in \mathbb{R}^F$ and $\mathcal{T}_{\text{test}}^{\text{rand}} = \mathcal{F}_\theta(\mathcal{N}_{\text{test}}^{\text{rand}}) \in \mathbb{R}^{N\times F}$. In the benign case, τ_{ben} lies on the manifold induced by $\mathcal{T}_{\text{ben}}^{\text{rand}}$, whereas in the adversarial case, τ_{adv} lies outside the manifold formed by $\mathcal{T}_{\text{adv}}^{\text{rand}}$. We formalize this observation in the following two propositions.

Proposition 3 (On-Manifold Feature Concentration for Benign Inputs). *Let η_{ben} denote the reconstruction noise corresponding to a benign test input x_{ben}. Let $\mathcal{N}_{\text{ben}}^{\text{rand}}$ be a set of randomized variants of η_{ben}. Then, the feature distribution $\mathcal{T}_{\text{ben}}^{\text{rand}} = \mathcal{F}_\theta(\mathcal{N}_{\text{ben}}^{\text{rand}})$ is concentrated within a low-variance manifold, and the feature vector $\tau_{\text{ben}} = \mathcal{F}_\theta(\eta_{\text{ben}})$ lies within this manifold.*

Proof. See Appendix C.1 for proof. □

Proposition 4 (Off-Manifold Feature Divergence for Adversarial Perturbation). *Let η_{adv} denote the reconstruction noise corresponding to an adversarial test input x_{adv}. Let $\mathcal{N}_{\text{adv}}^{\text{rand}}$ be a set of randomized variants of η_{adv}. Then, the feature distribution $\mathcal{T}_{\text{adv}}^{\text{rand}} = \mathcal{F}_\theta(\mathcal{N}_{\text{adv}}^{\text{rand}})$ is concentrated within a low-variance manifold, but the feature vector $\tau_{\text{adv}} = \mathcal{F}_\theta(\eta_{\text{adv}})$ significantly outside of this manifold.*

Proof. See Appendix C.2 for proof. □

3.4 Noise Discrepancy Module

The task of NDM is to determine whether a given noise representation is benign or malicious. Whereas existing works [25, 33] consider a predefined benign dataset and Gaussian-like noise to pre-train the OD model, this assumption fails in dynamic, real-world systems such as WiFi sensing. The noise distributions in practice are highly diverse, making it infeasible to guarantee a stable and Gaussian-like benign noise baseline. Hence, to overcome these limitations, we followed a dynamic and online OD approach in designing NDM that avoids reliance on any pre-collected benign dataset or noise baseline. Instead, the training distribution is derived directly from the test instance itself on a per-sample basis, as described in the 3.3. The NDM quantifies the deviation of the vector τ_{test} from the distribution $\mathcal{T}_{\text{test}}^{\text{rand}}$. This deviation serves as a statistical score of outliers, where a larger deviation indicates a higher likelihood of adversarial perturbation. To measure this deviation in a dynamic and distribution-agnostic manner, NDM considers a scalable OD—either based on statistical distance and density estimation. Assuming that the distribution $\mathcal{T}_{\text{test}}^{\text{rand}}$ forms a

concentrated manifold in the feature space, NDM, by default, considers the mean Euclidean distance (MED) as the anomaly score. Thus, the anomaly score α_{test} is defined as the average ℓ_2 distance from the test feature τ_{test} to all feature vectors in the manifold:

$$\alpha_{\text{test}} = \frac{1}{|\mathcal{T}_{\text{test}}^{\text{rand}}|} \sum_{\tau \in \mathcal{T}_{\text{test}}^{\text{rand}}} \|\tau_{\text{test}} - \tau\|_2. \tag{8}$$

Although MED directly measures the deviation of the test feature from the manifold, based on the distribution of the noise features in different application domains or setups, other OD methods such as K-Nearest Neighbors (KNN) [3], KDE-based detection [16], or One-class support vector machine (OCSVM) [31] can also be utilized, which are explained in Appendix D. The OD method can effectively discriminate between the benign and adversarial noises, which are formalized in the following lemmas.

Lemma 5 (Benign Noise Characterization). *For a benign scenario, if OD returns the anomaly score α_{ben}, then τ_{ben} and $\mathcal{T}_{\text{ben}}^{\text{rand}}$ hold the following properties:*

$$\tau_{\text{ben}} \in \text{support}(\mathcal{T}_{\text{ben}}^{\text{rand}}) \implies \alpha_{\text{ben}} = OD(\tau_{\text{ben}}, \mathcal{T}_{\text{ben}}^{\text{rand}}) \text{ is low.}$$

Proof. Given η_{ben} and $\mathcal{N}_{\text{ben}}^{\text{rand}}$ representing reconstructed and randomized noise originating from benign scenarios, the induced feature representations are statistically similar. This implies that τ_{ben} is located near the centroid of the distribution $\mathcal{T}_{\text{ben}}^{\text{rand}}$, resulting in a low distance $OD(\tau_{\text{ben}}, \mathcal{T}_{\text{ben}}^{\text{rand}})$. Since $\mathcal{T}_{\text{ben}}^{\text{rand}}$ captures this natural stochasticity, the deviation α_{ben} remains within a constant bound i.e., $\alpha_{\text{ben}} \approx 0$. □

Lemma 6 (Adversarial Noise Characterization). *For an adversarial scenario, τ_{adv} and $\mathcal{T}_{\text{adv}}^{\text{rand}}$ hold the following properties:*

$$\tau_{\text{adv}} \notin \text{support}(\mathcal{T}_{\text{adv}}^{\text{rand}}) \implies \alpha_{\text{adv}} = OD(\tau_{\text{adv}}, \mathcal{T}_{\text{adv}}^{\text{rand}}) \text{ is high.}$$

Proof. Although $\mathcal{T}_{\text{adv}}^{\text{rand}}$ and $\mathcal{T}_{\text{ben}}^{\text{rand}}$ are generated from different sources of noise—adversarial and benign, respectively—their randomized nature leads to statistically similar distributions. That is, $\mathcal{T}_{\text{adv}}^{\text{rand}} \approx \mathcal{T}_{\text{ben}}^{\text{rand}}$ in distributional shape and spread. However, the critical distinction lies in the feature representation of η_{adv}. Unlike τ_{ben}, which produces feature embeddings that remain within the high-density manifold of $\mathcal{T}_{\text{ben}}^{\text{rand}}$, τ_{adv} tends to lie outside the support of $\mathcal{T}_{\text{adv}}^{\text{rand}}$ as adversarial perturbations have gradient alignment and are crafted to induce such targeted shifts. This out-of-distribution behavior yields a large manifold distance $OD(\tau_{\text{adv}}, \mathcal{T}_{\text{adv}}^{\text{rand}})$, indicating a substantial deviation from expected feature behavior, inducing $\alpha_{\text{adv}} \gg 0$. □

During the training phase, anomaly scores are computed for all benign samples within $\mathcal{D}_{\text{val}}$, and a detection threshold α_{th} is then determined as the p-th percentile of the resulting score distribution or based on the allowed false positive rates. In the testing phase, the anomaly score α_{test} of a test input x_{test} is computed and compared against α_{th}; the input is classified as adversarial if $\alpha_{\text{test}} > \alpha_{\text{th}}$, and benign otherwise. Algorithm 1 shows the overall detection steps for the detection of an adversarial CSI sample.

4 Implementation

4.1 Datasets

To evaluate the effectiveness of NoiFi, we consider two wireless sensing datasets:

***UT-HAR* Dataset**. First, we utilize *UT-HAR*, a human activity recognition dataset, derived from WiFi CSI measurements [50]. It consists of amplitude-only CSI values captured across 90 channels, formed by combining 30 subcarriers with 3 antenna configurations. Each CSI instance comprises 500 consecutive timestamps, resulting in an input dimensionality of 90 × 500. The dataset includes seven distinct human activities. such as sit-down, stand-up, and walk, capturing a diverse range of motion patterns.

***RoboFiSense* Dataset**. For robotic activity recognition, we employ the *RoboFiSense* dataset [51], which records CSI corresponding to eight different types of motion trajectories executed by a robotic arm. These trajectories include linear displacements, angular motions, and structured geometric patterns such as arcs, rectangles, and triangles. CSI was captured using two spatially separated sniffers, resulting in high-resolution CSI data of size 360 x 360, enabling a fine-grained robotic activity classification.

4.2 Model Architectures

Here we describe our selection of classifiers, noise samplers, and OD methods for the evaluation.

Classifier Models. To evaluate NoiFi against both white-box and black-box adversarial attacks, we consider two types of classification models: a *target model* and a lightweight *surrogate model*. Both models are based on convolutional neural networks (CNN), as summarized in Table 3 (in Appendix E), that include batch normalization, ReLU activations, max pooling, and dropout. To further show the generalization of NoiFi, we also evaluate NoiFi with off-the-shelf ResNet [12] architectures—ResNet-34 and ResNet-18 as the target and surrogate models, respectively (results are shown in Appendix I.2).

Noise Sampling Models. As the noise sampler, we primarily use a data-driven solution, such as a DAE that consists of 6 convolutional layers in the encoder, followed by 2 fully connected layers, and 6 deconvolutional layers in the decoder. A Gaussian noise with a standard deviation of 0.05 is added during training to improve denoising. The dimensionality of the latent space is set to 1024 for both datasets. Generally, it is observed that the high-frequency components in CSI amplitude matrices are often dominated by noise rather than meaningful signal structure. Hence, to further show the effectiveness of NoiFi beyond DAE, we also consider two classical signal processing-based filtering methods using: (ii) two-dimensional discrete cosine transform (DCT) [6] filtering, and (iii) low-pass filtering (LPF) [9]. In both cases, during the reconstruction, we retain the 98% frequency components and hence, sample the top 2% high frequency components as the target noise.

Outlier Detection Models. Moreover, as an OD, we consider four candidate algorithms—MED, KNN, KDE, and OCSVM. For MED, we directly calculated the mean Euclidean distance from the samples to the randomized distribution. For the remaining models, we use the default configuration from PyOD [52]. For instance, for KNN, we use the default setting with $k = 5$ to measure

Table 1: ASR and L2 Norm of adversarial perturbation under different datasets, access types, and attacks.

Dataset	Access	FGSM		BIM		PGD		UAP	
		ASR	L2	ASR	L2	ASR	L2	ASR	L2
UT-HAR	W-Box	72.0	131.55	100.0	128.42	100.0	126.95	72.0	130.24
	B-Box	80.0	132.43	100.0	128.23	100.0	129.08	69.0	135.82
RoboFiSense	W-box	87.0	268.38	100.0	225.27	100.0	238.95	85.0	261.23
	B-box	86.0	268.35	89.0	222.70	92.0	238.38	88.0	261.39

neighborhood-based deviation, KDE is configured with a bandwidth of 1.0, and OCSVM with an RBF kernel with $\gamma = 1/512$.

4.3 Evaluation Settings

Adversarial Attacks and Benign Baselines. We evaluate NoiFi against the adversarial attacks described in Section 2.3. For each attack, we generate 100 adversarial perturbations δ_{adv} using both the target and surrogate models, which we use in evaluating in white-box and black-box settings, respectively. Representative adversarial perturbations and inputs from different attacks and datasets are shown in Table 5. To ensure a fair evaluation, we evaluate the performance of NoiFi by distinguishing between structured adversarial perturbations and unstructured benign perturbations. We construct such *randomized benign perturbations* δ_{ben} by permuting δ_{adv} so each preserves the ℓ_p magnitude of the adversarial perturbation while eliminating its gradient-aligned structure. Such randomized adversarial baselines are commonly used to assess whether defenses respond to adversarial *structure* rather than just perturbation energy [29]. For the defense, we consider the classifier's feature dimension F as 512, and the number of randomized noise N as 5.

To further evaluate the generalization of NoiFi against other natural noise baselines that mimic real-world disturbances, we consider three more representative cases:

- *Gaussian noise*, sampled from $\mathcal{N}(0, \sigma^2)$, is often employed to approximate channel fluctuations and hardware-induced distortions in WiFi sensing [13, 29]. We consider σ^2 so that the noise mimics the distributional strength of the adversarial noises.
- *Impulse noise*, also known as salt-and-pepper corruption, represents sparse but high-magnitude interference or bit errors [13]. We model this noise by replacing about 1% of the CSI values with random noise, mimicking such impulsive distortions.
- *Burst noise* introduces short, localized spikes or clusters of distortion, analogous to transient bursts in CSI measurements, which often stem from hardware imperfections or interference in the channel [30]. We model this by randomly adding multiple localized patches (e.g., 5×5) of random intensity into the CSI matrix.

Evaluation Metrics. To rigorously assess the efficacy of adversarial attacks and the robustness of the defense mechanism, we adopt a suite of standard metrics commonly used in adversarial machine learning and binary classification tasks. These include the *attack success rate (ASR)* to quantify the strength of the adversary, and metrics such as area under the ROC curve (*AUROC*), area under the PR curve (*AUPRC*), *Precision*, *Recall*, and *F1 Score* to evaluate the performance of the defense in distinguishing adversarial from benign inputs.

Software Implementation. NoiFi is implemented in Python 3.10, with the classifier and autoencoder models developed using PyTorch. Adversarial examples are generated using the Torchattacks [14] and Adversarial Robustness Toolbox (ART)[27] libraries. For OD models, we leverage the PyOD library[52]. All experiments are conducted on a machine running Ubuntu 18.04.3, equipped with an Intel Core i7-8700K CPU @ 3.70GHz and an NVIDIA GeForce RTX 2080 Ti GPU.

5 Results

This section summarizes the evaluation results of NoiFi from different aspects. Table 1 summarizes the ASR and L2 norm of the adversarial perturbation under both white-box and black-box attacks. Throughout the evaluation of the efficacy of NoiFi in detecting these attacks, we respond to the following four research questions:

- **RQ1**: Can adversarial noise, or its reconstruction, be reliably detected? (Section 2.5)
- **RQ2**: How effectively does NoiFi defend against various adversarial attacks in both white-box and black-box settings? (Section 5.1)
- **RQ3**: Does NoiFi generalize to different baseline noises and classical noise sampling methods? (Section 5.2)
- **RQ4**: Can NoiFi generalize beyond WiFi sensing? (Section 5.2.3)
- **RQ5**: Is NoiFi scalable for near-real-time deployment in WiFi-based recognition systems? (Section 5.3)
- **RQ6**: Can NoiFi defend against adaptive adversarial attacks that optimize both effectiveness and stealth? (Section 5.4)

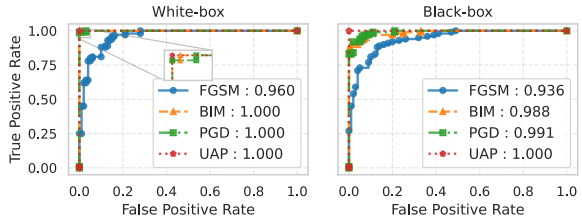
5.1 Overall Detection Performance of NoiFi

We evaluate the adversarial attack detection performance of NoiFi from two key perspectives. First, we present the ROC and PR curves along with the corresponding AUROC and AUPRC scores, respectively. Second, we report detailed classification metrics—including precision, recall, and F1 score—at a threshold point that maintains the FPR below 1%. All evaluations are conducted under both white-box and black-box threat models. For this evaluation, we consider the default setting of NoiFi, such as a CNN-based classifier, DAE as the noise sampler, and MED as the OD, and benign perturbation as the noise baseline.

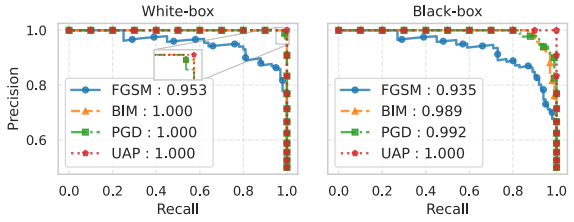
5.1.1 ***TPR–FPR and Precision–Recall Trade-off.*** Fig. 4 and Fig. 5 demonstrate the detection performance of the proposed defense NoiFi on the *UT-HAR* and *RoboFiSense* datasets, respectively. The results consistently demonstrate the effectiveness of NoiFi in handling a wide spectrum of attacks. On the *UT-HAR* dataset, NoiFi achieves AUROC scores of 0.960–1.000 in the white-box setting and 0.936–1.000 in the black-box setting, with corresponding AUPRC values ranging from 0.935–1.000. The *RoboFiSense* dataset further highlights the robustness of the approach, where most adversarial settings yield near-perfect detection with AUROC and AUPRC values at or extremely close to 1.000. These results confirm that NoiFi not only generalizes well across datasets but also sustains exceptional detection performance against both single-step and iterative adversaries. More importantly, despite the transferability gap between the surrogate and target models, NoiFi consistently captures the adversarial characteristics in the black-box setting, even when the attacks fail to fool the target model. Furthermore,

Table 2: Performance metrics per Attack, Detector, Threat Model, and Dataset.

Dataset	Threat	Detector	FGSM AUROC	FGSM AUPRC	FGSM Pre	FGSM Rec	FGSM F1	BIM AUROC	BIM AUPRC	BIM Pre	BIM Rec	BIM F1	PGD AUROC	PGD AUPRC	PGD Pre	PGD Rec	PGD F1	UAP AUROC	UAP AUPRC	UAP Pre	UAP Rec	UAP F1
UT-HAR	White-box	Artifacts	0.663	0.648	0.663	0.010	0.020	0.581	0.563	0.581	0.010	0.020	0.844	0.874	0.844	0.510	0.671	0.703	0.740	0.703	0.210	0.344
		MagNet	0.980	**0.990**	0.980	**0.980**	**0.990**	0.821	0.784	0.821	0.060	0.112	0.729	0.677	0.729	0.030	0.058	0.578	0.573	0.578	0.040	0.076
		Manda	0.958	0.948	0.958	0.530	0.688	0.247	0.389	0.247	0.000	0.000	0.229	0.379	0.229	0.000	0.000	0.585	0.688	0.585	0.000	0.000
		NoiSec	**0.984**	0.970	**0.984**	0.290	0.446	1.000	1.000	1.000	0.990	0.995	0.999	0.999	0.999	**0.990**	**0.995**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		NoiFi	0.960	0.953	0.960	0.450	0.616	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**1.000**	**1.000**	**1.000**	**0.990**	**0.995**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Black-box	Artifacts	0.580	0.573	0.580	0.010	0.020	0.635	0.623	0.635	0.050	0.094	0.816	0.856	0.816	0.360	0.526	0.665	0.604	0.665	0.030	0.058
		MagNet	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.767	0.768	0.767	0.130	0.228	0.790	0.777	0.790	0.160	0.274	0.381	0.432	0.381	0.000	0.000
		Manda	0.965	0.969	0.965	0.760	0.864	0.352	0.406	0.352	0.000	0.000	0.313	0.389	0.313	0.000	0.000	0.985	0.988	0.985	0.860	0.925
		NoiSec	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**0.995**	**0.995**	**0.900**	**0.942**	**0.998**	**0.997**	**0.998**	**0.930**	**0.959**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		NoiFi	0.936	0.935	0.936	0.450	0.616	0.988	0.989	0.988	**0.900**	**0.942**	0.991	0.992	0.991	0.840	0.908	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
RoboFiSense	White-box	Artifacts	0.632	0.576	0.632	0.020	0.039	0.751	0.736	0.751	0.030	0.058	0.800	0.759	0.800	0.020	0.039	0.553	0.517	0.553	0.000	0.000
		MagNet	0.979	0.964	0.979	0.300	0.458	0.548	0.600	0.548	0.040	0.076	0.474	0.541	0.474	0.030	0.058	0.931	0.933	0.931	0.510	0.671
		Manda	0.651	0.537	0.651	0.000	0.000	0.740	0.618	0.740	0.000	0.000	0.652	0.545	0.652	0.000	0.000	0.761	0.673	0.761	0.000	0.000
		NoiSec	0.837	0.843	0.837	0.270	0.422	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.998	0.998	0.998	0.950	0.969
		NoiFi	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Black-box	Artifacts	0.633	0.642	0.633	0.090	0.164	0.683	0.673	0.683	0.000	0.000	0.748	0.751	0.748	0.060	0.112	0.592	0.658	0.592	0.050	0.094
		MagNet	0.985	0.972	0.985	0.570	0.722	0.522	0.591	0.522	0.060	0.112	0.478	0.541	0.478	0.030	0.058	0.918	0.925	0.918	0.430	0.597
		Manda	0.649	0.535	0.649	0.000	0.000	0.802	0.683	0.802	0.000	0.000	0.769	0.664	0.769	0.000	0.000	0.613	0.559	0.613	0.000	0.000
		NoiSec	0.800	0.792	0.800	0.140	0.243	0.999	0.999	0.999	**1.000**	**0.995**	0.998	0.998	0.998	0.960	0.975	0.824	0.828	0.824	0.200	0.331
		NoiFi	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**0.999**	**0.999**	**0.999**	**0.970**	**0.985**	**0.993**	**0.993**	**0.993**	**0.930**	**0.959**

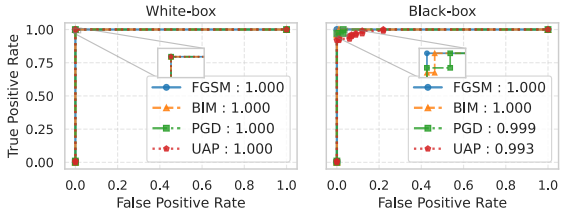


(a) ROC Curve with AUROC Scores.

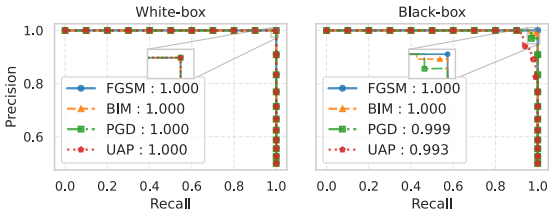


(b) PR Curve with AUPRC Scores

Figure 4: Detection performance on *UT-HAR* dataset.



(a) ROC Curve with AUROC Scores.



(b) PR Curve with AUPRC Scores

Figure 5: Detection performance on *RoboFiSense* dataset.

the defense preserves an almost perfect balance between false positives and true positives, as well as precision and recall—a property of paramount importance for real-world deployments.

It is worth noting that FGSM, being a single-step attack, is inherently less powerful than iterative counterparts such as BIM and PGD. As a result, its ASR is lower, and in some cases, the perturbed samples fail to retain strong adversarial properties. This explains why NoiFi has relatively lower AUROC and AUPRC scores (e.g., 0.936 and 0.935 on *UT-HAR* black-box FGSM) compared to its performance on stronger attacks. Conversely, the consistently near-perfect detection performance against iterative attacks demonstrates the ability of NoiFi to capture stronger and more transferable adversarial perturbations.

5.1.2 ***Detailed Results with Baseline Comparison.*** We evaluate NoiFi with MED against the closest baselines (i.e., MagNet [25], Artifact [7], Manda [39], and NoiSec [33]) under the same evaluation setting as mentioned above. Table 2 summarizes NoiFi 's detection performance across adversarial attacks, datasets, and threat models. Across nearly all scenarios, NoiFi delivers perfect or near-perfect AUROC, AUPRC, Precision, Recall, and F1, with particularly strong results against iterative attacks. On *RoboFiSense*, NoiFi reaches AUROC and AUPRC of 1.000 across all attacks in both white- and black-box settings, with Precision, Recall, and F1 likewise saturating at 1.000. Performance on the *UT-HAR* dataset is similarly robust, sustaining AUROC/AUPRC = 1.000 in most cases, with only minor deviations under weaker single-step FGSM attacks. These findings establish NoiFi as a consistently reliable detector across datasets, threat models, and attack types.

Against baselines, NoiFi demonstrates superiority in most cases. The closest competitor, NoiSec, remains strong (especially on *UT-HAR* dataset) but falls short in most cases—for example, on *UT-HAR* white-box FGSM, NoiFi achieves an F1 of 0.616 versus NoiSec's

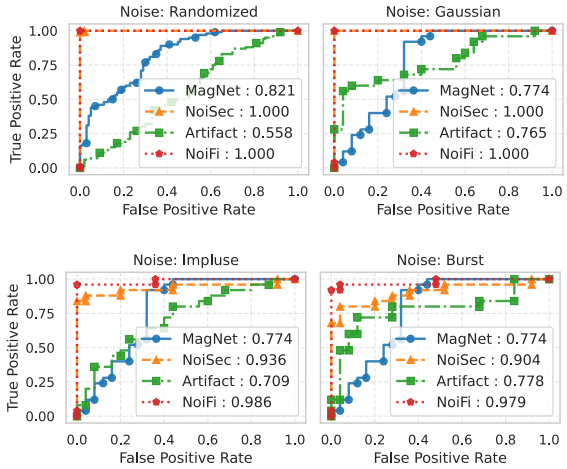


Figure 6: ROC curves (with AUROC scores) for NoiFi and other baseline detectors evaluated on BIM attack with different benign noise distributions.

0.446. Other baselines, including MagNet, Manda, and Artifacts, show occasionally elevated performance, but often collapse under black-box scenarios (e.g., *RoboFiSense* black-box FGSM, where MagNet recall = 0.058 and F1 = 0.091 vs. NoiFi 's perfect detection). Together, these results highlight NoiFi 's robustness not only against white-box adversaries but also against transfer-based black-box attacks, where competing detectors degrade substantially. It is noted that NoiSec employs a pre-trained OD method trained with Gaussian noise, and this evaluation also considers benign perturbation (similar to Gaussian noise) as the benign baseline. In contrast, NoiFi makes no assumptions about baseline noise and achieves this performance entirely without relying on any model or data. Overall, NoiFi still achieves an average recall improvement of approximately 97% under white-box attacks and 147% under black-box attacks compared to NoiSec. These results highlight NoiFi's superior generalization across diverse datasets and threat models, establishing it as a more reliable and practical defense compared to the current state-of-the-art.

5.2 Generalization Evaluation of NoiFi

This part analyzes the generalization of NoiFi by evaluating with different baseline noises, noise samplers, and even different modalities of datasets beyond Wifi sensing.

5.2.1 ***Different Noise Distribution.*** Figure 6 reports the detection performance against BIM attacks when benign noises are drawn from four distinct noise families (Randomized adversarial perturbation, Gaussian, Impulse, and Burst). The performances are also compared across baseline detectors. In the case of randomized and Gaussian noises, the benign baseline matches the autoencoder and OD training distribution (Gaussian); hence, the reconstruction-based methods, particularly NoiFi and NoiSec, attain excellent performance: NoiSec reaches perfect discrimination (AUROC = 1.000) while MagNet and Artifact show moderate performance (AUROC = 0.821 & 0.774 and 0.558 & 0.765, respectively). Our proposed NoiFi also attains near-perfect discrimination (AUROC = 1.000) in this setting.

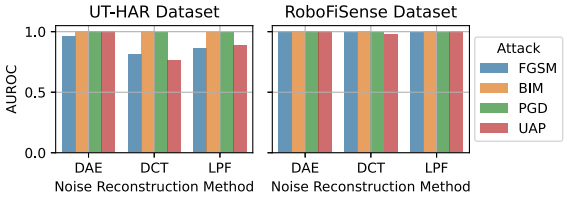


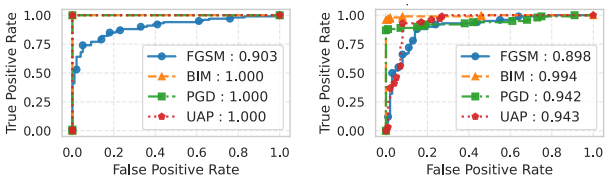
Figure 7: Compatibility of NoiFi with different noise sampling methods against different attacks and datasets.

Substituting a different benign noise distribution reveals the limitations of detectors that implicitly assume Gaussian noise during the OD training. Under Impulse noise, MagNet and Artifact degrade (AUROC = 0.774 and 0.709), and NoiSec drops to AUROC = 0.936. Under Burst noise, the trend continues: MagNet and Artifact remain moderate (AUROC = 0.774 and 0.778), and NoiSec decreases to AUROC = 0.904. By contrast, NoiFi remains highly effective across these challenging baselines (AUROC = 0.986 on Impulse and 0.979 on Burst), demonstrating strong discrimination between benign samples with practical noises and BIM adversarial examples despite its DAE being trained only with Gaussian-denoising data. These results indicate that (i) baseline detectors perform well when the deployment noise matches their training assumption but suffer when the benign noise distribution shifts, and (ii) NoiFi generalizes robustly across noise types and retains high AUROC against BIM attacks, making it more reliable in unpredictable noisy environments.

5.2.2 ***Effectiveness of Different Noise Sampler.*** This part studies the compatibilities of the classical signal processing-based filtering methods in NSM for noise sampling. We, along with the DAE, consider DCT and LPF as the noise sampler. Fig. 7 presents the AUROC across two datasets under multiple adversarial attacks. We observe that DAE consistently delivers the strongest performance, achieving near-optimal AUROC across all settings, which underscores the power of data-driven reconstruction for adversarial detection. However, the effectiveness of DAE is inherently contingent on the availability of representative training data, limiting its robustness under distributional shifts or in deployment scenarios with limited training resources.

By contrast, NoiFi also fits tightly with the signal-processing-based approaches, LPF and DCT, which provide competitive and stable detection performance without relying on data-driven training. Notably, despite a slight decrease in the AUROC for FGSM and UAP in *UT-HAR* dataset, their robustness on *RoboFiSense* demonstrates that NoiFi can also effectively capture the adversarial signatures from the noise sampled through LPF- or DCT-based filtering, making the deployment of NoiFi completely model and training-free. These results highlight a key trade-off: while DAE achieves the highest detection performance when training data is sufficient, LPF and DCT constitute strong, generalizable alternatives that also offer reliable detection in data-constrained settings.

5.2.3 ***Generalization Beyond Wireless Sensing.*** While NoiFi demonstrates strong effectiveness in detecting adversarial perturbations within wireless sensing applications, we further evaluate its applicability beyond this domain. In particular, we investigate



(a) PneumoniaMNIST Dataset (b) SpeechCommands Dataset

Figure 8: Evaluation of NoiFi on medical imaging (PneumoniaMNIST) and speech recognition (SpeechCommands).

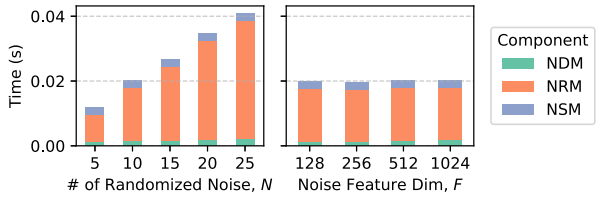


Figure 9: Scalability analysis of NoiFi for each modules.

whether NoiFi can generalize to other modalities, such as medical imaging and speech processing. Figure 8 presents the cross-domain generalization of NoiFi under its default configuration, as described in Section 5.1. We consider two representative datasets: PneumoniaMNIST[48], which contains 64×64 grayscale chest X-ray images for pediatric pneumonia detection, and SpeechCommands[43], which consists of 64×81 Mel-spectrograms from 35 spoken commands. As shown in Fig. 8, NoiFi consistently achieves strong detection performance across modalities and attack types. On PneumoniaMNIST, it attains near-perfect detection (AUC ≥ 0.903), while on SpeechCommands it maintains robust performance (AUC ≥ 0.898). These results highlight that NoiFi exploits a fundamental property of adversarial perturbations—their fragility under randomization—enabling it to reliably distinguish adversarial from benign inputs in a domain-agnostic manner. This underscores noise-aware modeling as a fundamental step toward cross-domain adversarial resilience.

5.3 Scalability Analysis of NoiFi

To assess the scalability of NoiFi, we analyze its end-to-end runtime across two hyperparameters: randomized noise size N and noise feature dimensionality F. Among these, only NRM is affected by variations in batch size. As shown in Fig. 9, increasing the N leads to a noticeable increase (although linearly) in runtime compared to increasing the F. This is primarily due to NRM, which must extract features for every N noise sample in the batch; as N grows, the number of forward passes through the FE increases proportionally, making NRM the dominant bottleneck. In contrast, increasing F affects the size of each sample's representation, but has a comparatively smaller impact on total inference time.

Despite these scaling effects, NoiFi remains highly efficient under practical configurations. Our evaluation shows, for N as 5 and F of 512—settings representative of standard ML deployment in WiFi sensing—NoiFi achieves an end-to-end runtime of approximately 10 milliseconds. This is well below the commonly accepted latency threshold of 100 milliseconds for responsive wireless sensing applications such as human activity recognition and device-free localization [41, 42]. These results demonstrate that NoiFi delivers near real-time performance while maintaining scalability across key operational parameters.

5.4 Evaluation against Adaptive Attacks

We consider a worst-case *fully adaptive* adversary with full knowledge of NoiFi's detection method, including the NRM and NDM (as described in Section 3.3 & 3.4). Hence, the attacker adapts the attack-generation process to jointly induce misclassification while explicitly minimizing the NoiFi's anomaly score as in (8).

5.4.1 ***Adaptive Attack Formulation.*** Let $x \in [0,1]^{K\times P}$ be a benign input with label y, and let x_{adv} denote its adversarial counterpart with perturbation $\delta_{\text{adv}} = x_{\text{adv}} - x$, constrained by $\|\delta_{\text{adv}}\|_\infty \le \epsilon$. Since the attacker directly injects and controls the perturbation, it can optimize δ_{adv} instead of sampling it. Therefore, an adaptive adversary optimizes the following multi-objective optimization:

$$\max_{x_{\text{adv}}} \quad J_{\text{cls}}\big(f(x_{\text{adv}}), y\big) - w \cdot \mathbb{E}_r\left[\frac{1}{|\mathcal{T}_{\text{adv}}^{\text{rand}}|}\sum_{\tau\in\mathcal{T}_{\text{adv}}^{\text{rand}}}\big\|\tau_{\text{adv}} - \tau\big\|_2\right]$$

$$\text{s.t.} \quad \|x_{\text{adv}} - x\|_\infty \le \epsilon, \qquad x_{\text{adv}} \in [0,1]^{K\times P},$$

where $\tau_{\text{adv}} = \mathcal{F}_\theta(\delta_{\text{adv}})$ and $\mathcal{T}_{\text{adv}}^{\text{rand}} = \mathcal{F}_\theta(\mathcal{R}and(\delta_{\text{adv}}))$ follow the NoiFi pipeline, and $w \ge 0$ controls the trade-off between attack success and stealth.

5.4.2 ***Gradient-based Attacks.*** We design a gradient-based adaptive attack, named *AdaptivePGD*, that optimizes the above objective using an iteratively projected gradient ascent method. Since $\mathcal{R}and(\cdot)$ is stochastic, we apply expectation over transformation (EOT) with S Monte Carlo samples with r_s as the seed to estimate the gradient:

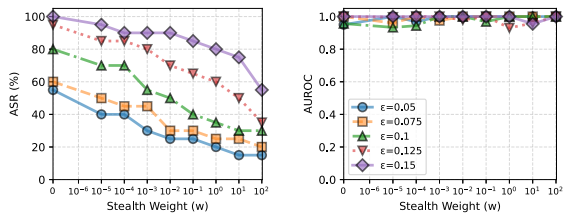
$$\hat{g}^{(t)} = \nabla_x J_{\text{cls}}\big(f(x_{\text{adv}}^{(t)}), y\big) - \frac{w}{S}\sum_{s=1}^{S}\nabla_x\left[\frac{1}{|\mathcal{T}_{\text{adv}}^{\text{rand}}|}\sum_{\tau\in\mathcal{T}_{\text{adv}}^{\text{rand}}}\big\|\tau_{\text{adv}}^{(t)} - \tau\big\|_2\right]_{r_s}.$$

Each iteration applies an L_∞-normalized projection as follows:

$$x_{\text{adv}}^{(t+1)} = \text{Clip}_{x,\epsilon}\left(x_{\text{adv}}^{(t)} + \alpha\cdot\frac{\hat{g}^{(t)}}{\|\hat{g}^{(t)}\|_\infty}\right),$$

where α is the step size. This formulation explicitly optimizes against the NoiFi anomaly score, providing a comprehensive evaluation of NoiFi under fully adaptive attack scenarios. To further ensure the worst-case attack cases, we generate multiple adversarial sample candidates for a single test input and select the one that induces misclassification and has the lowest anomaly score.

5.4.3 ***Gradient-free Attacks.*** Gradient-free attacks require querying confidence scores, loss values, or predicted labels [53] and incur substantial query overhead, resulting in high latency that makes them unsuitable for real-time systems such as WiFi sensing. Hard-label attacks [53], which observe only predicted class labels, further increase this overhead, often requiring thousands of queries. This conflicts with our threat model, where CSI inputs arrive continuously, and attackers operate under strict latency constraints. Empirical evaluation confirms this gap: gradient-based attacks are compatible with real-time operation (FGSM: 4.4 ms, 1 query; PGD: 80 ms, 25 queries), whereas gradient-free and hard-label attacks impose prohibitive overhead (Square [2]: 242 ms, 2,500 queries, 55x slower; HopSkipJump [5]: 37.2 s, 8,040 queries, 8,000x slower



(a) ASR under adaptive attacks **(b) AUROC under adaptive attacks**

Figure 10: Performance of NoiFi under adaptive attacks with varying stealth weights on the *UT-HAR* dataset.

than FGSM). Accordingly, we exclude gradient-free and hard-label attacks from our evaluation due to their infeasibility in real-time WiFi sensing systems.

5.4.4 ***Detection Performance.*** Here we evaluate NoiFi against the AdaptivePGD attack, where Table 4 (in Appendix F) shows the hyperparameters selected for the evaluation. Figure 10 presents two complementary views of the study with (i) the effectiveness of AdaptivePGD attacks that explicitly optimize for detection evasion, and (ii) the efficacy of NoiFi in detecting these stealthy adversarial behaviors. As shown in Fig. 10(a), introducing a non-zero stealth objective ($w > 0$) leads to a consistent and monotonic reduction in the ASR across all perturbation budgets ϵ. This trend highlights an inherent trade-off between attack stealthiness and maliciousness, where optimizing for stealth directly contradicts the attacker's goal of inducing misclassification. The monotonic degradation of ASR with increasing w indicates that NoiFi targets the key structural properties and root causes of adversarial perturbations; consequently, AdaptivePGD attacks are progressively pushed from a malicious adversarial noise space toward behavior resembling benign random noise, thereby weakening their impact.

Despite the reduced effectiveness of AdaptivePGD attacks at higher stealth, NoiFi still remains highly effective in detecting the remaining successful attacks. As illustrated in Fig. 10(b), the AUROC remains consistently high and close to unity across all values of ϵ and w, even when ASR is significantly reduced. This observation indicates that the successful adaptive attacks still retain sufficient malicious characteristics to be reliably distinguished from benign inputs by NoiFi, underscoring the robustness of the proposed defense under adaptive threat models.

6 Related Work

WiFi signals have been extensively used for device-free sensing tasks such as activity recognition, gesture identification, occupancy monitoring, and vital-sign detection. These systems typically extract CSI features from WiFi packet preambles and apply DL for classification [50]. Recent benchmarks demonstrate high accuracy in such applications, but also reveal new security vulnerabilities. In particular, DL models for WiFi sensing can be deceived by adversarial perturbations injected at both the digital and the physical layer. Li et al. introduced the first practical over-the-air attack against WiFi sensing [17], subtly modifying WiFi pilot symbols to control the CSI observed by the receiver. This packet perturbation attack can induce misclassification of activities without disrupting normal communication. Similarly, Zhou et al. propose WiAdv, which synthesizes adversarial wireless signals that mimic legitimate motion features [55] but achieved over high attack success while remaining robust across settings.

Other works focus on interference-style attacks. Li et al. [18] show that an external source of WiFi interference can serve as a simple adversarial trigger: by choosing the interference's spectral band and traffic pattern, they reduce a WiFi-based network traffic classifier's accuracy. Xu et al. [45] formalize "attack imperceptibility" and design *WiCAM*, which uses a DNN's temporal-spatial attention map to mask adversarial noise to only the most salient CSI components. Moreover, Lu et al. [20] proposed an imperceptible eavesdropping attack: a passive adversary outside the environment can infer users' private behaviors from intercepted CSI without alerting the system.

On the other hand, there are limited defenses, such as adversarial training, smoothing, and controlled perturbation, that have been proposed to mitigate these threats. Ambalkar et al. [1] demonstrate that adversarial training improves robustness in a WiFi-based apnea detection system. Yin et al. [49] apply adversarial training and randomized smoothing as a defense against white-box and black-box attacks on WiFi gesture models. At the radio level, Shankar and Chakraborty [34] propose a transmitter-side defense that injects small perturbations into outgoing WiFi signals. This causes an attacker's sensing model to misclassify almost all inputs while maintaining reliable communication. More broadly, some works use adversarial methods for privacy protection: e.g., Zhou et al. [54] train an adversarial network to modify CSI so as to obscure designated private activities while preserving recognition of others.

However, existing defenses are often limited in their ability to generalize across diverse types of adversarial attacks and frequently incur a significant degradation in the system's performance on benign inputs. In contrast, NoiFi introduces a model-agnostic detection framework that does not require modifications to the target model or retraining. Instead, it effectively identifies a broad spectrum of adversarial perturbations and offers a comprehensive layer of protection without compromising the system's benign accuracy.

7 Conclusion

In this work, we presented NoiFi, a lightweight and practical defense mechanism for securing WiFi sensing systems against adversarial perturbations. By exploiting the inherent fragility of adversarial noise, NoiFi dynamically constructs randomized noise manifolds that expose the rigid structure of adversarial manipulations. This design enables interpretable, attack-agnostic detection that operates online with near real-time efficiency. Extensive evaluations on multiple WiFi sensing datasets, along with transfer experiments across modalities such as medical imaging and speech, demonstrate that NoiFi achieves consistently high detection performance (AUROC/AUPRC up to 1.00), low false positives, and robustness against white-box, black-box, and adaptive attacks. Moreover, its resilience under diverse benign noise distributions and model architectures highlights strong generalization and practicality for real-world deployment. Overall, these results establish NoiFi as advancing the state of the art in noise-based defenses, paving the way for trustworthy and secure WiFi sensing in safety- and privacy-critical applications

Acknowledgments

This work was supported in part by the Office of Naval Research under grants N00014-24-1-2730, the National Science Foundation under grants 2235232, 2312447, 2247560, 2433904, 2312794, and 2509636; and a fellowship from the Amazon–Virginia Tech Initiative for Efficient and Robust Machine Learning.

References

[1] Harshit Ambalkar, Tianya Zhao, Xuyu Wang, and Shiwen Mao. 2023. Adversarial attack and defense for WiFi-based apnea detection system. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 1–2.

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*. Springer, 484–501.

[3] Fabrizio Angiulli and Clara Pizzuti. 2002. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*. Springer, 15–27.

[4] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728* (2018).

[5] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. 2020. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*. IEEE, 1277–1294.

[6] Bowonkoon Chitprasert and KR Rao. 1990. Discrete cosine transform filtering. *Signal processing* 19, 3 (1990), 233–245.

[7] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. 2017. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410* (2017).

[8] Karl Pearson F.R.S. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572. doi:10.1080/14786440109462720

[9] Rafael C Gonzales and Richard E Woods. 2002. Digital image processing, 2-nd edition. *Prentice Hall* (2002), 2278.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

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

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[13] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261* (2019).

[14] Hoki Kim. 2020. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950* (2020).

[15] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[16] Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. 2007. Outlier detection with kernel density functions. In *International workshop on machine learning and data mining in pattern recognition*. Springer, 61–75.

[17] Changming Li, Mingjing Xu, Yicong Du, Limin Liu, Cong Shi, Yan Wang, Hongbo Liu, and Yingying Chen. 2024. Practical adversarial attack on wifi sensing through unnoticeable communication packet perturbation. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 373–387.

[18] Junye Li, Deepak Mishra, Dilip Krishnaswamy, Ayon Chakraborty, Joseph G Davis, and Aruna Seneviratne. 2022. WiFi Interference-Based Adversarial Attacks on NTC Using CSI Sensing. In *ICC 2022-IEEE International Conference on Communications*. IEEE, 4354–4359.

[19] Jianwei Liu, Yinghui He, Chaowei Xiao, Jinsong Han, Le Cheng, and Kui Ren. 2022. Physical-world attack towards WiFi-based behavior recognition. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 400–409.

[20] Li Lu, Meng Chen, Jiadi Yu, Zhongjie Ba, Feng Lin, Jinsong Han, Yanmin Zhu, and Kui Ren. 2024. An imperceptible eavesdropping attack on wifi sensing systems. *IEEE/ACM Transactions on Networking* (2024).

[21] Shiqing Ma and Yingqi Liu. 2019. Nic: Detecting adversarial samples with neural network invariant checking. In *Proceedings of the 26th network and distributed system security symposium (NDSS 2019)*.

[22] Yongsen Ma, Gang Zhou, and Shuangquan Wang. 2019. WiFi sensing with channel state information: A survey. *ACM Computing Surveys (CSUR)* 52, 3 (2019), 1–36.

[23] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[25] Dongyu Meng and Hao Chen. 2017. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 135–147.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1765–1773.

[27] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. 2018. Adversarial Robustness Toolbox v1.2.0. *CoRR* 1807.01069 (2018). https://arxiv.org/pdf/1807.01069

[28] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 372–387.

[29] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. 2019. Adversarial robustness through local linearization. *Advances in neural information processing systems* 32 (2019).

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

[31] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13, 7 (2001), 1443–1471.

[32] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. 2012. You are facing the Mona Lisa: Spot localization using PHY layer information. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. 183–196.

[33] Md Hasan Shahriar, Ning Wang, Naren Ramakrishnan, Y Thomas Hou, and Wenjing Lou. 2024. Let the Noise Speak: Harnessing Noise for a Unified Defense Against Adversarial and Backdoor Attacks. *arXiv e-prints* (2024), arXiv–2406.

[34] Yamini Shankar and Ayon Chakraborty. 2024. Practical Defense Against Adversarial WiFi Sensing. In *2024 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 1–6.

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[36] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[37] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. 1096–1103.

[38] Meng Wang, Jinyang Huang, Xiang Zhang, Zhi Liu, Meng Li, Peng Zhao, Huan Yan, Xiao Sun, and Mianxiong Dong. 2024. Target-Oriented WiFi Sensing for Respiratory Healthcare: from Indiscriminate Perception to In-Area Sensing. *IEEE Network* (2024).

[39] Ning Wang, Yimin Chen, Yang Xiao, Yang Hu, Wenjing Lou, and Y Thomas Hou. 2022. Manda: On adversarial example detection for network intrusion detection system. *IEEE Transactions on Dependable and Secure Computing* 20, 2 (2022), 1139–1153.

[40] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st annual international conference on mobile computing and networking*. 65–76.

[41] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2017. Device-free human activity recognition using commercial WiFi devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1118–1131.

[42] Yuxi Wang, Kaishun Wu, and Lionel M Ni. 2016. Wifall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing* 16, 2 (2016), 581–594.

[43] Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209* (2018).

[44] Zhongcheng Wei, Wei Chen, Shuli Ning, Weidong Lin, Nan Li, Bin Lian, Xiang Sun, and Jijun Zhao. 2025. A Survey on WiFi-based Human Identification: Scenarios, Challenges, and Current Solutions. *ACM Transactions on Sensor Networks* 21, 1 (2025), 1–32.

[45] Leiyang Xu, Xiaolong Zheng, Xiangyuan Li, Yucheng Zhang, Liang Liu, and Huadong Ma. 2022. WiCAM: Imperceptible adversarial attack on deep learning based WiFi sensing. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 10–18.

[46] Weilin Xu, David Evans, and Yanjun Qi. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155* (2017).

[47] Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society.
[48] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. 2023. MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data* 10, 1 (2023), 41.
[49] Guolin Yin, Junqing Zhang, Xinping Yi, and Xuyu Wang. 2025. Evasion Attacks and Countermeasures in Deep Learning-Based Wi-Fi Gesture Recognition. *IEEE Transactions on Mobile Computing* (2025).
[50] Siamak Yousefi, Hirokazu Narui, Sankalp Dayal, Stefano Ermon, and Shahrokh Valaee. 2017. A survey on behavior recognition using WiFi channel state information. *IEEE Communications Magazine* 55, 10 (2017), 98–104.
[51] Rojin Zandi, Kian Behzad, Elaheh Motamedi, Hojjat Salehinejad, and Milad Siami. 2024. Robofisense: Attention-based robotic arm activity recognition with wifi sensing. *IEEE Journal of Selected Topics in Signal Processing* (2024).
[52] Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. Pyod: A python toolbox for scalable outlier detection. *Journal of machine learning research* 20, 96 (2019), 1–7.
[53] Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S Yu. 2022. Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *Comput. Surveys* 55, 8 (2022), 1–39.
[54] Siwang Zhou, Wei Zhang, Dan Peng, Yonghe Liu, Xingwei Liao, and Hongbo Jiang. 2019. Adversarial WiFi sensing for privacy preservation of human behaviors. *IEEE Communications Letters* 24, 2 (2019), 259–263.
[55] Yuxuan Zhou, Huangxun Chen, Chenyu Huang, and Qian Zhang. 2022. WiADv: Practical and robust adversarial attack against WiFi-based gesture recognition system. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–25.

A Common Adversarial Attacks

Here, we provide the details of each attack generation method, which include:

Fast Gradient Sign Method (FGSM), introduced by Goodfellow et al. [10], perturbs the input in a single step by adding a scaled sign of the gradient of the loss with respect to the input. Given a model parameterized by θ, an input x with true label y, and a loss function $J(\theta, x, y)$, the adversarial example x_{adv} is computed as:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}\left(\nabla_x J(\theta, x, y)\right) \tag{9}$$

where ϵ controls the magnitude of the perturbation.

Basic Iterative Method (BIM), proposed by Kurakin et al. [15], extends FGSM by applying it iteratively. At each step, a small perturbation is added in the direction of the gradient, followed by clipping to ensure the result remains within an ϵ-ball around the original input. Formally, for step size α, and clip function $\text{Clip}_{x,\epsilon}$, the update rule is:

$$\begin{aligned} &x_{\text{adv}}(0) = x, \\ &x_{\text{adv}}(t+1) = \text{Clip}_{x,\epsilon}\left(x_{\text{adv}}(t) + \alpha \cdot \text{sign}\left(\nabla_x J(\theta, x_{\text{adv}}(t), y)\right)\right) \end{aligned} \tag{10}$$

Projected Gradient Descent (PGD), presented by Madry et al. [24], is a widely adopted iterative adversarial attack that generalizes BIM by incorporating projection onto the feasible perturbation space. At each iteration, the perturbation is updated via gradient ascent and then projected back to the ϵ-ball around the original input:

$$\begin{aligned} &x_{\text{adv}}(0) = x, \\ &x_{\text{adv}}(t+1) = \text{Proj}_{x,\epsilon}\left(x_{\text{adv}}(t) + \alpha \cdot \text{sign}\left(\nabla_x J(\theta, x_{\text{adv}}(t), y)\right)\right) \end{aligned} \tag{11}$$

Universal Adversarial Perturbation (UAP), introduced by Moosavi-Dezfooli et al. [26], constructs a single, input-agnostic perturbation δ that can be added to any input x^i to induce misclassification. This perturbation is computed offline by aggregating gradients across a subset of the training data. The optimization objective is to maximize the expected model error over this dataset:

$$\delta = \arg\max_{\delta} \sum_{(x^i, y^i) \in \mathcal{D}} \|f(x^i + \delta) - y^i\|_2^2 \tag{12}$$

subject to $\|\delta\| \leq \epsilon$, where $f(\cdot)$ denotes the model's output and $\|\cdot\|$ is typically an ℓ_p norm.

B NoiFi Algorithm

Algorithm 1 shows the high-level steps of NoiFi in detecting adversarial attacks. For clarity and reproducibility, we explicitly separate the training phase (threshold estimation on benign validation data) from the testing phase (per-sample anomaly scoring and decision).

Algorithm 1 NoiFi: Adversarial Input Detection

Function NoiFiInference($x, \mathcal{A}, \mathcal{F}_\theta, N$)
  $\hat{x} \leftarrow \mathcal{A}(x)$ // Reconstruct input via denoising autoencoder
  $\tau \leftarrow \mathcal{F}_\theta(\eta)$ //Feature of extracted noise
  $\tau \leftarrow \mathcal{F}(\eta)$ // Compute feature representation of noise
  Initiate $\mathcal{N}^{\text{rand}} = [\,]$ // Placeholder for randomized noise
  for $i = 1$ to N **do** // Create multiple instance
    $\mathcal{N}^{\text{rand}}[i] \leftarrow \text{Randomize}(\eta)$ //Randomize noise
  end for
  $\mathcal{T}^{\text{rand}} \leftarrow \mathcal{F}_\theta(\mathcal{N}^{\text{rand}})$ // Feature of randomized noises
  $\alpha \leftarrow OD(\tau, \mathcal{T}^{\text{rand}})$ // Compute anomaly score
  return α // Return anomaly score
/*==========*/
/* — Training Phase — */
Train autoencoder $\mathcal{A}$ with benign validation dataset $\mathcal{D}_{\text{val}}$
for all benign inputs x_i in $\mathcal{D}_{\text{val}}$ **do**
  $\alpha_{\text{train}}[i] \leftarrow$ NoiFiInference($x_i, \mathcal{A}, \mathcal{F}_\theta, N$)
end for
Estimate threshold $\alpha_{\text{th}} \leftarrow \text{Quantile}_{(p)}(\alpha_{\text{train}})$
/*==========*/
/* — Testing Phase: — */
$\alpha_{\text{test}} \leftarrow$ NoiFiInference($x_{\text{test}}, \mathcal{A}, \mathcal{F}_\theta, N$)
if $\alpha_{\text{test}} < \alpha_{\text{th}}$ **then**
  return benign
else
  return adversarial
end if

C Proof of Propositions in NRM

This section provides proof sketches for the two propositions in Section 3.3. We use the Information Bottleneck principle to explain why benign noises map to stable, low-variance feature representations, while adversarial noises preserve task-relevant structure and thus deviate from the randomized-noise manifold.

C.1 Representation of Benign Inputs

Here is the proof of Proposition 3.

Proof. For this proof, we analyze the behavior of $\mathcal{F}_\theta$ through the lens of the Information Bottleneck (IB) principle [36], which characterizes the trade-off between compression and relevance in learned representations. Let $X \in \mathcal{X}$ denote the input random variable, $Y \in \mathcal{Y}$ the corresponding label, and $Z = \mathcal{F}_\theta(X)$ the intermediate representation used for prediction $\mathcal{P}_\theta(Z)$. The IB framework

seeks to extract a representation Z that retains maximal information about Y while discarding irrelevant aspects of X, formalized by the minimization of:

$$\mathcal{L}_{\text{IB}} = I(X;Z) - \beta I(Z;Y), \tag{13}$$

where $I(\cdot;\cdot)$ denotes mutual information and $\beta > 0$ controls the balance between compression ($I(X;Z)$) and predictive sufficiency ($I(Z;Y)$). For benign inputs, the reconstruction noise η_{ben} corresponds to directions in input space that are uninformative with respect to Y. Accordingly, $I(\eta_{\text{ben}};Y) \approx 0$, and the IB objective in Eq. (13) dictates that $\mathcal{F}_\theta$ should minimize $I(\eta_{\text{ben}};\tau_{\text{ben}})$. This incentivizes $\mathcal{F}_\theta$ to map η_{ben} to compressed, low-variance representations that do not encode task-relevant structure. Due to the continuity of $\mathcal{F}_\theta$ and the randomized construction of $\mathcal{N}_{\text{ben}}^{\text{rand}}$, all such variants similarly yield feature embeddings that are low in magnitude and tightly clustered. That is, the distribution $\mathcal{T}_{\text{ben}}^{\text{rand}}$ concentrates around a low-dimensional manifold, and the original embedding τ_{ben} lies within this manifold. This reflects the invariance encouraged by the IB principle when the input provides no information about the target. □

C.2 Representation of Adversarial Inputs

Here is the proof of Proposition 4.

Proof. The reconstruction noise η_{adv} is derived from an adversarial input intentionally optimized to induce erroneous classification. By construction, such perturbations are entangled with the target prediction: $I(\eta_{\text{adv}};y) > 0$. Consequently, under the IB principle, $\mathcal{F}_\theta$ must preserve the predictive structure of η_{adv} to induce misclassification, which implies that $I(\eta_{\text{adv}};\tau_{\text{adv}})$ must be high. This leads to $\tau_{\text{adv}} = \mathcal{F}_\theta(\eta_{\text{adv}})$ exhibiting input-specific, high-magnitude features. However, randomized variants $\mathcal{N}_{\text{adv}}^{\text{rand}}$ are drawn from a process that disrupts this adversarial structure. The removal of predictive information from these inputs effectively sets $I(\eta_{\text{rand}};y) \approx 0$ for $\eta_{\text{rand}} \in \mathcal{N}_{\text{adv}}^{\text{rand}}$. Based on the IB objective once again: $\mathcal{F}_\theta$ maps these inputs to low-magnitude, invariant features, resulting in a compact distribution $\mathcal{T}_{\text{adv}}^{\text{rand}}$. Thus, due to the preservation of adversarial signal in η_{adv} and its absence in $\mathcal{N}_{\text{adv}}^{\text{rand}}$, the feature τ_{adv} lies significantly outside the support of $\mathcal{T}_{\text{adv}}^{\text{rand}}$. This mismatch reflects the IB trade-off: adversarial inputs retain task-relevant perturbations that resist compression, unlike their randomized counterparts. □

D Additional Outlier Detection Methods in NDM

While NoiFi uses the mean Euclidean distance (MED) as the default outlier score in Section 3.4, other choices may be preferable depending on the feature distribution, sample size N, and runtime constraints. Below, we summarize alternative outlier detection (OD) scoring functions that operate on the same feature representations ($\tau_{\text{test}}, \mathcal{T}_{\text{test}}^{\text{rand}}$), along with their key intuition and trade-offs.

K-Nearest Neighbor (KNN)-based Deviation. Instead of estimating a full density, KNN relies on geometric proximity in feature space to quantify abnormality. The core idea is that anomalies tend to lie farther away from dense clusters of reference points. For a test task τ_{test}, the anomaly score α_{test} is derived from its distances to the k-nearest neighbors in $\mathcal{T}_{\text{test}}^{\text{rand}}$. A common formulation uses either the maximum or the average of these distances:

$$\alpha_{\text{test}} = OD_{\text{KNN}}(\tau_{\text{test}}) = \frac{1}{k}\sum_{j=1}^{k} \|\tau_{\text{test}} - \tau_{(j)}\|_2,$$

where $\tau_{(j)}$ denotes the j-th nearest neighbor of τ_{test}. This non-parametric approach is particularly advantageous when the data geometry is irregular or fragmented, as it requires no explicit distributional assumptions. However, its sensitivity to local density variations and computational overhead at large scales can be limiting factors.

Kernel Density Estimation (KDE)-based Deviation. When the distribution $\mathcal{T}_{\text{test}}^{\text{rand}}$ deviates from Gaussian assumptions or exhibits more complex, possibly multimodal structures, a non-parametric KDE can offer greater flexibility. KDE does not assume any parametric form and can capture intricate distributional shapes by leveraging kernel functions that locally smooth the data. Assuming $\mathcal{T}_{\text{test}}^{\text{rand}}$ is smooth and supported in a relatively low-dimensional space, the anomaly score α_{test} is computed via the negative log-likelihood under the estimated density:

$$\alpha_{\text{test}} = OD_{\text{KDE}}(\tau_{\text{test}}) = -\log\left(\frac{1}{Nh^F}\sum_{i=1}^{N} K\left(\frac{\tau_{\text{test}} - \tau_i}{h}\right)\right),$$

where $K(\cdot)$ is a kernel function (commonly Gaussian), h is the bandwidth parameter, and $\{\tau_i\}_{i=1}^N \sim \mathcal{T}_{\text{test}}^{\text{rand}}$ are reference tasks. KDE is particularly suitable when no strong parametric form can be assumed and the data exhibits locally smooth variation.

One-Class SVM (OCSVM)-based Deviation. One-Class Support Vector Machines (OCSVMs) detect anomalies by estimating the boundary of the high-density region of a reference distribution. Given a set of training representations $\{\tau_i\}_{i=1}^N \sim \mathcal{T}_{\text{test}}^{\text{rand}}$, the OCSVM seeks a hyperplane in feature space that best separates the training data from the origin, under the assumption that most data lies close to a common mode, and anomalies are dispersed.

The linear OCSVM solves the following convex optimization problem:

$$\begin{aligned} \min_{w,\rho,\xi} \quad & \frac{1}{2}\|w\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \\ \text{subject to} \quad & w^\top \tau_i \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1,\ldots,N, \end{aligned}$$

where $w \in \mathbb{R}^d$ defines the normal vector of the separating hyperplane, ρ is the offset, and ξ_i are slack variables that allow some points to lie within the margin or be considered outliers. The parameter $\nu \in (0,1]$ controls the trade-off between model simplicity (i.e., a larger margin) and the number of allowed violations.

After training, the learned hyperplane defines a decision function

$$OD_{\text{OCSVM}}(\tau_{\text{test}}) = w^\top \tau_{\text{test}} - \rho,$$

which serves as the anomaly score for a test point τ_{test}. A negative score indicates that the point lies outside the estimated support of the distribution (i.e., is anomalous), while a positive score suggests inclusion within the high-density region.

The linear kernel is particularly appropriate when the ambient dimension d is much larger than the number of training samples N, a setting where even simple hyperplanes can effectively separate data due to the geometry of high-dimensional spaces. Unlike

kernel-based OCSVM variants, the linear model avoids costly kernel computations and overfitting risks, making it a scalable and robust choice in high-dimensional, low-sample regimes common in adversarial and robust ML tasks.

Table 3: Architecture of the Classification Models.

Dataset	Model	Input	Conv Channels	Feat	Out
UT-HAR	Target	90×500	[32,64,64,128,128,256,256]	512	7
	Surrogate	90×500	[16,16,32,32,64,64,128]	512	7
RoboFiSense	Target	360×360	[32,64,64,128,128,256,256]	512	8
	Surrogate	360×360	[16,16,32,32,64,64,128]	512	8

E Architecture of the Classification Models

Table 3 shows the architecture of CNN-based classifier models considered in evaluation.

F Hyper-parameters of Adaptive Attacks

Table 4 summarizes all parameters used in the evaluation of the adaptive attack. Parameters ϵ and w are varied to assess sensitivity, and other values are used for optimization and reporting. Here, ϵ controls the L_∞ perturbation budget: larger ϵ permits more intrusive changes to the CSI input and typically increases misclassification success, but overly large budgets blur whether misclassification stems from adversarial structure or simply excessively strong noise. We therefore sweep $\epsilon \in [0.05, 0.15]$, where $\epsilon = 0.05$ yields very subtle perturbations (often lower ASR) and $\epsilon = 0.15$ yields higher ASR but more noticeable input distortion. On the other hand, w controls the strength of the stealth term in the adaptive objective: $w = 0$ prioritizes purely malicious perturbations (stealth arises only indirectly via small ϵ), whereas larger w forces the optimizer to produce more random-noise-like perturbations at the cost of maliciousness/ASR. As a result, small ϵ combined with large w tends to produce the stealthiest adaptive attacks, but may substantially reduce ASR as shown in Fig. 10.

G Visualization of Adversarial Attack

Table 5 visualizes the adversarial perturbation and adversarial samples generated with different attack algorithms for two datasets. This qualitative view complements the quantitative metrics by showing how different attacks manifest in the input space and how perturbations differ across datasets and threat settings.

H t-SNE Visualization of Raw Noise vs Noise Features

Fig. 11, 12, 13, and 14 show the t-SNE visualization of raw sampled noise vs feature representation of such raw noise. As shown in the figures, once the raw noises are passed through the feature extractor, the contrast between the benign and adversarial noise becomes obvious, showing the effectiveness of NoiFi. In particular, the feature-space plots highlight improved separability compared to raw noise, supporting the design choice of using the target model as a feature extractor in NRM.

I Results on Supplementary Experiment

This section reports supplementary results and ablations that support the main evaluation but are omitted from the main text due to space constraints. We focus on how key hyperparameters and design choices (e.g., the number of randomized samples N and OD choice) affect detection performance.

I.1 Evaluation on the Number of Randomized Noises and OD Methods

Here, we investigate the role of randomized noise size N in the construction of randomized feature representations for adversarial detection. For this evaluation, we also evaluate four OD methods—OCSVM, KDE, MED, and KNN—across multiple attack types for the *UT-HAR* dataset. Fig. 15 illustrates the impact of N on adversarial detection performance (AUROC). Among the attacks, FGSM exhibits slightly lower detection performance compared to BIM, PGD, and UAP, with AUROC peaking around an N of 5. In contrast, the latter three attacks are consistently easy to detect across all OD methods, with near-perfect AUROC regardless of sampling size.

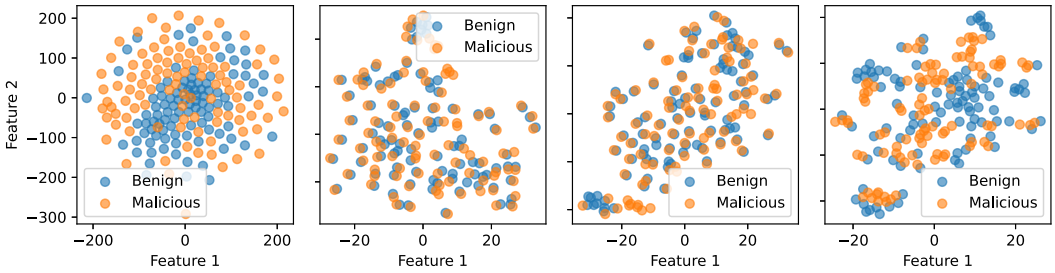
Across detectors, OCSVM demonstrates the higher sensitivity to sampling variation, particularly under BIM, whereas the rest maintain stable and robust performance. These results indicate that while most adversarial perturbations yield strong and stable detection, weaker attacks such as FGSM can expose vulnerabilities in sampling-sensitive detectors, underscoring the importance of robust method and sample size selection for reliable adversarial detection. Across all the OD methods, MED demonstrates the most consistent generalization, underscoring their suitability for practical deployment of noise-driven, randomized defenses.

I.2 Evaluation on ResNet Architecture with *RoboFiSense* Dataset

To test whether NoiFi is tied to a specific classifier architecture, we also integrate it with a ResNet-based backbone and report detection performance under the same threat model. Figure 16 illustrates the detection performance of NoiFi when deployed with a ResNet-based classifier on the *RoboFiSense* dataset under a white-box threat model. The ROC and PR curves demonstrate that NoiFi maintains high robustness across multiple adversarial attacks, achieving perfect AUROC and AUPRC scores against FGSM and UAP, while also performing strongly against more advanced iterative attacks such as BIM (AUROC = 0.929, AUPRC = 0.947) and PGD (AUROC = 0.947, AUPRC = 0.956). These results confirm that the effectiveness of NoiFi is not tied to a specific classifier architecture. In particular, even with ResNet, which differs significantly from the default backbone used in earlier evaluations, the system consistently detects adversarial perturbations with high fidelity. This highlights the adaptability and generalization capability of NoiFi, reinforcing its practicality for deployment across diverse model architectures.

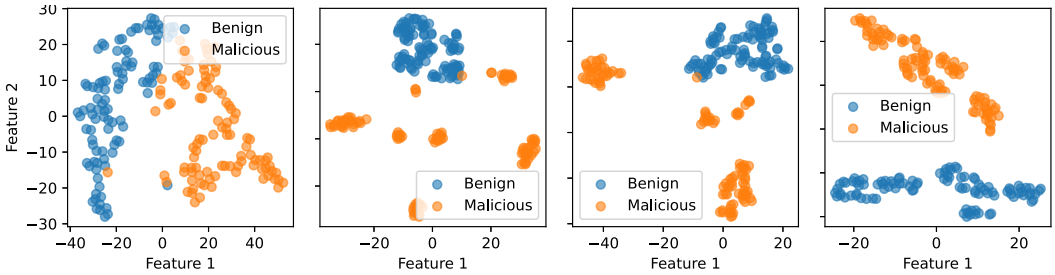
I.3 Comparison of Different OD Methods

Beyond AUROC/AUPRC, practical deployments often require fixing a decision threshold to control the false positive rate. In this subsection, we compare OD methods under such a fixed-threshold setting to highlight precision–recall trade-offs. We further evaluate



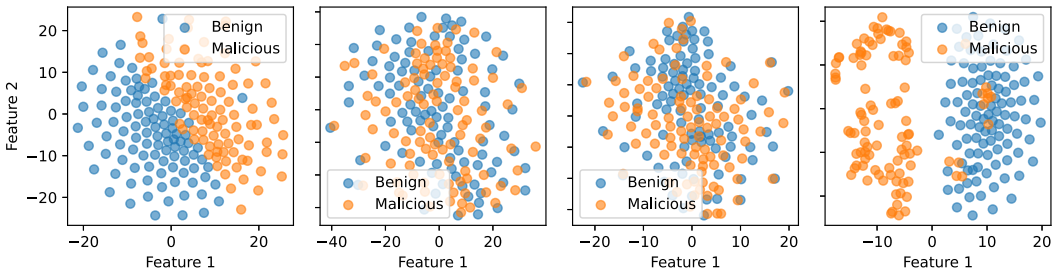
(a) FGSM (b) BIM (c) PGD (d) UAP

Figure 11: t-SNE representation of reconstructed raw noise under white-box attacks on *UT-HAR* dataset



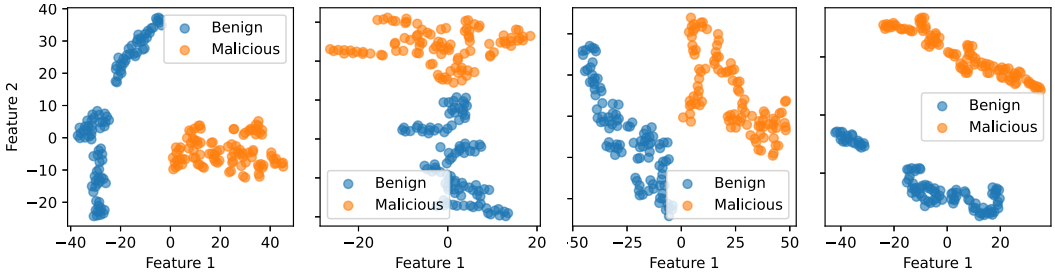
(a) FGSM (b) BIM (c) PGD (d) UAP

Figure 12: t-SNE representation of reconstructed noise features under white-box attacks on *UT-HAR* dataset



(a) FGSM (b) BIM (c) PGD (d) UAP

Figure 13: t-SNE representation of reconstructed raw noise under white-box attacks on *RoboFiSense* dataset



(a) FGSM (b) BIM (c) PGD (d) UAP

Figure 14: t-SNE representation of reconstructed noise features under white-box attacks on *RoboFiSense* dataset

Table 4: Parameters used in the evaluation of the adaptive attack.

Symbol	Description	Value(s) in evaluation
ϵ	L_∞ perturbation budget	0.05, 0.075, 0.10, 0.125, 0.15
w	Stealth weight (trade-off)	0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2
S	EOT Monte Carlo samples	10
-	Number of attack candidates (worst-case ASR)	10
α	Step size (gradient ascent)	0.025
-	Number of PGD iterations	100
$\lvert\mathcal{T}_{\text{adv}}^{\text{rand}}\rvert$	Random refs. per EOT for stealth term	10

Table 5: Adversarial examples across attacks and datasets.

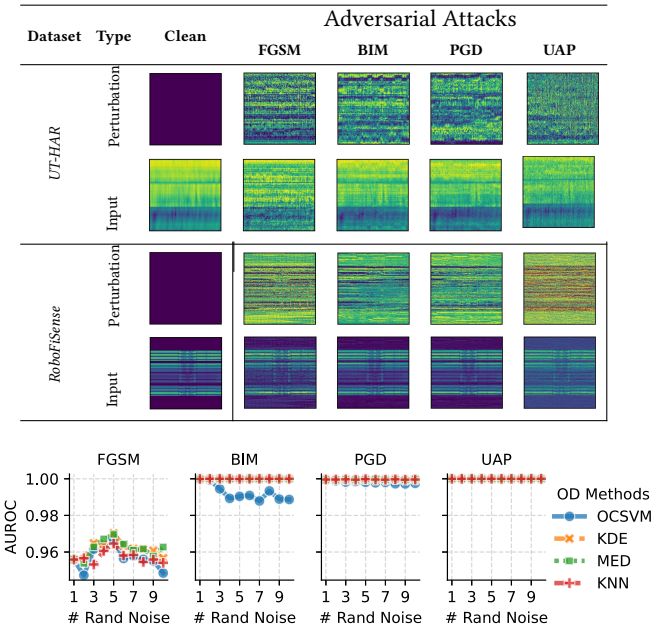
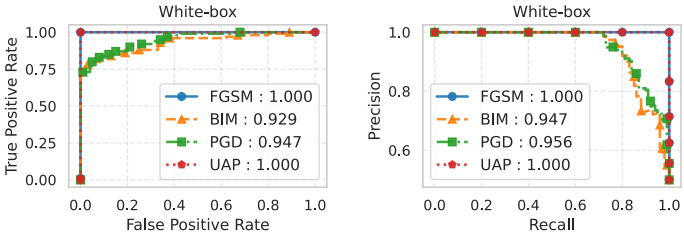


Figure 15: Effect of the number of randomized noise size N on adversarial attack detection for different OD methods for the *UT-HAR* dataset in the white-box setting.



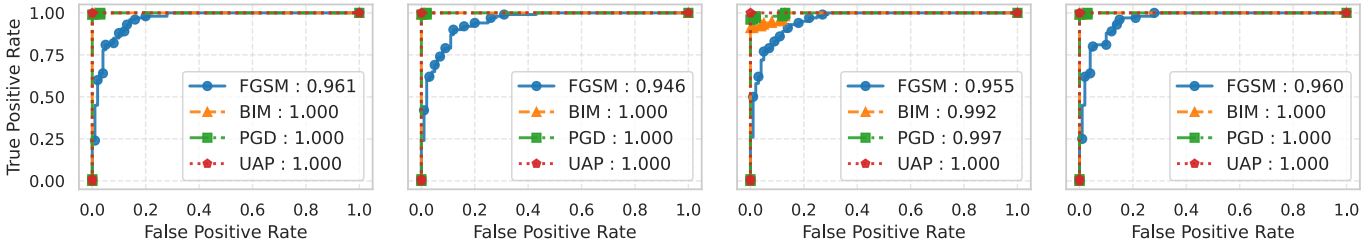
(a) ROC Curve with AUROC Scores. **(b) PR Curve with AUPRC Scores**

Figure 16: Detection performance of NoiFi when integrated with a ResNet-based classifier on the *RoboFiSense* dataset under white-box adversarial attacks.

the detailed detection performance of NoiFi under a fixed decision threshold designed to ensure a low FPR. For this evaluation, the threshold is chosen such that the FPR remains below 1%, which aligns with practical deployment. Under this stringent evaluation criterion, Table 6 reports precision, recall, and F1-score, along with AUROC scores, for three OD detection methods under similar evaluation settings. Under the white-box attack setting, performance on the *UT-HAR* dataset remains strong across most attacks. Notably, KDE and MED consistently outperform OCSVM detectors Overall, these results demonstrate that NoiFi, when paired with an effective OD detector such as MED, achieves consistently high detection performance under stringent low-FPR constraints, even in transfer-based black-box scenarios—underscoring its potential for deployment in practical systems. Fig. 17, 18, 19, and 20 show the ROC curves for different datasets and threat models.

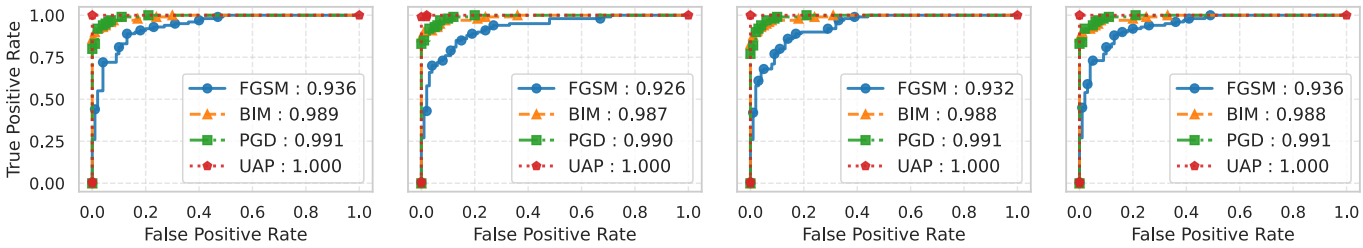
Table 6: Performance metrics per Attack, Threat Model, and Dataset for different OD Methods.

Dataset	Threat	Detector	FGSM					BIM					PGD					UAP				
			AUROC	AUPRC	Pre	Rec	F1	AUROC	AUPRC	Pre	Rec	F1	AUROC	AUPRC	Pre	Rec	F1	AUROC	AUPRC	Pre	Rec	F1
UT-HAR	White-box	KDE	**0.961**	**0.954**	**0.961**	0.450	0.616	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**1.000**	**1.000**	**1.000**	**0.990**	**0.995**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		OCSVM	0.955	0.949	0.955	**0.500**	**0.662**	0.992	0.993	0.992	0.920	0.953	0.997	0.997	0.997	0.970	0.980	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		MED	0.960	0.953	0.960	0.450	0.616	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**1.000**	**1.000**	**1.000**	**0.990**	**0.995**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Black-box	KDE	0.936	0.934	0.936	0.440	0.607	**0.989**	**0.990**	**0.989**	**0.900**	**0.942**	0.991	0.991	0.991	0.830	0.902	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		OCSVM	0.932	0.930	0.932	0.420	0.587	0.988	0.989	0.988	0.880	0.931	0.991	0.991	0.991	0.820	0.896	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		MED	**0.936**	**0.935**	**0.936**	**0.450**	**0.616**	0.988	0.989	0.988	**0.900**	**0.942**	**0.991**	**0.992**	**0.991**	**0.840**	**0.908**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
UT-HAR	White-box	KDE	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		OCSVM	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		MED	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Black-box	KDE	1.000	1.000	1.000	**1.000**	0.995	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**0.999**	**0.999**	**0.999**	**0.970**	**0.985**	**0.993**	**0.993**	**0.993**	**0.930**	0.959
		OCSVM	1.000	1.000	1.000	**1.000**	0.995	0.999	0.999	0.999	0.970	0.980	0.999	0.999	0.999	**0.970**	0.980	0.991	0.992	0.991	**0.930**	**0.964**
		MED	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**0.999**	**0.999**	**0.999**	**0.970**	**0.985**	0.993	0.993	0.993	**0.930**	0.959



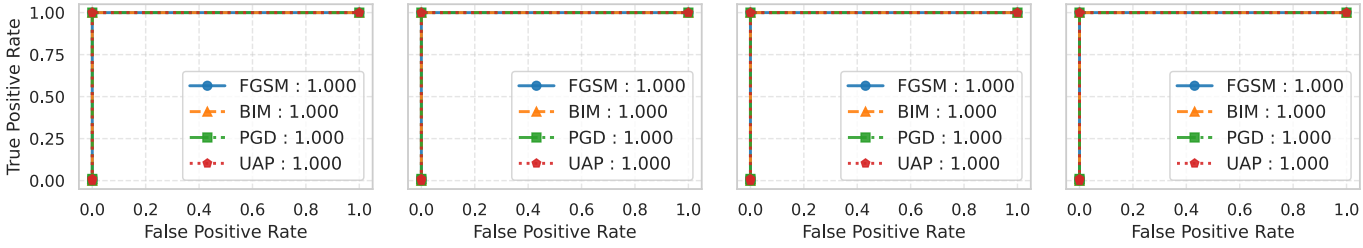
(a) KDE **(b) KNN** **(c) OCSVM** **(d) MED**

Figure 17: ROC curves with AUROC scores of different OD methods under White-box Attacks on *UT-HAR* Dataset



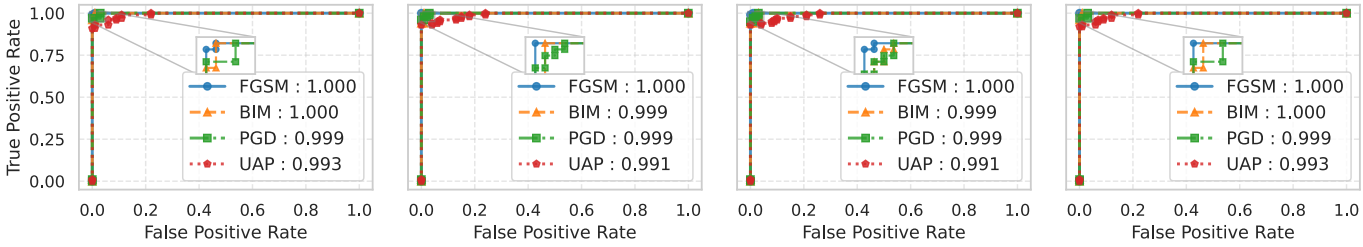
(a) KDE **(b) KNN** **(c) OCSVM** **(d) MED**

Figure 18: ROC curves with AUROC scores of different OD methods under Black-box Attacks on *UT-HAR* Dataset



(a) KDE **(b) KNN** **(c) OCSVM** **(d) MED**

Figure 19: ROC curves with AUROC scores of different OD methods under White-box Attacks on *RoboFiSense* Dataset



(a) KDE (b) KNN (c) OCSVM (d) MED

Figure 20: ROC curves with AUROC scores of different OD methods under Black-box Attacks on *RoboFiSense* Dataset