

Article

DL-Based Physical Tamper Attack Detection in OFDM Systems with Multiple Receiver Antennas: A Performance–Complexity Trade-Off

Eshagh Dehmollaian ^{1,*} , Bernhard Etzlinger ² , Núria Ballber Torres ²  and Andreas Springer ¹ ¹ JKU LIT SAL eSPML Lab, Institute for Communications Engineering and RF-Systems, Johannes Kepler University, 4040 Linz, Austria² Institute for Communications Engineering and RF-Systems, Johannes Kepler University, 4040 Linz, Austria

* Correspondence: eshagh.dehmollaian@jku.at

Abstract: This paper proposes two deep-learning (DL)-based approaches to a physical tamper attack detection problem in orthogonal frequency division multiplexing (OFDM) systems with multiple receiver antennas based on channel state information (CSI) estimates. The physical tamper attack is considered as the unwanted change of antenna orientation at the transmitter or receiver. Approaching the tamper attack scenario as a semi-supervised anomaly detection problem, the algorithms are trained solely based on tamper-attack-free measurements, while operating in general scenarios that may include physical tamper attacks. Two major challenges in the algorithm design are environmental changes, e.g., moving persons, that are not due to an attack and evaluating the trade-off between detection performance and complexity. Our experimental results from two different environments, comprising an office and a hall, show the proper detection performances of the proposed methods with different complexity levels. The optimal proposed method achieves a 93.32% true positive rate and a 10% false positive rate with a suitable level of complexity.

Keywords: physical tamper attack; OFDM; CSI; deep learning; anomaly detection



Citation: Dehmollaian, E.; Etzlinger, B.; Torres, N.B.; Springer, A. DL-Based Physical Tamper Attack Detection in OFDM Systems with Multiple Receiver Antennas: A Performance–Complexity Trade-Off. *Sensors* **2022**, *22*, 6547. <https://doi.org/10.3390/s22176547>

Academic Editor: Giovanni Andrea Casula

Received: 27 July 2022

Accepted: 24 August 2022

Published: 30 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wireless networks in critical infrastructures require a high level of security. Therefore, different network security threats need to be considered in such applications. Among them, physical tampering with a device is one that is missing in many applications. As discussed by [1], a possible physical tamper attack is the altering of the orientation of surveillance cameras that monitor a critical infrastructure. Radio-frequency (RF) fingerprint-based localization systems using physical (PHY) layer measurements (e.g., received signal strength indication (RSSI) [2], channel impulse response (CIR) [3], channel state information (CSI) [4], etc.) comprise another example in which physical tampering can significantly distort the system function (e.g., by changing antenna characteristics). According to the European Union Agency for Cybersecurity (ENISA) [5], physical tamper attacks within IoT applications are one of the main threats faced by healthcare organizations as well. An assumption in all these cases is that the transceivers must not be tampered for the system to work correctly. Thus, the functionality of the systems is destroyed with high probability, if the transceivers are tampered. In order to recognize such attacks, a physical tamper attack detection mechanism is required.

To address this issue, radio channel characteristics, which are observed by measurements [1,3,6,7], can help us to detect the tamper attack. Such measurements are the CIR [3], CSI [1,6,7], and received packet features [8,9]. However, the characteristics are not solely influenced by the tamper attack, but also by regular environmental changes. As one of the first works, the proposed CIR method in [3] was only tested in environments with

few dynamic elements and, thus, experienced high misdetection rates in dynamic environments. Reference [7] also investigated the feasibility of using a commercial off-the-shelf (COTS) Wi-Fi device as the physical tamper detector based on CSI collection in an almost static environment. To achieve resilience to regular environmental changes, Reference [1] proposed to increase space diversity by using multiple antennas at the receiver. However, others proposed to make use of machine learning (ML) approaches to tackle this issue. Reference [6] showed that a semi-supervised deep learning (DL) algorithm with a postprocessing unit can extract the characteristics of environments and outperforms the approach by [1]. References [8,9] also showed that their ML approaches for the detection of removal/addition of sensors within IoT applications perform with high accuracy in a dynamic environment. The summary of the aforementioned works can be seen in Table 1.

Table 1. A comparison of the relevant literature.

Reference-Year	System	Data	Environment	Remark
[3]-2007	DSSS	CIR	few dynamic elements	high misdetection rate in dynamic environments
[1]-2015	802.11n Wi-Fi	CSI	dynamic	using multiple receivers
[8]-2018	IoT	sensor data	dynamic	supervised ML algorithm
[6]-2021	OFDM-based	CSI	dynamic	semi-supervised DL algorithm
[7]-2021	COTS Wi-Fi	CSI	static	using a COTS Wi-Fi device as a detector
[9]-2021	IoT	packet length	dynamic	unsupervised ML algorithm

DSSS: direct sequence spread spectrum; in this table, the term “IoT” is used to indicate that the corresponding works utilized data from different networks such as WiFi, Zigbee, and Bluetooth.

To perform physical tamper attack detection, three main strategies have been applied in the literature. These are: (i) distance computation between previous measurements and new measurements either with hypothesis testing for CIR values in [3] or with direct threshold detection for the CSI in [1]; (ii) distance computation to a lower-dimensional signal representation obtained in a DL framework in [6]; and (iii) a direct detection using the ML algorithms in [7–9].

The most recent methods, namely (ii) and (iii), outperform previous methods in terms of the attack detection accuracy. However, it is a cumbersome task to directly compare the aforementioned methods due to their different communication systems. Therefore, since currently, orthogonal frequency division multiplexing (OFDM) is one of the most common transmission technologies [10], we followed [1,6,7] and based our proposed physical tamper attack detection methods on the estimated CSI in an OFDM-based wireless system.

To detect physical tamper attacks in an OFDM-based system, Reference [1] proposed to use multiple antennas at the receiver and calculate the distances between tamper-free CSI in the offline and online phases. This approach was based on the assumption that environmental variations will not affect all CSI received from different antennas at the receiver. In [6], a mixed DL approach, including a deep convolutional autoencoder (DCAE) and a postprocessing, was applied. The DCAE tried to reconstruct the measured CSI with lower-dimensional features. The reconstructed version was then compared to the measurement and used (after robust postprocessing) for attack detection. The disadvantages of [6] were the time delay introduced by the necessary postprocessing unit, no possibility for using multiple CSI estimates at the receiver(s), and the high number of parameters that have to be adjusted manually. In [7], a fully DL approach (cf. [6]) was applied, where a deep neural network with two hidden layers was used. The network used the CSI as the input and output the probability of a tamper attack at predetermined reference positions. While the computational complexity of the method was low, it aimed to identify 1 of N authorized positions, which is a different problem compared to the problem described in [1,6]. Therefore, the method in [7] could not be applied to the problem and was not

compared with the proposed methods. (In this work, the performance comparison was made with [1,6] in the Experimental Section.)

As almost all modern wireless communication systems support multiple-input multiple-output (MIMO), we were motivated to extend the proposed methods in [6] for the case that multiple CSI estimates at the receiver(s) are available (multi-CSI estimates can be available from either a single receiver with multiple antennas or multiple receivers, each with a single antenna). In this work, we thus expanded the framework of [6] to multi-CSI estimates at the receiver(s). Moreover, the drawback of [6] motivated us to propose using a fully DL approach to detect the physical tamper attack. As shown in [6], a DL approach for dimensionality reduction followed by a postprocessing unit detects the physical tamper attack with a high time delay due to using the postprocessing unit. To solve that, we proposed to use a fully DL approach to simultaneously reduce the dimensionality of the input data along with the anomaly detection task. As stated in [11], support vector data description (SVDD) [12] is one of the popular approaches for anomaly detection. We show that a well-tuned version of the proposed method in [13], namely the Deep SVDD, can be used for physical tamper attack detection. In order to fairly evaluate the physical tamper attack detection methods, both the detection performance and the efficiency (e.g., the time delay due to computational complexity in [6]) were taken into account, which was neglected in previous works. The proposed methods offer different characteristics in detection performance, time complexity, and database space budget.

In summary, this paper extends the aforementioned works in two ways: (i) simultaneous with the detection performance analysis, we evaluate efficiency by means of time delay and database space budget; (ii) centralized or decentralized detectors are considered in the design of the proposed methods. In detail, the contributions of the paper are as follows:

- Extending the framework for physical tamper attack detection presented in [6] for the case that multiple CSI estimates at the receiver (s) are available: We suggest two distinct approaches, i.e., centralized and decentralized processing. We show that centralized processing has better detection performance and requires lower database space, while having higher time complexity.
- Proposing the Deep SVDD framework to overcome complexity and latency limitations: We apply Deep SVDD to the physical tamper attack detection problem and show that it has significantly lower complexity compared to the DCAE approach, while having only slightly decreased detection performance.
- Complexity analysis: We characterize the algorithmic complexity by the number of mathematical operations and required database space to compare all investigated methods. We show that there is a trade-off between detection performance and complexity in the proposed methods.
- We evaluate all methods on experimental data from a measurement campaign in a university building.

The rest of this paper is organized as follows: Section 2 introduces the tamper attack detection framework. The tamper attack detection methods will be presented in Section 3. The experimental results are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Tamper Attack Detection Framework

2.1. Problem Statement

As illustrated in Figure 1, considering an OFDM system with one transmitter and R antennas in the receiver(s), the transmitter sends regular messages containing pre-defined preambles. Based on the preamble of the i th packet, the CSI estimate from the r th antenna $\hat{\mathbf{H}}_r^i \in \mathbb{C}^{1 \times S}$ with S being the number of subcarriers is obtained. Meanwhile, an attacker intends to physically manipulate the transmitter by relocating and/or reorienting the transmitter. This physical tamper attack has to be detected using the magnitude of CSI values $|\hat{\mathbf{H}}_r^i|$, $r = 1, \dots, R$. (According to [14], we took only the magnitude values as the input to our algorithm and neglected the phase.) Undoubtedly, several challenges have to be addressed to solve this problem.

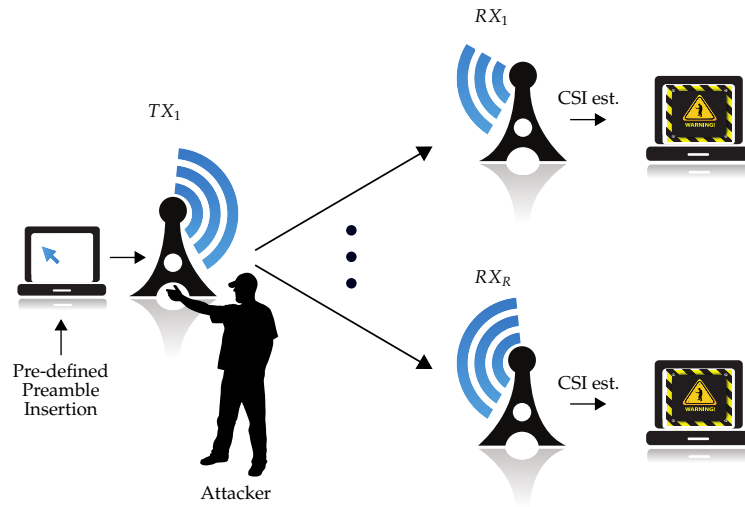


Figure 1. Problem statement.

The transmitted signal is altered by the mobile radio channel. The propagation of the electromagnetic waves from transmitter to receiver(s), as determined by the surrounding environment, determines the mobile radio channel. Any change in the environment also changes the mobile radio channel. As a consequence, the main challenge in detecting a physical tamper attack based on the CSI is to distinguish between changes induced by the physical tamper attack and changes caused by modifications in the surrounding such as people passing by.

Other challenges are time and space constraints during the test phase in applications in which fast system reaction within a restricted database space budget is required. Therefore, time and space complexities have to be taken into account in the evaluation of the proposed methods.

2.2. Detection Framework

We formulated the problem of detecting the physical tamper attack as a data-driven semi-supervised anomaly detection problem. Semi-supervised anomaly detection, according to [15], refers to the problem of finding patterns in data that do not correspond to the expected behavior, i.e., finding if the latest measured $|\hat{\mathbf{H}}_t^i|$ relates to the attack-free training or not.

The proposed algorithms consist of two phases, i.e., training (offline) and testing (online). According to the structure of the receiver(s), the algorithms are applied on either a centralized or decentralized unit. Collections of tamper-free CSI estimates based on different time and environmental conditions are used for training during the offline phase.

Data are collected in the offline phase for N_{Off}^* packets, $\mathbf{H}_{\text{Off},r}$, and in the online phase for N_{On} packets, $\mathbf{H}_{\text{On},r}$ as below:

$$\forall r \in \{1, \dots, R\} :$$

$$\mathbf{H}_{\text{Off},r} \triangleq [|\hat{\mathbf{H}}_{\text{Off},r}^1|, |\hat{\mathbf{H}}_{\text{Off},r}^2|, \dots, |\hat{\mathbf{H}}_{\text{Off},r}^{N_{\text{Off}}^*}|]^T \in \mathbb{R}^{N_{\text{Off}}^* \times S} \quad (1)$$

$$\mathbf{H}_{\text{On},r} \triangleq [|\hat{\mathbf{H}}_{\text{On},r}^1|, |\hat{\mathbf{H}}_{\text{On},r}^2|, \dots, |\hat{\mathbf{H}}_{\text{On},r}^{N_{\text{On}}}|]^T \in \mathbb{R}^{N_{\text{On}} \times S}.$$

For training, the total number of training samples is split into batches with batch size N_{Off} , such that N_{Off}^* is an integer multiple of N_{Off} .

The proposed detection framework is illustrated in Figure 2 in which the CSI is estimated from multiple antennas. Based on the structure of the systems, multiple receivers, each with one antenna or a receiver with multiple antennas, we propose the following detection methods: (i) decentralized processing and the combination of anomaly scores;

(ii) centralized processing with multi-channel extension of the detection algorithms. The blocks in Figure 2 termed “deep” represent either mixed or fully deep approaches.

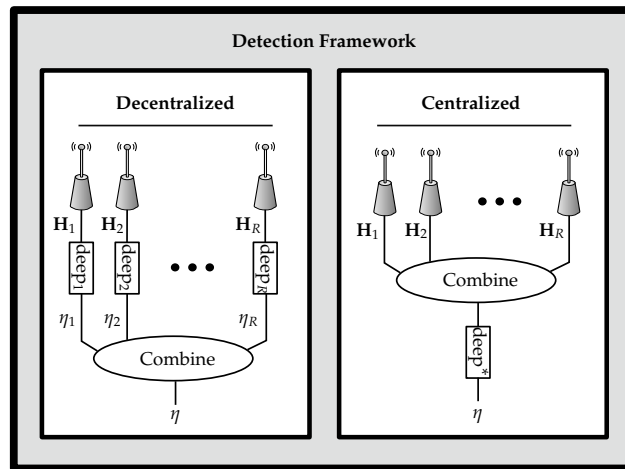


Figure 2. Proposed detection framework.

3. Tamper Attack Detection Methods

In this section, we start with the reformulation of the relevant work [1] in Section 3.1 for the sake of clarity. Afterwards, an extended framework of [6] for multiple antennas at the receiver(s) is proposed in Section 3.2. Finally, a fully DL approach is proposed in Section 3.3 to detect the tamper attack. The proposed methods are compared with respect to detection performance, time complexity, and required database space in the subsequent sections.

3.1. Conventional Threshold Detection

The algorithm presented in [1] is a simple and straightforward approach to detect tamper attacks. Distance values $D_{i,j,r}$ between N_{On} successive CSI estimates in the online phase and N_{Off}^* recorded values from the offline phase for all R receivers are computed as

$$D_{i,j,r} \triangleq \text{Distance}(|\hat{\mathbf{H}}_{\text{Off},r}^i|, |\hat{\mathbf{H}}_{\text{On},r}^j|) \quad (2)$$

where $i = 1, \dots, N_{\text{Off}}^*$, $j = 1, \dots, N_{\text{On}}$, $r = 1, \dots, R$ and then compared to a threshold (Different threshold selection methods are available in the literature. However, threshold selection methods are out of the scope of this work.) value according to

$$\frac{1}{N_{\text{Off}}^* N_{\text{On}} R} \sum_{i=1}^{N_{\text{Off}}^*} \sum_{j=1}^{N_{\text{On}}} \sum_{r=1}^R D_{i,j,r} \underset{\text{Tampering}}{\overset{\text{Tamper Free}}{\leq}} \text{Threshold}. \quad (3)$$

As in [1], *Distance* in (2) refers to a distance metric (e.g., normalized Euclidean distance). The distance metric quantifies the distance between two CSI vectors. Afterwards, in (3), the mean value of the distance over all data is considered to make the decision. This makes the method quite complex since (2) has to be computed for the entire training data set. In this work, *Threshold* was set based on a given performance detection (i.e., false positive rate (FPR)). In Section 4.6.1, we compare the proposed method at FPR = 10%.

To distinguish between conventional environmental changes and a tamper attack, Reference [1] relies on diversity introduced by multiple receiver antennas. This follows the assumption that, e.g., a moving person will impact the CSI only on a subset of the receivers, while the tamper attack impacts the CSI at all receivers. Consequently, the attack detection performance is poor for a single-receiver system. In this work, this approach is referred to as **Threshold Detection**.

3.2. Mixed Deep Approaches: DCAE with PDF Estimator

As an extension of the work in [6], during the offline phase, DCAE_r (see Figure 3) learns a low-dimensional representation of the tamper-free CSI, $|\mathbf{H}_{\text{Off},r}^i|$, for $i = 1, \dots, N_{\text{Off}}^*$ and $r = 1, \dots, R$. From this representation, the CSI is reconstructed (referred to as $|\hat{\mathbf{H}}_r^i|_{\text{rec}}$) and the reconstruction error $\mathbf{e}_r^i$ is computed. After training the network, the Euclidean norm of the reconstruction error is used as the anomaly score $\mathbf{a}_r$ as:

$$\mathbf{a}_r = [\|\mathbf{e}_r^1\|_2, \|\mathbf{e}_r^2\|_2, \dots, \|\mathbf{e}_r^{N_{\text{Off}}^*}\|_2]^T \in \mathbb{R}^{N_{\text{Off}}^*}. \quad (4)$$

The overall scheme for the r -th receiver is depicted in Figures 3 and 4, for the offline and online phases, respectively. For training, the offline CSI estimates in the r -th receiver (denoted by $\mathbf{H}_{\text{Off},r}$ in Figure 3) are tamper-free under different environmental conditions including the movement of people and static environments at different times. Then, the trained DCAE_r (denoted as DCAE_r^{*} (the asterisk superscripts indicate the trained versions of DCAE)) is used for calculating the anomaly score $\mathbf{a}_r$ of the r -th receiver.

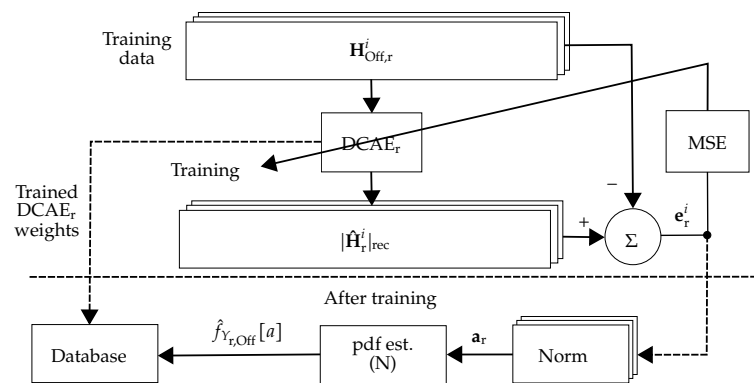


Figure 3. Schematic view of the offline phase of decentralized processing: In the offline phase of the r -th receiver, DCAE_r is trained. Then, the pdf approximation of the anomaly score ($\hat{f}_{Y_r, \text{Off}}[a]$) and the weights of the trained network (DCAE_r^{*}) are stored in the database.

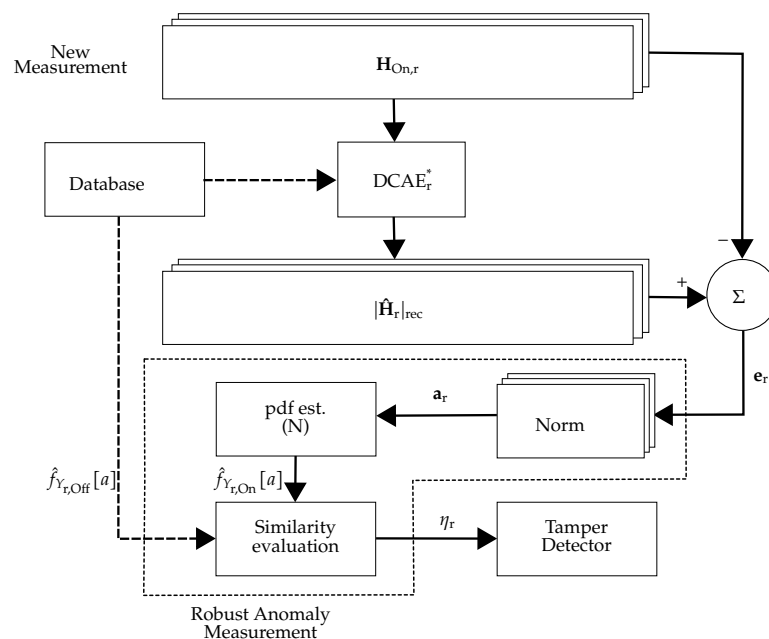


Figure 4. Schematic view of the online phase of decentralized processing: In the online phase of the r -th receiver, DCAE_r^{*} is used to calculate the anomaly score and the pdf approximation of the anomaly score ($\hat{f}_{Y_r, \text{On}}[a]$).

To increase the robustness of physical tamper attack detection, a pdf estimation is applied to the reconstruction error to capture the statistics of the variation in the tamper-free scenario. A straightforward pdf estimation approach is non-parametric kernel density estimation [16]. Let $\mathbf{a}_r \triangleq (a_r^{(1)}, a_r^{(2)}, \dots, a_r^{(N)})$ be independent and identically distributed samples drawn from some univariate distribution with an unknown density $f[a]$ at any given point a . Thus, the kernel density estimator is:

$$\hat{f}[a] = \frac{1}{Nh} \sum_{i=1}^N K\left[\frac{a - a_r^{(i)}}{h}\right], \quad (5)$$

where h is a smoothing parameter referred to as the bandwidth and $K[\cdot]$ is the kernel function. In this work, h is calculated using Silverman's rule of thumb [17] as:

$$h = 0.9 \min\left(\hat{\sigma}, \frac{IQR}{1.34}\right) N^{-\frac{1}{5}}, \quad (6)$$

where $\hat{\sigma}$ is the standard deviation of the samples, IQR is the interquartile range, and N is the sample size. Herein, we make use of the Gaussian kernel function to estimate the pdfs.

To evaluate the anomaly score for R receiver antennas, a decentralized and a centralized approach are considered. Obviously, data processing at each receiver is preferred in cases in which multi-CSI estimates are available from multiple receivers, whereas centralized data processing is preferred in cases in which multi-CSI estimates are available from a single receiver with multiple antennas.

3.2.1. Decentralized Processing with Multiple DCAEs

If decentralized processing is chosen, a single DCAE is applied at each receiver. As shown in the subsequent sections, the size of the input data for this approach is $[N_{\text{Off}}, 1, S]$. The weights of DCAE_r^* and its pdf estimation $\hat{f}_{Y_{r,\text{Off}}}[a]$ are stored in the database for further actions in the online phase.

For each receiver, we used the overlapping index approach [18] to measure the distance of the online-phase pdf $\hat{f}_{Y_{r,\text{On}}}[a]$ to the offline-phase pdf (see Figure 4). The overlapping index distance measure for the r -th receiver, $\eta_r : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0, 1]$, is defined as:

$$\forall r \in \{1, \dots, R\} : \quad \eta_r(\hat{f}_{Y_{r,\text{Off}}}[a], \hat{f}_{Y_{r,\text{On}}}[a]) = \sum_{i=1}^n \min\{\hat{f}_{Y_{r,\text{Off}}}[a_i], \hat{f}_{Y_{r,\text{On}}}[a_i]\} \quad (7)$$

where $\hat{f}_{Y_{r,\text{Off}}}[a]$ and $\hat{f}_{Y_{r,\text{On}}}[a]$ are the pdf approximations of the anomaly score in the offline and online phase of the r -th receiver, respectively. The anomaly score is averaged across the R receivers and compared to a threshold to decide about an attack, i.e.,

$$\frac{1}{R} \sum_{r=1}^R \eta_r \underset{\text{Tampering}}{\overset{\text{Tamper Free}}{\leq}} \text{Threshold}. \quad (8)$$

In this work, two different approaches were considered: (i) Using DCAE without pdf estimator unit: The approach is the same as in Figures 3 and 4, but the pdf estimator unit is exchanged with a unit that calculates the mean value of its inputs. This approach is referred to as **DCAE-D** (**D** for decentralized). (ii) Using DCAE with the pdf estimator unit: The approach is depicted in Figures 3 and 4, which is referred to as **DCAE-DP** (**DP** for decentralized and postprocessing).

3.2.2. Centralized Processing with a Single Multi-Channel DCAE

Here, we propose to use a single DCAE, which is capable of learning the combination of input data from all R receiver antennas. As shown in the subsequent sections, the size

of the input for this structure is $[N_{\text{Off}}, R, S]$. This approach is also enhanced in terms of detection performance by utilizing a pdf estimator. In the offline phase (see Figure 5), the weights of DCAE* and its pdf estimation of the anomaly scores are stored in the database.

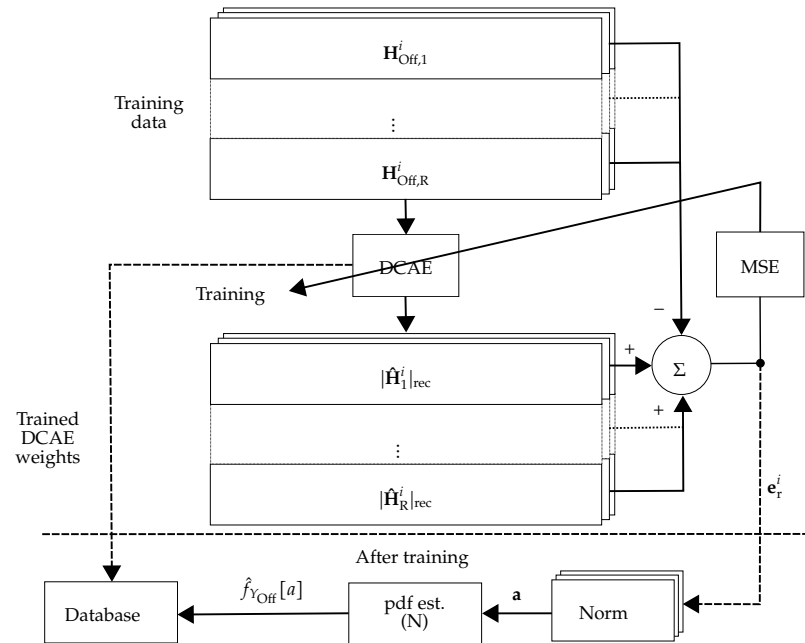


Figure 5. Schematic view of the offline phase of centralized processing: In the offline phase, DCAE is trained. Then, the pdf approximation of the anomaly score ($\hat{f}_{Y_{\text{Off}}}[a]$) and the weights of the trained network (DCAE*) are stored in the database.

In the online phase, DCAE* is used to calculate the anomaly score of newly received CSI estimates. Afterward, its pdf approximation is compared with the pdf stored in the database (see Figure 6). The overlapping index distance measure for this approach, $\eta : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0, 1]$, is defined as:

$$\eta(\hat{f}_{Y_{\text{Off}}}[a], \hat{f}_{Y_{\text{On}}}[a]) = \sum_{i=1}^n \min\{\hat{f}_{Y_{\text{Off}}}[a_i], \hat{f}_{Y_{\text{On}}}[a_i]\} \quad (9)$$

where $\hat{f}_{Y_{\text{Off}}}[a]$ and $\hat{f}_{Y_{\text{On}}}[a]$ are the pdf approximations of the anomaly score in the offline and online phase, respectively. The anomaly score is compared to a threshold to decide about an attack, i.e.,

$$\eta \underset{\text{Tampering}}{\overset{\text{Tamper Free}}{\leq}} \text{Threshold}. \quad (10)$$

As will be shown in Section 4.6.2, by utilizing one DCAE instead of multiple DCAEs, the computational complexity is significantly reduced.

In this work, the approach that utilizes the multi-channel DCAE without a pdf estimator is considered as **DCAE-C** (C for centralized) and the one with a pdf estimator is regarded as **DCAE-CP** (CP for centralized and postprocessing).

3.3. A Fully Deep Approach: Deep SVDD

In this subsection, we exploit Deep SVDD, a novel method proposed by [13], for solving the physical tamper attack detection problem. Deep SVDD is a method that is usually utilized for one-class classification problems in image processing applications. It is a fully deep approach for anomaly detection, which maps the input space into an output space (i.e., a hypersphere of minimum volume) with a neural network (see Figure 7). Feature representations of the data, as well as the one-class classification objective are learned by the neural network. Similar to the previous methods, two different approaches

were considered for multi-receiver operation: either using a neural network for each receiver or using a neural network that accepts multiple inputs. We first reformulated the notation of the method of Deep SVDD [13] to be suitable for the problem at hand. Afterwards, we discuss the details of each approach.

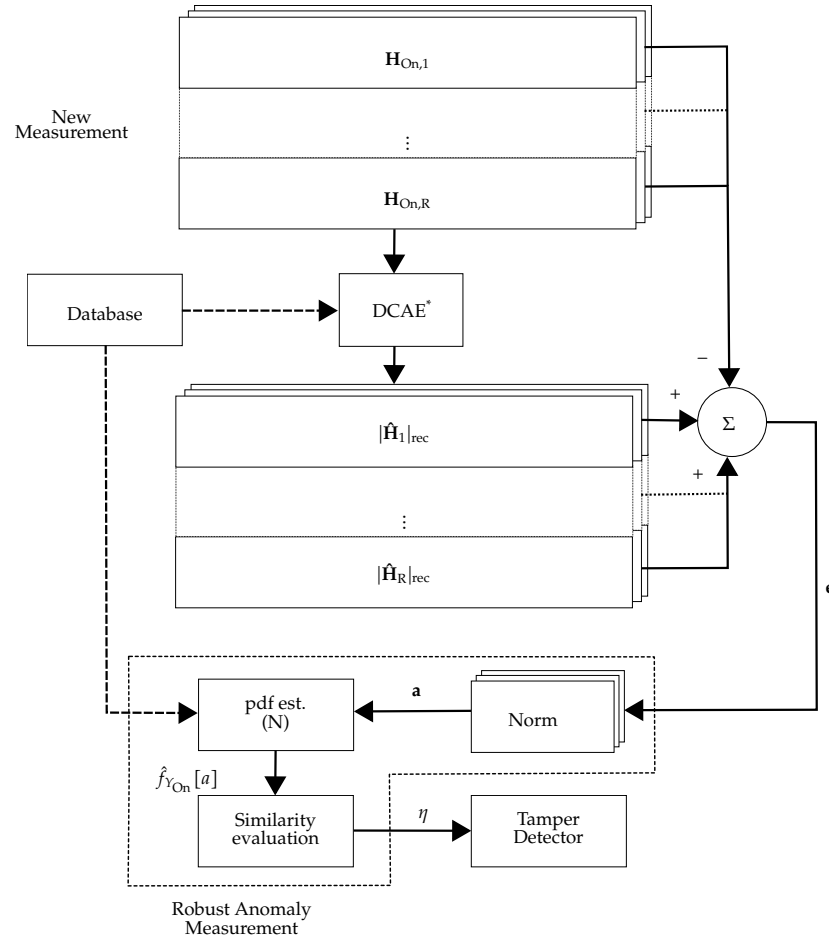


Figure 6. Schematic view of the online phase of centralized processing: In the online phase, DCAE* is used to calculate its anomaly score and the pdf approximation of the anomaly score ($\hat{f}_{Y_{On}}[a]$).

Let $\phi(\cdot; \mathcal{W}) : \chi \rightarrow \mathcal{F}$ be a neural network with $L \in \mathbb{N}$ hidden layers and a set of weights $\mathcal{W} = \{\mathcal{W}^1, \dots, \mathcal{W}^L\}$, where $\mathcal{W}^l$ are the weights of layer $l \in \{1, \dots, L\}$ for some input space $\chi \subseteq \mathbb{R}^S$ and output space $\mathcal{F} \subseteq \mathbb{R}^P$. The feature representation of $|\hat{\mathbf{H}}| \in \chi$ is $\phi(|\hat{\mathbf{H}}|; \mathcal{W})$, i.e., the network ϕ with the weights $\mathcal{W}$. Then, the objective is to learn the network weights $\mathcal{W}$ while minimizing the volume of a data-enclosing hypersphere in output space $\mathcal{F}$, which is defined by radius R and center $\mathbf{c}$. Given the training data $\mathcal{D}_{N_{Off}^*} = \{|\hat{\mathbf{H}}|_1, \dots, |\hat{\mathbf{H}}|_{N_{Off}^*}\}$ on χ , the objective function is defined as:

$$\min_{\mathcal{W}} \frac{1}{N_{Off}^*} \sum_{i=1}^{N_{Off}^*} \|\phi(|\hat{\mathbf{H}}|_i; \mathcal{W}) - \mathbf{c}\|^2 + \frac{\lambda}{2} \sum_{l=1}^L \|\mathcal{W}^l\|_F^2. \quad (11)$$

The first term in (11) considers the distance of each network representation $\phi(|\hat{\mathbf{H}}|_i; \mathcal{W})$ to $\mathbf{c} \in \mathcal{F}$. The second term is a network weight decay regularizer with hyperparameter λ to reduce overfitting of the DL model, where $\|\cdot\|_F$ refers to the Frobenius norm. According to [13], $\mathbf{c} \in \mathcal{F}$ is any fixed hypersphere center.

In the offline phase, the Deep SVDD network is trained using tamper-free CSI estimates (i.e., training data $\mathcal{D}_{N_{Off}^*}$) from different environmental conditions, such as static environments and the movement of people at different times with a batch size of N_{Off} .

After training the Deep SVDD network, the weights of the trained network $\mathcal{W}^*$ and the hypersphere in output space $\mathcal{F}$, which is defined by radius R^* and center $\mathbf{c}^*$, are stored in the database for the following use in the online phase.

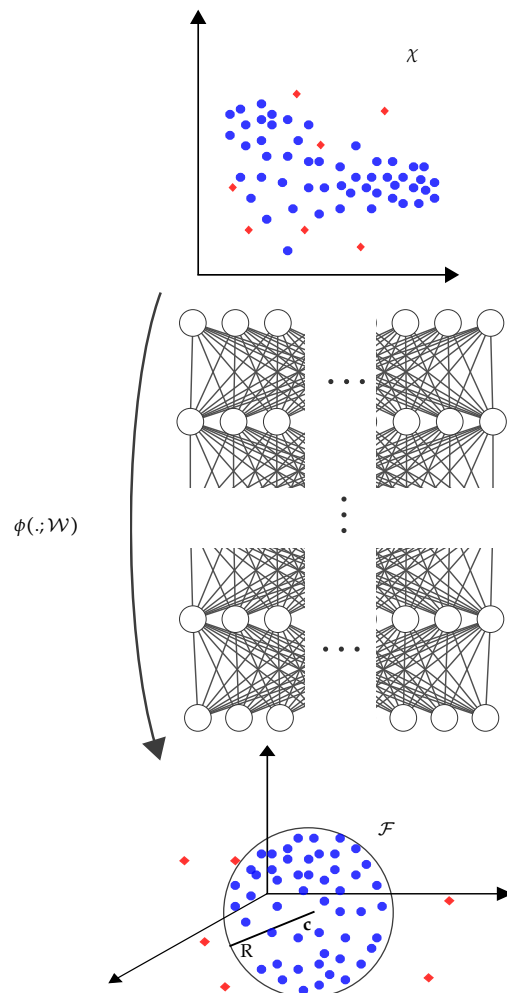


Figure 7. Deep SVDD maps most of the input data into a hypersphere specified by center $\mathbf{c}$ and radius R of a minimum volume using a neural network $\phi(;\mathcal{W})$ with weights $\mathcal{W}$.

In the online phase, the anomaly score of each new estimate, $|\hat{\mathbf{H}}_{\text{On}}| \in \chi$, is calculated by the distance of $|\hat{\mathbf{H}}_{\text{On}}|$ to the center of the hypersphere, i.e.,

$$s(|\hat{\mathbf{H}}_{\text{On}}|) = \|\phi(|\hat{\mathbf{H}}_{\text{On}}|; \mathcal{W}^*) - \mathbf{c}^*\|^2. \quad (12)$$

As stated, two different approaches are considered for multi-receiver operation. Therefore, the decision is made based on the following approaches in the online phase.

3.3.1. Multiple Neural Networks in Decentralized Mode

For using R SVDD-based detectors in decentralized mode, each is trained according to the objective function:

$$\min_{\mathcal{W}_r} \frac{1}{N_{\text{Off}}^*} \sum_{i=1}^{N_{\text{Off}}^*} \|\phi_r(|\hat{\mathbf{H}}_{r|i}; \mathcal{W}_r) - \mathbf{c}_r\|^2 + \frac{\lambda}{2} \sum_{l=1}^L \|\mathcal{W}_r^l\|_F^2, \quad (13)$$

for $r = 1, \dots, R$. In the online phase, the attack detection follows the threshold detection:

$$\frac{1}{R} \sum_{r=1}^R \|\phi_r(|\hat{\mathbf{H}}_r|; \mathcal{W}_r^*) - \mathbf{c}_r^*\|^2 \underset{\text{Tampering}}{\overset{\text{Tamper Free}}{\leq}} R^*, \quad (14)$$

i.e., the average of the distance between the mapped inputs and the stored hypersphere centers $\mathbf{c}_r^*$ is compared with the stored radius R^* . (In this work, R^* was set based on a given performance detection (i.e., FPR)).

3.3.2. Multi-Channel Single Neural Network

In this approach, a single neural network is utilized, which is capable of learning all data from R receivers simultaneously. Therefore, its objective function is:

$$\min_{\mathcal{W}} \frac{1}{N_{\text{Off}}^* R} \sum_{r=1}^R \sum_{i=1}^{N_{\text{Off}}^*} \|\phi(|\hat{\mathbf{H}}|_{i,r}; \mathcal{W}) - \mathbf{c}\|^2 + \frac{\lambda}{2} \sum_{l=1}^L \|\mathcal{W}^l\|_F^2. \quad (15)$$

The decision is made based on the following equation:

$$\|\phi(|\hat{\mathbf{H}}|; \mathcal{W}^*) - \mathbf{c}^*\|^2 \underset{\text{Tampering}}{\overset{\text{Tamper Free}}{\leq}} R^*. \quad (16)$$

It is worth noting that the network weights $\mathcal{W}_r^*$, R^* and $\mathbf{c}_r^*$, for $r = 1, \dots, R$, in the decentralized approach (see (14)), and the network weights $\mathcal{W}^*$, R^* , and $\mathbf{c}^*$ in the centralized approach (see (16)) are sufficient to characterize the multiple Deep SVDD models and the multi-channel Deep SVDD model, respectively. No further data need to be stored in the database for physical tamper attack detection. In contrast, for the DCAE-based approaches, not only the weights of the trained network(s), but also a representation of the offline anomaly scores (i.e., the trained pdf) and the threshold have to be stored. As a result, it is expected that Deep SVDD has a lower database space requirement, which leads to faster testing and lower time delay in the online phase. Moreover, the pdf estimator in the DCAE-based methods relies on N_{On} successive CSI estimates in the online phase (i.e., referred to as the online batch). In contrast to the DCAE-based methods, the value of N_{On} in the tamper attack decision for Deep-SVDD-based methods does not significantly affect the detection performance (shown in Section 4.5.3).

In this work, the decentralized and centralized Deep SVDD approaches are referred to as **SVDD-D** (D for decentralized) and **SVDD-C** (C for centralized), respectively.

4. Experimental Results

The presented methods were evaluated and compared in a setup in which two receivers, each equipped with a single antenna, should detect a tamper attack performed at the transmitter, also equipped with a single antenna. As the tamper attack, we considered a rotation of the transmitter compared to its default orientation. The tamper attack detection is based on CSI estimates acquired with a software-defined radio and the Gnuradio OFDM project [19]. The parameters were selected as described in the following. (Note that the underlying OFDM estimates are identical to those in our previous work in [6]).

4.1. OFDM System

The transmitter and receiver nodes were composed of a host computer connected to a USRP X310 equipped with a directional antenna [20]. In the system, data were exchanged among nodes with a frame structure. Each frame consisted of nine data OFDM symbols and three preamble symbols. In the frequency domain, each symbol contained 200 data subcarriers, 48 null subcarriers, and 8 pilot subcarriers for a channel bandwidth of 25 MHz. Each OFDM symbol duration consisted of a 10.24 μs IFFT period followed by a 1.25 μs guard interval. The carrier frequency was set to 2.55 GHz. The channel was estimated

based on one preamble symbol with a least-squares approach. The channel estimation (i.e., denoted by $\hat{\mathbf{H}}$) was used in the aforementioned physical tamper attack detection methods.

4.2. Environment

As seen in Figure 8, we evaluated the tamper attack detection methods in two distinct environments: an office and a hall. The transmitter (indicated by TX) and two receivers (indicated by RX1 and RX2) were placed on the top of shelves with a 230 cm elevation in the office and desks with a 140 cm elevation in the hall environment. To have physical tamper attacks and tamper-free scenarios in the estimates, we took into account eight different antenna orientations, including the tamper-free default orientation and rotations $r1, r2, \dots, r7$ (cf. Figure 8).

Since discriminating between environmental changes and physical tamper attacks is the key challenge in this attack detection problem, we considered eight scenarios for the tamper-free default orientation (see Table 2). In these eight scenarios, typical situations in an office and a hall were considered in which small-scale (scenarios A and B) and large-scale (scenarios C to H) movements of persons appear, which introduce variations in the CSI. These CSI variations due to the movement of people (i.e., environmental changes) need to be learned for our proposed methods to be able to distinguish them from the CSI changes due to physical tamper attacks.

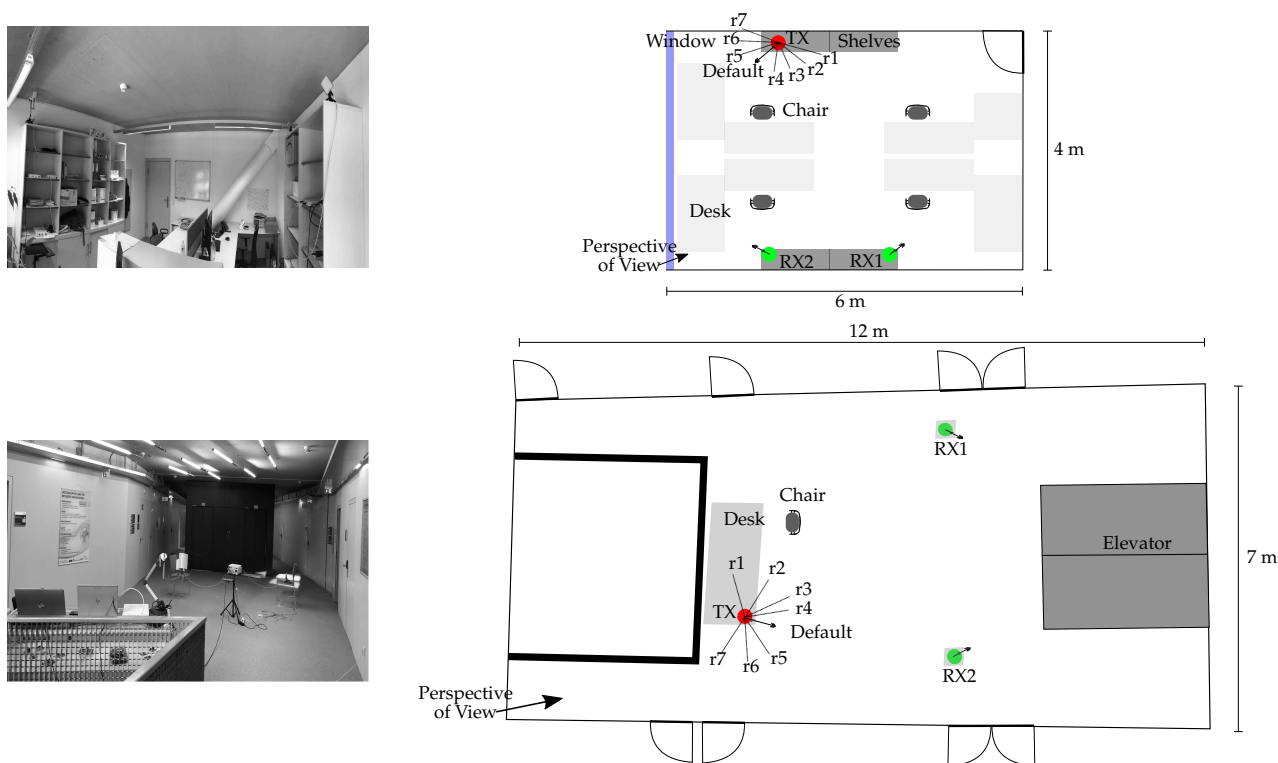


Figure 8. Measurement environments: (top) office (4 m × 6 m) and (bottom) hall (7 m × 12 m), depicted with photos and the layout. The transmitter (indicated by TX) and two receivers (indicated by RX1 and RX2) are denoted. Orientations $r1, r2, \dots, r7$ are considered as physical tamper attacks.

4.3. Parameters of the DL-Based Methods

For all neural networks in this work, we used LeNet-type convolutional neural networks (CNNs). In the following, two structures are described in Figures 9 and 10, which were used for the aforementioned methods.

Table 2. Measurement scenarios for the tamper-free default orientation.

Scenario	Description
A	a person sits in chair 1
B	same as A one hour later
C	a person walks in the area randomly
D	same as C one hour later
E	two persons walk in the area randomly
F	same as E one hour later
G	three persons walk in the area randomly
H	same as G one hour later

4.3.1. DCAE

The DCAE-based methods require parameter selection for the DCAE core block (responsible for representation learning) and the postprocessing unit (responsible for anomaly detection). For the latter, the free parameter is the online batch size N_{On} . For the DCAE, each convolutional module is composed of a one-dimensional convolutional layer followed by ELU activation functions (as depicted in Figure 9 with Conv1D), BatchNorm1D, and 2×1 Max-Pooling layers. The transposed convolutional modules are composed of a one-dimensional transposed convolutional layer followed by ELU activation functions (as depicted in Figure 9 with ConvT1D), BatchNorm1D, and 2×1 Interpolate layers. As depicted in Figure 9, the encoder is composed of three convolutional modules with different numbers of filters (eight, four, and one) and a final fully dense layer of 26 units. The mirrored structured is used for the decoder. The colors in Figure 9 show the operations that were applied. Dark red, light red, dark blue, and light blue color tensors illustrate that a convolutional module, BatchNorm1D followed by 2×1 Max-Pooling, BatchNorm1D followed by 2×1 Interpolate layers, and a transposed convolutional module were applied on the previous corresponding tensor, respectively. The black color tensor indicates the compressed representation of the input. The data flow starts from the top left-hand side and ends at the bottom left-hand side.

4.3.2. Deep SVDD

As stated in Section 3.3, representation learning and anomaly detection are simultaneously performed in Deep SVDD. Therefore, the structure of the encoder of DCAE is adopted for the structure of Deep SVDD (cf. Figures 9 and 10). By this approach, the time complexity along with the space complexity of the Deep-SVDD-based methods compared to the DCAE-based methods are reduced. (Note that time and space complexities in the ML context are equivalent to computational time and memory requirements in communications engineering.)

4.4. How to Train the Networks

As Table 3 shows, for the training, validation, and testing data, different CSI estimate datasets from the office and the hall environment were collected. To observe the genuine performance results of the neural networks, they were trained in the default antenna orientation in scenarios A, C, E, and G. Other scenarios and other antenna orientation were utilized for evaluating and testing.

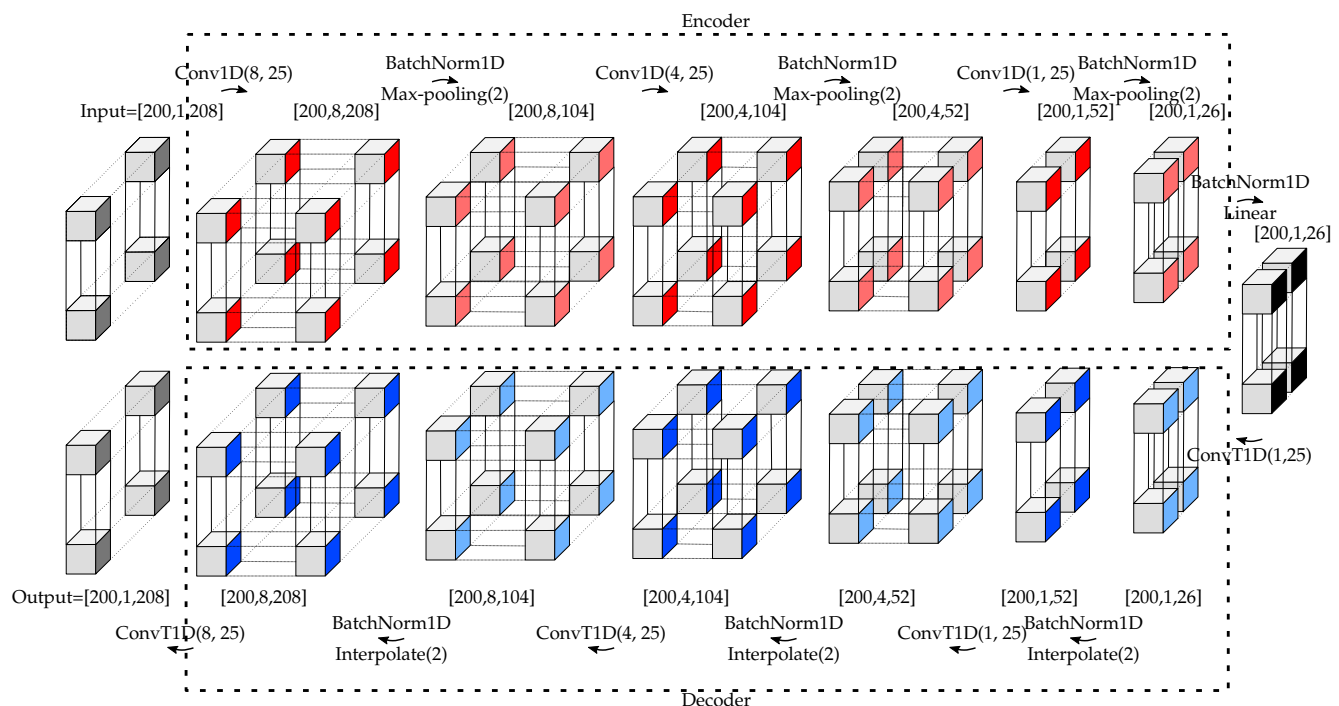


Figure 9. Structure of DCAE. The size of each tensor is depicted in the figure. This structure is used for the proposed decentralized approaches. In the case of centralized approaches, only the size of the input is changed (in the second dimension, 1 is replaced by R).

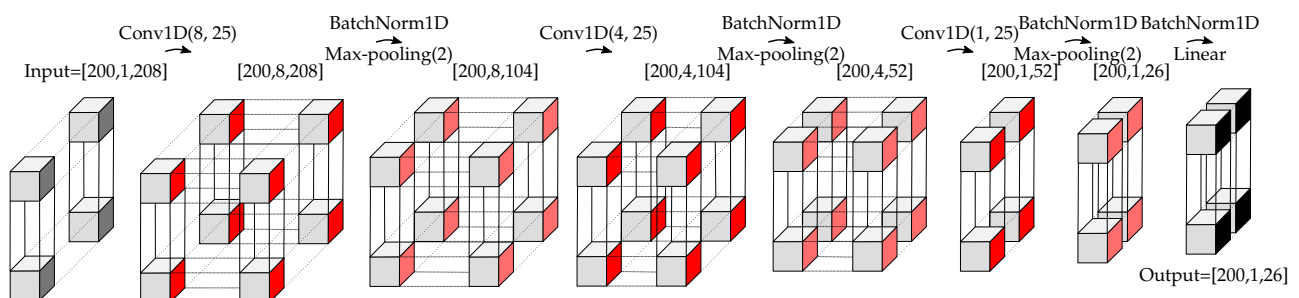


Figure 10. Structure of Deep SVDD. The size of each tensor is depicted in the figure. This structure is used for the proposed decentralized approaches. In the case of centralized approaches, only the size of the input is changed (in the second dimension, 1 is replaced by R).

Table 3. Size of the CSI estimate datasets from the office and the hall environment.

Data Set	Office	Hall
Training	96,000	101,800
Validation	23,600	24,800
Testing	214,400	263,800

A variant of stochastic gradient descent (i.e., Adam [21]) was utilized to optimize the weights for each method using backpropagation. We made use of the Keras library [22] and implemented Deep SVDD in Pytorch [23].

With an initial learning rate of 10^{-5} and then 10^{-6} , we used a two-phase learning rate schedule (searching and fine-tuning). We trained 20 epochs with the learning rate for searching and 5 epochs with the learning rate for fine-tuning. This was repeated 300 times to obtain precise results. The networks were implemented on TensorFlow with Linux

(ubuntu 18.04) running on an 8-core ARM v8.2 64-bit CPU and a 512-core NVIDIA Volta GPU. Table 4 summarizes the parameters used to train the neural networks.

Table 4. Neuralnetwork parameters.

Description	Value
Optimizer	Adam
N_{Off}	200
Weight Decay	10^{-1}
Number of Epochs	20 + 5
Activation Function	ELU
Learning Rate	$10^{-5} \rightarrow 10^{-6}$

The learning curves of the two neural networks are shown in Figure 11. We defined a training batch size of 200. It is worth noting that **DCAE-DP** and **DCAE-CP** use the same neural networks as **DCAE-D** and **DCAE-C**, respectively. Hence, their learning curves are the same as the learning curves of **DCAE-D** and **DCAE-C**, respectively, and they are not plotted in Figure 11. Moreover, in **DCAE-D** and **SVDD-D**, there are R (i.e., two in this work) neural networks. Therefore, there are two learning curves for these methods in Figure 11.

Figure 11 shows that the neural networks were appropriately trained. For an easier interpretation, the quadratic loss functions for Deep SVDD and the other DL methods are not normalized in the figure. Therefore, a shift between the learning curves of the Deep-SVDD-based methods and the DCAE-based methods is observed in Figure 11.

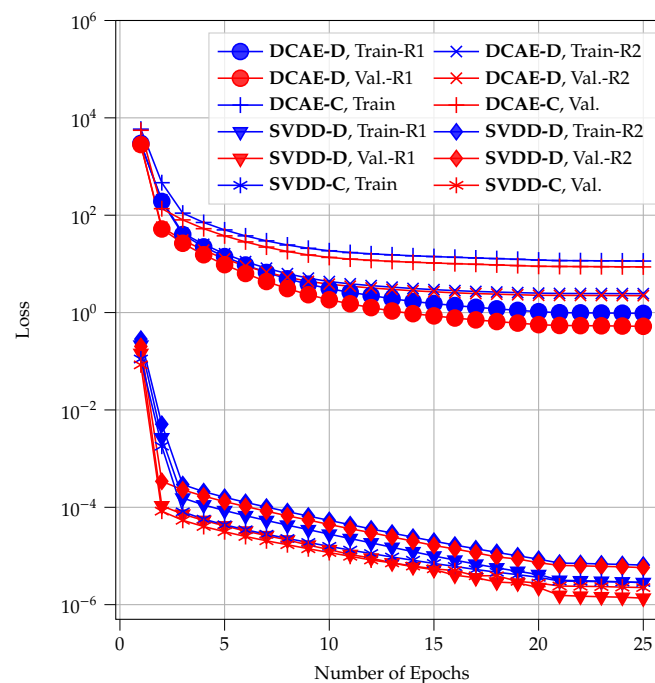


Figure 11. Learning curves of the DCAE used in **DCAE-D**, **DCAE-DP**, **DCAE-C**, and **DCAE-CP** and the Deep SVDD used in **SVDD-D** and **SVDD-C**. The learning curves for the training dataset (denoted as Train), the validation dataset (denoted as Val.), and two receivers, in the case of decentralized approaches (denoted as R1 and R2), are depicted.

4.5. Evaluation Criteria

4.5.1. AUC-ROC

The receiver operating characteristic (ROC) is a graphical representation of classifier performance. At various threshold settings, the ROC curve plots the true positive rate (TPR) versus the FPR. The most important evaluation metric for the performance of any classification model is the area under the curve (AUC) of the ROC. We measured the AUC-ROC of the different methods where an excellent one has an AUC close to one, which means it has an almost perfect capability of separation.

4.5.2. Complexity

The complexity of an algorithm indicates the number of features or terms included in the algorithm to execute its task. Time and space are two main complexity measures of the efficiency of an algorithm.

Time complexity is the number of operations in an algorithm required to complete its task with respect to the input size. It can be experimentally evaluated by measuring the time the algorithm requires to accomplish the task. Space complexity denotes the amount of space used by the algorithm for its task, for various input sizes. In this work, we considered the database space the algorithm requires to accomplish its task as the space complexity. We evaluated the time complexity experimentally by runtime measurements of the algorithms and theoretically by the number of basic operations (multiplications and additions) for the forward path. We evaluated the space complexity of an algorithm by the number of floating point elements that are required. Note that time and space complexity depends on other factors as well, including hardware, operating system, processors, other running programs, etc. Since we established the same conditions for the aforementioned algorithms while analyzing them, we did not consider any of these factors in their evaluation.

4.5.3. Detection Performance Alongside Complexity

An ideal tamper-detection method is achieved when the detection performance is high and the complexity is low. However, there is usually a trade-off between the detection performance and the complexity. To fairly evaluate the tamper attack detection methods, the detection performance along with the complexity should be considered.

4.6. Tamper Attack Detection Performance

The aim of this subsection is to compare the performance of the proposed methods in terms of attack detection, time complexity, and space complexity. Furthermore, environmental dependency along with detection performance versus complexity are discussed.

4.6.1. ROC Evaluation

The comparison of the detection performance of the aforementioned methods is shown in Figure 12. In this work, similar to [7], we chose the threshold in Figure 12 such that it maximizes the TPR and minimizes the FPR in almost all methods, resulting in an FPR = 10%, which is indicated by the dashed-dotted line. As can be seen from Figure 12, the DCAE-based approaches have a better tamper attack detection rate compared to the Deep SVDD approaches. This is because they spend more effort on signal processing (representing the learned CSI and the postprocessing), which results in higher time and space complexity. As illustrated in Figure 12, DCAE-C and DCAE-CP have the best tamper attack detection rates. The Deep SVDD approaches have a better tamper attack detection rate compared to the threshold detection approach, although they have lower space complexity.

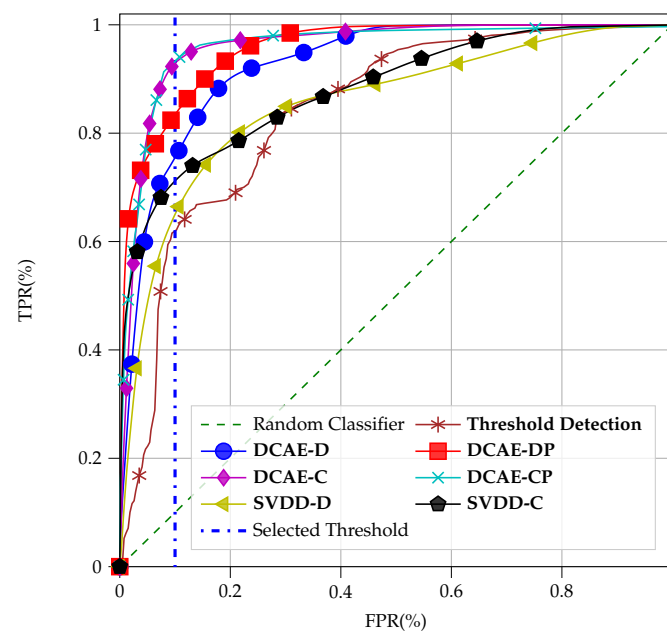


Figure 12. ROC curve of the methods with $N_{On} = 300$ in the two environments on average.

4.6.2. Time and Space Complexities

The number of basic operations (multiplications and additions) in the forward path that each method requires is compared in Table 5. Furthermore, the time each method required in the offline and online phases was measured and is shown in Table 6. There is a direct relation between time complexity and test duration for each method (cf. Tables 5 and 6). The amount of data each method stores in the database is another important factor, at least in some applications. Thus, they are presented in Table 7.

Table 5. Time complexity of the methods for each sample in the online phase. N_{Off} and S are the training batch size and number of subcarriers, respectively.

Method	No. of Multiplications	No. of Additions
Threshold Detection	$\approx 2SN_{Off}^*$	$\approx 4SN_{Off}^*$
DCAE-D	$\approx 1978.75N_{Off}S$	$\approx 1987.95N_{Off}S$
DCAE-DP	$\approx (1978.75 + \frac{N_{On}-1}{S})N_{Off}S$	$\approx 1987.95N_{Off}S$
DCAE-C	$\approx 1189.38N_{Off}S$	$\approx 1193.94N_{Off}S$
DCAE-CP	$\approx (1189.38 + \frac{N_{On}-1}{S})N_{Off}S$	$\approx 1193.94N_{Off}S$
SVDD-D	$\approx 702.00N_{Off}S$	$\approx 702.95N_{Off}S$
SVDD-C	$\approx 351.00N_{Off}S$	$\approx 352.44N_{Off}S$

As stated, the drawback of the DCAE-based method proposed in [6] is its time delay. The reason is that the pdf estimations in the online phase are time consuming. This issue is addressed in this paper by using the Deep SVDD approach. According to Table 5 (number of basic operations) and Table 6 (test duration), it is shown that the test duration for **SVDD-D** and **SVDD-C** is significantly shortened compared to **DCAE-D** (31.5%), **DCAE-DP** (72.6%), **DCAE-C** (39.0%), and **DCAE-CP** (62.8%), respectively. Note that there is a slight difference in the training duration between the hall and office environment due to the different sizes of their datasets (cf. Table 3).

Table 6. Performance of the methods on the measurement data (average over online batches).

Dataset	Method	AUC	Training Dur.	Test Dur.
Office	Threshold Detection	85.66%	-	3.92 s
Hall	Threshold Detection	83.53%	-	4.76 s
Office	DCAE-D	92.49%	329.50 s	16.99 s
Hall	DCAE-D	92.46%	334.40 s	17.14 s
Office	DCAE-DP	94.56%	337.34 s	42.08 s
Hall	DCAE-DP	94.53%	342.25 s	42.46 s
Office	DCAE-C	99.93%	237.65 s	12.95 s
Hall	DCAE-C	91.30%	247.54 s	16.08 s
Office	DCAE-CP	99.99%	244.54 s	21.33 s
Hall	DCAE-CP	94.91%	261.24 s	26.24 s
Office	SVDD-D	85.87%	242.63 s	11.59 s
Hall	SVDD-D	85.64%	246.99 s	11.79 s
Office	SVDD-C	93.00%	177.63 s	7.95 s
Hall	SVDD-C	82.03%	185.22 s	9.76 s

Table 7. Space complexity of methods.

Method	No. of Floating Point Elements
Threshold Detection	SN_{Off}^*R
DCAE-D	2901R
DCAE-DP	$(2901 + N_{\text{On}})R$
DCAE-C	3301
DCAE-CP	$3301 + N_{\text{On}}R$
SVDD-D	1776R
SVDD-C	1976

4.6.3. Environmental Dependency

We can conclude that physical tamper attack detection in the hall environment is more challenging than in the office environment. As shown in Table 6, the attack detection performance of each method in the hall environment is poorer than the corresponding one in the office environment in terms of the AUC. The reason is that the surrounding environment in the hall is prone to be more time-variant compared to the office environment since the environment is not bounded on two sides. Thus, there are more possibilities of people moving around in the hall than in the office. As we used approximately the same amount of training data in both environments, we could not learn the environmental variation to the same amount as in the office environment. We expect that, with a more extensive training, the tamper detection performance can reach similar values as in the office environment.

4.6.4. Detection Performance vs. Complexity

According to Table 6, using **DCAE-CP** not only has superior detection performance compared to the other methods, but shows also relatively fast training and testing phases. In general, there is a trade-off among time complexity, space complexity, and detection performance. As shown in Tables 5 and 7, **SVDD-D** and **SVDD-C** show the smallest time

and space complexity, but as expected, their detection performance is not as high as the others. In contrast, **DCAE-D** and **DCAE-DP** have a larger space complexity than **DCAE-CP**, but achieve a lower detection performance. All in all, the aforementioned factors have to be considered simultaneously to select the optimal method.

It is worth noting that, although the detection performances of **SVDD-D** and **SVDD-C** are acceptable, but not as high as the performances of the other DL methods, their time and space complexities are very low and, thus, might be suitable for certain applications.

As brought up in [6], although the time delay will be longer by increasing the online batch size for **DCAE-DP**, the detection performance will be enhanced. Therefore, we investigated the impact of the online batch size on the detection performance. Figure 13 plots the AUC-ROC over different values of N_{On} . Since N_{On} significantly influences the performance of the pdf estimators, only the detection performances of **DCAE-DP** and **DCAE-CP** change over different values of N_{On} . As expected, by increasing N_{On} , the AUC-ROC improves. Note that the value of N_{On} in the depicted range does not affect the performance of **DCAE-CP** in the office environment because the method can learn environmental variation very well. Obviously, the detection performances of the DCAE-based methods with postprocessing will be affected if the value of N_{On} is chosen too small. According to Figure 13 and the relation between N_{On} and time complexity, the optimal value of N_{On} for **DCAE-CP** in the hall environment is 200.

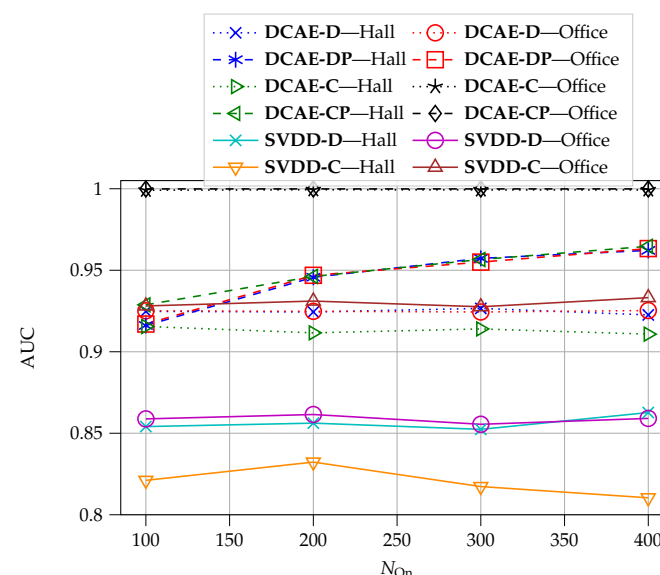


Figure 13. Impact of online batch size N_{On} on the AUC-ROC.

4.7. Further Discussion

The structures of **DCAE-C** and **DCAE-CP** are similar to those of **DCAE-D** and **DCAE-DP**, respectively. Instead of using R neural networks in **DCAE-D** and **DCAE-DP**, a single multi-channel input neural network is utilized in **DCAE-C** and **DCAE-CP**. According to Table 7, the space complexity of each centralized method is almost R -times less than the corresponding decentralized method. In general, the centralized methods outperform the decentralized method in terms of the AUC and space complexity. This is because the centralized methods consider the CSI from all receivers together in the offline phase. However, this is not the case for the SVDD-based method in the hall environment. The reason is the low complexity of the method and the high variation of its signal due to the environment structure. We expect that, with a more complex neural network in the structure of the SVDD-based method, the tamper attack detection performance can achieve similar values as in the office environment.

As stated in [6], DCAE is a method for representation learning. Physical tamper attack detection with DCAE-based methods is performed by the postprocessing units. With

Deep SVDD, representation learning and physical tamper attack detection are performed simultaneously. Therefore, increasing the complexity of the neural networks used for Deep SVDD leads to improved detection performance. As in our work, the goal was to minimize complexity, we restricted the number of hidden layers to four in the neural networks used for Deep SVDD. A larger number of hidden layers could have improved the detection performance, but would have increased the complexity.

From Tables 5 and 6, we found that **Threshold Detection** performs much faster than the other methods; however, its tamper attack detection performance is rather low. It is worth mentioning that there is a linear relation between the number of collected packets in the offline phase (N_{Off}^*) and the time complexity for **Threshold Detection** (see Table 5). However, N_{Off}^* does not affect the time complexity for the DL methods. This is a significant advantage of DL methods over non-DL methods.

5. Conclusions

In this paper, we proposed and evaluated two DL-based approaches for detecting a physical tamper attack using CSI in an OFDM-based wireless communication system with multiple receiver antennas with respect to detection performance, time, and space complexities. The two main challenges of this problem were to distinguish between antenna orientation changes and communication environment changes and to achieve high detection performance along with a low level of complexity. To achieve a robust attack detector based on different levels of complexity, we used DCAE and Deep SVDD neural networks in centralized and decentralized structures. With our experiment, we concluded that there is a trade-off between detection performance and complexity in the proposed methods. It was shown that the DCAE-based methods outperform the SVDD-based methods in terms of detection, while the SVDD-based methods have almost two-times lower time and space complexities.

Author Contributions: Conceptualization, E.D., B.E. and A.S.; Data curation, E.D.; Funding acquisition, A.S.; Investigation, B.E.; Methodology, E.D., B.E., N.B.T. and A.S.; Project administration, A.S.; Software, E.D.; Supervision, B.E. and A.S.; Validation, E.D. and B.E.; Visualization, E.D.; Writing—original draft, E.D.; Writing—review & editing, E.D., B.E., N.B.T. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported in part by the “University SAL Labs” initiative of Silicon Austria Labs (SAL) and its Austrian partner universities for applied fundamental research for electronic based systems and by the InSecTT project, Johannes Kepler University, and the COMET-K2 “Center for Symbiotic Mechatronics” of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria. InSecTT received funding from the Electronic Component Systems for European Leadership Joint Undertaking under Grant Agreement No. 876038. The document reflects only the authors' views, and the Commission is not responsible for any use that may be made of the information it contains. This Joint Undertaking receives support from the European Union's Horizon 2020 Research and Innovation Programme and Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium, and Norway.

Data Availability Statement: The dataset used in this work is available at <https://github.com/isaac1369/Physical-Tamper-Attack-Detection.git> (accessed on 20 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bagci, I.E.; Roedig, U.; Martinovic, I.; Schulz, M.; Hollick, M. Using Channel State Information for Tamper Detection in the Internet of Things. In Proceedings of the 31st Annual Computer Security Applications Conference, Los Angeles, CA, USA, 7–11 December 2015; Association for Computer Machinery: New York, NY, USA, 2015; pp. 131–140. [CrossRef]
2. Kaemarungsi, K.; Krishnamurthy, P. Properties of Indoor Received Signal Strength for WLAN Location Fingerprinting. In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, MA, USA, 26 August 2004; pp. 14–23. [CrossRef]

3. Patwari, N.; Kaseria, S.K. Robust Location Distinction Using Temporal Link Signatures. In Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, Montreal, QC, Canada, 9–14 September 2007; Association for Computer Machinery: New York, NY, USA, 2007; pp. 111–122. [\[CrossRef\]](#)
4. Wu, K.; Xiao, J.; Yi, Y.; Chen, D.; Luo, X.; Ni, L.M. CSI-Based Indoor Localization. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 1300–1309. [\[CrossRef\]](#)
5. Maia, E.; Praça, I.; Mantzana, V.; Gkotsis, I.; Petrucci, P.; Biasin, E.; Kamenjasevic, E.; Lammari, N. Security Challenges for the Critical Infrastructures of the Healthcare Sector. In *Cyber-Physical Threat Intelligence for Critical Infrastructures Security: A Guide to Integrated Cyber-Physical Protection of Modern Critical Infrastructures*; Now Publishers: Boston, MA, USA, 2020; pp. 142–165.
6. Dehmollaian, E.; Etzlinger, B.; Torres, N.B.; Springer, A. Using Channel State Information for Physical Tamper Attack Detection in OFDM Systems: A Deep Learning Approach. *IEEE Wireless Commun. Lett.* **2021**, *10*, 1503–1507. [\[CrossRef\]](#)
7. Chan, P.Y.; Lai, A.I.C.; Wu, P.Y.; Wu, R.B. Physical Tampering Detection Using Single COTS Wi-Fi Endpoint. *Sensors* **2021**, *21*, 5665. [\[CrossRef\]](#)
8. Elngar, A. IoT-based Efficient Tamper Detection Mechanism for Healthcare Application. *Int. J. Netw. Secur.* **2018**, *20*, 489–495. [\[CrossRef\]](#)
9. Pathak, A.K.; Saguna, S.; Mitra, K.; Åhlund, C. Anomaly Detection using Machine Learning to Discover Sensor Tampering in IoT Systems. In Proceedings of the IEEE International Conference on Communications, Montreal, QC, Canada, 14–23 June 2021; pp. 1–6. [\[CrossRef\]](#)
10. Prasad, R. *OFDM for Wireless Communications Systems*; Artech House: New York, NY, USA, 2004.
11. Bergman, L.; Cohen, N.; Hoshen, Y. Deep nearest neighbor anomaly detection. *arXiv* **2020**, arXiv:2002.10445.
12. Tax, D.M.; Duin, R.P. Support vector data description. *Mach. Learn.* **2004**, *54*, 45–66. [\[CrossRef\]](#)
13. Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S.A.; Binder, A.; Müller, E.; Kloft, M. Deep One-Class Classification. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; PMLR: Stockholm, Sweden, 2018; Volume 80, pp. 4393–4402.
14. Sobehy, A.; Renault, É.; Mühlethaler, P. CSI-MIMO: K-nearest Neighbor applied to Indoor Localization. In Proceedings of the 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [\[CrossRef\]](#)
15. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 1–58. [\[CrossRef\]](#)
16. Rudemo, M. Empirical Choice of Histograms and Kernel Density Estimators. *Scand. J. Stat.* **1982**, *9*, 65–78.
17. Silverman, B. *Density Estimation for Statistics and Data Analysis Estimation Density*; Monographs on Statistics and Applied Probability; Chapman and Hall: London, UK, 1986; pp. 45–47.
18. Pastore, M.; Calcagni, A. Measuring Distribution Similarities Between Samples: A Distribution-Free Overlapping Index. *Front. Psychol.* **2019**, *10*, 1089–1096. [\[CrossRef\]](#)
19. Zivkovic, M.; Mathar, R. Design Issues and Performance Evaluation of a SDR-based Reconfigurable Framework for Adaptive OFDM Transmission. In Proceedings of the 6th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, Las Vegas, NV, USA, 19 September 2011; pp. 75–82.
20. Huber+Suhner. SENCITY Spot-S Indoor Antenna 1324.19.0002. Available online: https://ecatalog.hubersuhner.com/product/E-Catalog/Radio-frequency/Antennas-accessories/Antennas/22649580/1324.19.0002?gv_refer=catsearch (accessed on 20 July 2022).
21. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
22. Chollet, F. Keras: The Python Deep Learning library. Available online: <https://keras.io/#support> (accessed on 25 July 2022).
23. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in PyTorch. In Proceedings of the NIPS 2017 Autodiff Workshop, Long Beach, CA, USA, 9 December 2017; pp. 1–4.