DeepCSI: Rethinking Wi-Fi Radio Fingerprinting Through MU-MIMO CSI Feedback Deep Learning

Francesca Meneghello*, Michele Rossi*†, and Francesco Restuccia§
* Department of Information Engineering, University of Padova, Italy
† Department of Mathematics, University of Padova, Italy
§ Institute for the Wireless Internet of Things, Northeastern University, United States
e-mail: meneghello@dei.unipd.it, michele.rossi@unipd.it, frestuc@northeastern.edu

2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS) | 978-1-6654-7177-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDCS54860.2022.00106

***Abstract*—We present DeepCSI, a novel approach to Wi-Fi radio fingerprinting (RFP) which leverages standard-compliant beamforming feedback matrices to authenticate MU-MIMO Wi-Fi devices on the move. By capturing unique imperfections in off-the-shelf radio circuitry, RFP techniques can identify wireless devices directly at the physical layer, allowing low-latency low-energy cryptography-free authentication. However, existing Wi-Fi RFP techniques are based on software-defined radio (SDRs), which may ultimately prevent their widespread adoption. Moreover, it is unclear whether existing strategies can work in the presence of MU-MIMO transmitters – a key technology in modern Wi-Fi standards. Conversely from prior work, DeepCSI does not require SDR technologies and can be run on any low-cost Wi-Fi device to authenticate MU-MIMO transmitters. Our key intuition is that imperfections in the transmitter's radio circuitry percolate onto the beamforming feedback matrix, and thus RFP can be performed without explicit channel state information (CSI) computation. DeepCSI is robust to inter-stream and inter-user interference being the beamforming feedback not affected by those phenomena. We extensively evaluate the performance of DeepCSI through a massive data collection campaign performed *in the wild* with off-the-shelf equipment, where 10 MU-MIMO Wi-Fi radios emit signals in different positions. Experimental results indicate that DeepCSI correctly identifies the transmitter with an accuracy of up to 98%. The identification accuracy remains above 82% when the device moves within the environment. To allow replicability and provide a performance benchmark, we pledge to share the 800 GB datasets – collected in static and, for the first time, dynamic conditions – and the code database with the community.**

I. Introduction and Motivation

The sheer expansion of Internet of Things (IoT) is rapidly saturating unlicensed spectrum bands [1]. With the global mobile data traffic projected to reach 164 exabytes per month in 2025 [2], spectrum congestion will soon decrease data throughput to intolerable levels. To alleviate the issue, the Federal Communication Commission (FCC) has recently released 150 MHz additional bandwidth in the 3.5 GHz spectrum band [3], as well as 1.2 GHz in the 6 GHz band (5.925–7.125), the latter providing opportunities to use up to 320 MHz channels to expand capacity and increase network performance [4].

The release of these spectrum bands for unlicensed use implies that previously licensed users (also known as incumbents), unlicensed Wi-Fi devices [5] and 5G cellular networks [6] will need to coexist in the same spectrum bands. This will necessarily require the enactment of strict, fine-grained dynamic spectrum access (DSA) rules [7], which will require spectrum administrators to continuously monitor *which* unlicensed Wi-Fi device is using the spectrum, and *when* the device is using it. To this end, cryptography-based techniques are substantially unfeasible in this context, since a spectrum observer should possess the private keys exchanged among all the nodes in the network, which is unrealistic.

On the other hand, radio fingerprinting (RFP) has attracted significant attention as reliable and effective spectrum-level authentication technique [8–14]. RFP leverages naturally-occurring circuitry imperfections to compute a unique "fingerprint" of the device directly at the waveform level [15]. Although RFP for physical layer (PHY) Wi-Fi authentication has been explored, existing approaches require software-defined radio (SDR) devices to extract RFP features. This may ultimately prevent widespread adoption, since SDRs require expert knowledge and are usually more expensive than off-the-shelf devices. Moreover, existing work has tackled Wi-Fi fingerprinting up to the legacy 802.11a/g/b standards, which do not support multi-input, multi-output (MIMO) techniques. However, newer Wi-Fi releases such as 802.11ac/ax and the upcoming 802.11be will heavily rely on multi-user MIMO (MU-MIMO) techniques to deliver significantly higher throughput than previous standards [16–18]. Thus, it is still unknown whether existing RFP strategies can be applied in the significantly more complex MU-MIMO scenario, where inter-user interference (IUI) and inter-stream interference (ISI) can significant decrease the quality of the fingerprint itself.

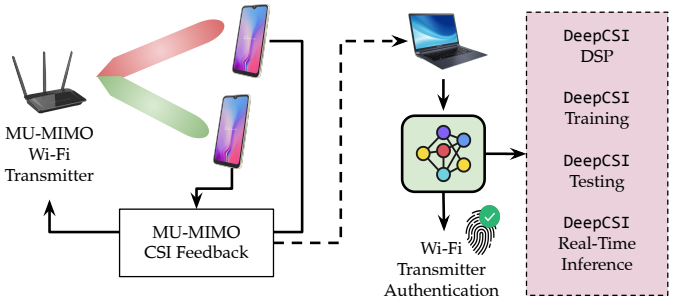


Fig. 1: Main operations of DeepCSI. The beamformer's fingerprint can be independently extracted from the feedback of any beamformee.

To fill the current research gap, in this paper we propose DeepCSI, a brand-new technique for RFP of Wi-Fi devices which is summarized in Fig. 1. The core intuition behind DeepCSI is that the circuitry imperfections in the transmitter's radio interface will percolate onto the MU-MIMO channel state information (CSI) feedback sent by the receiver to the transmitter to perform beamforming. By demodulating this PHY-level information and performing deep learning tech-

2575-8411/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDCS54860.2022.00106

niques on a processed version of the feedback, an observer can fingerprint the transmitter without the need of SDR capabilities. Note that the observer can leverage the feedback from any beamformee associated with the target beamformer to compute the beamformer's fingerprint. The key advantage is that our technique is not affected by ISI nor by IUI – see Section II-A. As pointed our earlier, the effect of IUI and ISI may prevent the correct devices' authentication. We stress that previous work only considered non-MIMO transmissions where IUI/ISI are not present and, in turn, the plain CSI information suffices to perform RFP. On the other hand, the core concern of performing RFP without direct CSI access is that it is unknown whether the imperfections will actually percolate onto the beamforming feedback matrix. In this context, it is crucial to evaluate the PHY fingerprinting techniques as a function of different channels and different transmitter-receiver positions, since these can significantly undermine the fingerprint [14].

To summarize, this paper makes the following contributions:

• We propose `DeepCSI`, the first approach to perform RFP of MU-MIMO Wi-Fi devices. `DeepCSI` uses deep learning of the standard-compliant beamforming matrices to learn the device-unique imperfections located in the CSI and authenticate MU-MIMO Wi-Fi devices directly at the PHY layer. The core intuition is that imperfections in the transmitter's radio circuitry are also present in the beamforming feedback matrix that is transmitted in clear text. Thus, conversely from prior work, explicit CSI computation through SDR technologies are not needed and `DeepCSI` can be run on any low-cost Wi-Fi device. Through `DeepCSI` an observer can leverage the beamforming feedback matrix from any beamformee – *one at a time* – associated with the beamformer to be authenticated. Given the small memory footprint, the trained learning algorithm can be run to perform the online inference on low-cost Wi-Fi devices, e.g., laptops, without the need for powerful facilities.

• We extensively evaluate the performance of `DeepCSI` through a massive data collection campaign performed *in the wild* with off-the-shelf equipment, where 10 Wi-Fi radios emit MU-MIMO signals to multiple receivers located at different positions (and thus, with different beam patterns). Experimental results indicate that `DeepCSI` is able to correctly identify the transmitter with an accuracy above 98%, which shows that RFP of MU-MIMO devices can be performed leveraging the CSI beamforming feedback matrices. We evaluate the impact of the feedback quantization error on the performance – where quantization is applied for transmission efficiency reasons as per the Wi-Fi standards [19, 20] – observing an accuracy increase of up to 63% when changing the feedback PHY parameters. We show that `DeepCSI` achieves at least 17% more accuracy than methods based on CSI phase cleaning, since the latter partially remove the imperfections due to the hardware circuitry. Finally, we evaluate the beamformer identification accuracy on the move where `DeepCSI` achieves an accuracy above 82%. **We pledge to share our code and 800 GB datasets with the community, which will allow replicability and provide a performance benchmark to other researchers in the field [21]**.

II. Background, Related Work, and Challenges

Thanks to their capability of identifying transmitters without the need of computation-hungry cryptography techniques, RFP techniques have received a significant amount of attention from the research community [9–11, 13, 22]. While early work has demonstrated the feasibility of RFP, it has focused on the extraction of complex hand-tailored features, which do not scale well with the device population, or work in ad hoc propagation settings only. Among the first works on Wi-Fi-specific RFP, Vo *et al.* [13] propose RFP techniques that extract features from the scrambling seed, the level of frequency offset and transients between symbols. However, the models achieve accuracy up to 50% on 100 devices. The authors in [9], instead, demonstrated that up to 54 ZigBee devices can be fingerprinted with about 95% accuracy through PSK transients. More recently, Zheng *et al.* [8] studied and evaluated in a testbed of 33 devices a model-based approach to summarize imperfections in the modulation, timing, frequency and power amplifier noise. It is not clear, however, whether the approach in [8] generalizes to different channel environments.

In stark contrast with early work, recent RFP papers have leveraged deep learning techniques to fingerprint wireless devices [12, 23–26]. A key advantage of deep learning techniques is that they are able to perform feature extraction and classification at the same time, thus avoiding manual extraction of device-distinguishing features. For example, Das *et al.* [26] and Merchant *et al.* [25] deep neural networks (DNNs) achieve more than 90% accuracy with a population of 7 ZigBee devices and 30 LoRa devices. To further increase accuracy, [12, 24] proposed the introduction of artificial impairments at the transmitter's side. However, without compensation, this approach inevitably increases the bit error rate (BER). The usage of complex-valued convolutional neural networks (CNNs) has been explored by Gopalakrishnan *et al.* [27], while in [23] and [28] the authors propose the usage of finite impulse response (FIR) filters to compensate for the adverse action of the wireless channel on the fingerprinting accuracy. The key limitation of existing work is that it is entirely based on SDRs, which is very specialized, expensive equipment that is not widely available in common Wi-Fi networks. Moreover, to the best of our knowledge, no prior work has tackled the issue of assessing whether RFP is feasible in MU-MIMO Wi-Fi networks. In this work, we address both issues at once by presenting `DeepCSI`, a framework that (i) can be run on any off-the-shelf Wi-Fi-compliant device, and (ii) can accurately fingerprint MU-MIMO devices. We evaluate the performance of `DeepCSI` in static and – for the first time – dynamic conditions, assessing the robustness of the learned fingerprint to changing transmission channel characteristics.

A. Challenges of MU-MIMO Fingerprinting

Performing RFP of devices operating in downlink (DL) MU-MIMO mode is significantly more challenging than RFP of devices operating with omnidirectional antennas. First, transmissions are inevitably impaired by imperfect beamforming weights that do not accurately compensate the wireless

channel. Secondly, (i) inter-stream interference (ISI) occurs between streams transmitted to the same receiver, and (ii) inter-user interference (IUI) affects streams directed to different receivers. The time-varying behavior of both ISI and IUI complicates the identification of the device-specific imperfections. Moreover, it has been shown in prior work that the RFP process may be adversely impacted by the presence of the wireless channel [12, 14]. This reasoning led us to design a different approach for extracting effective radio fingerprints. Specifically, we use the beamforming feedback matrix $\tilde{\mathbf{V}}$ described in Section III-B. The matrix $\tilde{\mathbf{V}}$ is estimated based on the very high throughput (VHT)-long training fields (LTFs) of the null data packet (NDP) that is sent in broadcast mode without being beamformed. Moreover, the VHT-LTFs are sent over the different antennas in subsequent time slots of 4 μs each. Therefore, the NDP and, in turn, $\tilde{\mathbf{V}}$, *are not affected by IUI nor by ISI*. However, since the feedback matrix is *quantized* before transmission, quantization errors are inevitable. In Section V, we analyze the effect of the quantization error and investigate the generalization capability of our RFP approach to multiple channels and beamformee positions, and to beamformer's mobility.

III. The DeepCSI Framework

Henceforth, we will adopt the following notation for mathematical expressions. The superscripts T and $\dagger$ respectively denote the transpose and the Hermitian of a matrix, i.e., the complex conjugate transpose. By $\angle\mathbf{C}$, we refer to the matrix whose elements are the phases of the corresponding elements in the complex-valued matrix $\mathbf{C}$. $\text{diag}(c_1, \ldots, c_j)$ indicates the diagonal matrix with elements $(c_1, \ldots, c_j)$ on the main diagonal. The (c_1, c_2) entry of matrix $\mathbf{C}$ is denoted by $[\mathbf{C}]_{c_1,c_2}$. Finally, $\mathbf{I}_c$ refers to a $c \times c$ identity matrix while $\mathbf{I}_{c\times d}$ is a $c \times d$ matrix with ones on the main diagonal and zeros elsewhere.

A. Preliminaries on MU-MIMO in Wi-Fi

In the following, we will consider Wi-Fi devices operating with the IEEE 802.11ac (Wi-Fi 5) standard and 802.11ax (WiFi 6 and 6E) [19, 20]. These devices operate on the 2.4 GHz, 5 GHz and 6 GHz frequency bands with channels having up to 160 MHz of bandwidth. In Wi-Fi, data is transmitted via orthogonal frequency-division multiplexing (OFDM) by dividing the selected channel into K partially overlapping and orthogonal sub-channels, spaced apart by $1/T$. The input bits are grouped into OFDM samples, a_k, and symbols, $\mathbf{a} = [a_{-K/2}, \ldots, a_{K/2-1}]$, collecting K samples each. After being digitally modulated, the K samples of one OFDM symbol are simultaneously transmitted though the K OFDM sub-channels, occupying the channel for T seconds. Up-converted to the carrier f_c, the transmitted signal is

$$s_{\text{tx}}(t) = e^{j2\pi f_c t} \sum_{k=-K/2}^{K/2-1} a_k e^{j2\pi kt/T}. \tag{1}$$

To improve the signal-to-noise ratio (SNR), the transmitter can use *beamforming* to focus the power toward the intended receiver. The beamforming may also compensate the effect of the wireless channel from the transmitter (*beamformer*) to the receiver (*beamformee*). When both devices in the communication link are equipped with antenna arrays (MIMO system), each pair of transmitter and receiver antennas forms a physical channel that can be exploited for wireless communication. This spatial diversity allows shaping multiple beams, referred to as *spatial streams*, to transmit different signals to the beamformee, in a parallel fashion. To this end, the signals are combined at each transmitter antenna through steering weights, $\mathbf{W}$, derived from the channel frequency response (CFR) matrix $\mathbf{H}$. The CFR needs to be estimated for every OFDM sub-channel over each pair of transmitter (TX) and receiver (RX) antennas, thus obtaining a $K \times M \times N$ matrix, where M and N are respectively the number of TX and RX antennas. In Fig. 2 we report an example of beamforming for a 3×2 MIMO system. At the beamformee side, the original signals are retrieved from their combination exploiting the fact that, ideally, $[\mathbf{H}]_{\bar{\ell},\bar{i}}[\mathbf{W}]_{\ell,i} = 0$ when $\bar{\ell} \neq \ell$ or $\bar{i} \neq i$.

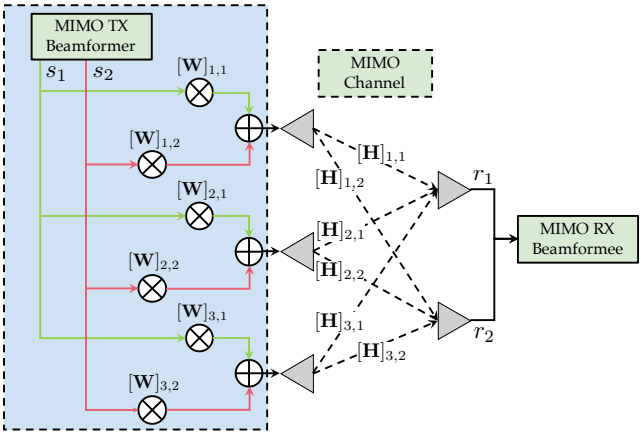


Fig. 2: 3×2 MIMO system. The grey triangles represent the antennas. s_1, s_2 and r_1, r_2 stand for the transmitted and received signals respectively. $\mathbf{W}$ is the steering matrix containing the weights to shape the beams. $\mathbf{H}$ is the CFR.

A meaningful model for the CFR $\mathbf{H}$ in indoor spaces is obtained by considering the proprieties of the wireless propagation. After being irradiated by the transmitter antenna $m \in \{0, \ldots, M-1\}$, the signal is reflected by objects in the environment and, in turn, P different copies of $s_{\text{tx}}(t)$ are collected at the receiver antenna $n \in \{0, \ldots, N-1\}$. Each received signal is characterized by an attenuation A_p and a delay τ_p that depends on the length of the path followed by the transmitted wave. Thus, the (k, m, n) element of $\mathbf{H}$ is

$$[\mathbf{H}]_{k,m,n} = \sum_{p=0}^{P-1} A_{m,n,p} e^{-j2\pi(f_c + k/T)\tau_{m,n,p}}. \tag{2}$$

By knowing $\mathbf{H}$, the beamformer can generate the steering matrix $\mathbf{W}$ to maximize the power sent toward the beamformee or simultaneously send parallel data streams to multiple beamformees. These communication modes are respectively referred to as single-user MIMO (SU-MIMO) and MU-MIMO. While IEEE 802.11n only supports SU-MIMO mode, in 802.11ac and above MU-MIMO can be enabled in the DL direction, i.e., at the access point (AP) side [19]. In 802.11ax MU-MIMO can be also enabled in the uplink (UL) [20].

B. Compressed Beamforming Feedback

In IEEE 802.11ac/ax, DL MU-MIMO is enabled by the *pre-coding* and the *channel sounding* procedures [19]. Pre-coding linearly combines the signals to be simultaneously transmitted to the different beamformees. This procedure shapes the beams focusing the power in the correct directions. The combination weights are antenna-specific and are computed based on channel sounding performed through a NDP, transmitted without beamforming. After receiving the NDP, each beamformee estimates $\mathbf{H}$ based on a VHT-LTF for each spatial stream. Next, the beamformee feeds back the matrix to the beamformer in the form of a *compressed beamforming feedback*, which is computed for each sub-channel k as follows.

Let $\mathbf{H}_k$ be the $M \times N$ sub-matrix of $\mathbf{H}$ containing the CFR samples (see Eq. (2)) related to sub-channel k. $\mathbf{H}_k$ is first decomposed via singular value decomposition (SVD):

$$\mathbf{H}_k^T = \mathbf{U}_k \mathbf{S}_k \mathbf{Z}_k^\dagger, \tag{3}$$

where $\mathbf{U}_k$ and $\mathbf{Z}_k$ are, respectively, $N \times N$ and $M \times M$ unitary matrices, while $\mathbf{S}_k$ is an $N \times M$ diagonal matrix collecting the singular values. Next, the first $N_{\text{SS}} \leq N$ columns of $\mathbf{Z}_k$ are extracted to form the complex-valued beamforming matrix $\mathbf{V}_k$ that is used by the beamformer to compute the pre-coding weights for the N_{SS} spatial streams directed to the beamformee. Note that the beamformee can be served with at maximum $N_{\text{SS}} = N$ spatial streams (see Chapter 13 of [29]). Thus, the beamformee is required to send back $\mathbf{V}_k$ to the beamformer. To do that efficiently, instead of sending the complete matrix, the beamformee derives and transmits its *compressed representation*. Specifically, the feedback is a number of angles obtained by converting $\mathbf{V}_k$ into polar coordinates. The transformation is based on the procedure in Algorithm 1, where $\mathbf{D}_{k,i}$ and $\mathbf{G}_{k,\ell,i}$ are defined as

$$\mathbf{D}_{k,i} = \begin{bmatrix} \mathbf{I}_{i-1} & 0 & \dots & 0 \\ 0 & e^{j\phi_{k,i,i}} & 0 & \dots \\ \vdots & 0 & \ddots & 0 \\ & \vdots & 0 & e^{j\phi_{k,M-1,i}} & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}, \tag{4}$$

$$\mathbf{G}_{k,\ell,i} = \begin{bmatrix} \mathbf{I}_{i-1} & 0 & \dots & 0 \\ 0 & \cos\psi_{k,\ell,i} & 0 & \sin\psi_{k,\ell,i} & \vdots \\ \vdots & 0 & \mathbf{I}_{\ell-i-1} & 0 & \\ & -\sin\psi_{k,\ell,i} & 0 & \cos\psi_{k,\ell,i} & 0 \\ 0 & \dots & 0 & \mathbf{I}_{M-\ell} \end{bmatrix}. \tag{5}$$

The obtained matrices allows rewriting $\mathbf{V}_k$ as

$$\mathbf{V}_k = \tilde{\mathbf{V}}_k \tilde{\mathbf{D}}_k, \tag{6}$$

with

$$\tilde{\mathbf{V}}_k = \prod_{i=1}^{\min(N_{\text{SS}}, M-1)} \left(\mathbf{D}_{k,i} \prod_{l=i+1}^{M} \mathbf{G}_{k,l,i}^T \right) \mathbf{I}_{M \times N_{\text{SS}}}, \tag{7}$$

where the products represent matrix multiplications. Note that, by construction, the last row of the complex-valued $\tilde{\mathbf{V}}_k$ matrix, i.e., the feedback for the M-th transmitter antenna, consists of non-negative real numbers. Next, the $K \times M \times N_{\text{SS}}$ beamforming matrix $\tilde{\mathbf{V}}$ is obtained by stacking the $\tilde{\mathbf{V}}_k$ matrices for $k \in \{-K/2, \dots, K/2 - 1\}$. Thanks to this transformation, the beamformee is only required to transmit the ϕ and ψ angles from which the $\tilde{\mathbf{V}}_k$ matrices can be reconstructed. The beamforming performance is equivalent at the beamformee when using $\mathbf{V}_k$ or $\tilde{\mathbf{V}}_k$ to construct the steering matrix $\mathbf{W}$ and, in turn, the feedback for $\tilde{\mathbf{D}}_k$ is not sent [29].

Algorithm 1 $\mathbf{V}_k$ matrix decomposition

Input: $\mathbf{V}_k$
Output: $\mathbf{D}_{k,i}$ and $\mathbf{G}_{k,\ell,i}$ for $i \in \{1, \dots, \min(N_{\text{SS}}, M-1)\}$, $\ell \in \{i+1, \dots, M\}$

$\tilde{\mathbf{D}}_k = \text{diag}(e^{j\angle[\mathbf{V}_k]_{M,1}}, \dots, e^{j\angle[\mathbf{V}_k]_{M,N_{\text{SS}}}})$
$\boldsymbol{\Omega}_k = \mathbf{V}_k \tilde{\mathbf{D}}_k^\dagger$
for $i \leftarrow 1$ to $\min(N_{\text{SS}}, M-1)$ **do**
  $\phi_{k,\ell,i} = \angle[\boldsymbol{\Omega}_k]_{\ell,i}$ with $\ell = i, \dots, M-1$
  compute $\mathbf{D}_{k,i}$ through Eq. (4)
  $\boldsymbol{\Omega}_k \leftarrow \mathbf{D}_{k,i}^\dagger \boldsymbol{\Omega}_k$
  for $\ell \leftarrow i+1$ to M **do**
    $\psi_{k,\ell,i} = \arccos\left(\frac{[\boldsymbol{\Omega}_k]_{i,i}}{\sqrt{[\boldsymbol{\Omega}_k]_{i,i}^2 + [\boldsymbol{\Omega}_k]_{\ell,i}^2}} \right)$
    compute $\mathbf{G}_{k,\ell,i}$ through Eq. (5)
    $\boldsymbol{\Omega}_k \leftarrow \mathbf{G}_{k,\ell,i} \boldsymbol{\Omega}_k$

The angles are quantized for transmission using $b_\phi \in \{7, 9\}$ bits for ϕ and $b_\psi = b_\phi - 2$ bits for ψ. Next, the quantized values are packed into the VHT compressed beamforming frame and transmitted without encryption, thus allowing any device that can access the wireless channel to capture the information sent by the beamformee to the beamformer. The b_ϕ and b_ψ values can be read in the VHT MIMO control field of the frame, together with other information including the number of columns (N_{SS}) and rows (M) in the beamforming matrix and the channel bandwidth. At the beamformer, the ϕ and ψ angles are retrieved from their quantized versions $q_\phi = \{0, \dots, 2^{b_\phi} - 1\}$ and $q_\psi = \{0, \dots, 2^{b_\psi} - 1\}$ using

$$[\phi, \psi] = \left[\pi \left(\frac{1}{2^{b_\phi}} + \frac{q_\phi}{2^{b_\phi - 1}} \right), \pi \left(\frac{1}{2^{b_\psi + 2}} + \frac{q_\psi}{2^{b_\psi + 1}} \right) \right] \tag{8}$$

C. `DeepCSI` Workflow and Learning Architecture

Fig. 3 summarizes how `DeepCSI` leverages the sounding protocol mechanism described in Section III-B to obtain a fingerprint of the IEEE 802.11ac/ax AP (beamformer). The sounding is triggered by the beamformer before sending data in DL MU-MIMO mode to the beamformees, and concludes with the transmission of the feedback angles. `DeepCSI` exploits the fact that the angles can be easily collected by any Wi-Fi compliant device by setting the Wi-Fi interface in monitor mode and using a network analyzer toolkit, e.g., Wireshark [30], to capture the packet containing the feedback. Notice that `DeepCSI` does not require the monitor device to be authenticated with the target AP. Once obtained the feedback angles, `DeepCSI` reconstructs $\tilde{\mathbf{V}}$ through Eq. (7). Next, the beamforming feedback matrix is used as input for

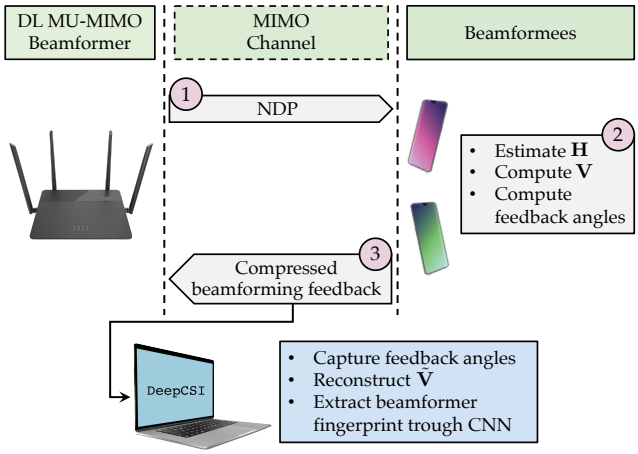


Fig. 3: DeepCSI workflow. The compressed beamforming feedback computed by any of the beamformees as the final step of the sounding protocol is leveraged by DeepCSI to obtain a fingerprint of the beamformer.

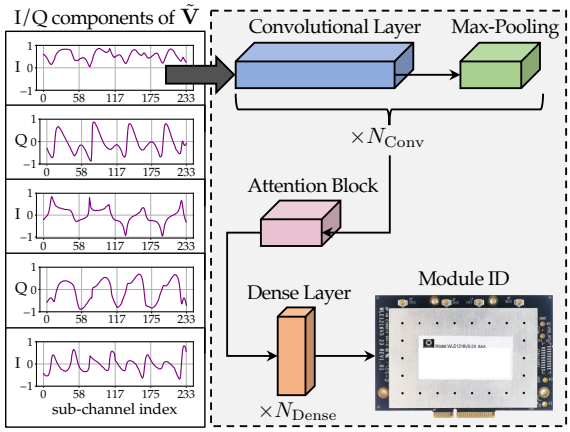


Fig. 4: DeepCSI learning algorithm. The I and Q components of $\tilde{\mathbf{V}}$ serves as input for a neural network classifier that computes the beamformer fingerprint and returns, as output, the estimated Wi-Fi module ID.

the DNN classifier depicted in Fig. 4 to extract the RFP of the beamformer. Once trained, the DNN can be deployed and utilized in real time for device authentication at the PHY level. The observer can leverage the feedback from *any beamformee* associated with the target beamformer to compute a beamformer's fingerprint. In turn, DeepCSI is independent of the number of terminals connected to the AP. Moreover, different fingerprints can be obtained for the same beamformer and can be indifferently used to authenticate the device.

The elements of the feedback matrix are fed to the DNN as follows. The I/Q components of the beamforming feedback are stacked into an $N_{\text{row}} \times N_{\text{col}} \times N_{\text{ch}}$ matrix, where $N_{\text{col}} \leq K$ identifies the number of selected OFDM sub-channels, $N_{\text{row}} \leq N_{\text{SS}}$ and $N_{\text{ch}} < 2M$ refer to the columns and rows of $\tilde{\mathbf{V}}$ used for fingerprinting and the 2-factor is for the I/Q components. Note that the feedback for the last transmitting antenna consists of the sole I information as, by construction, the last row of each $\tilde{\mathbf{V}}_k$ (Eq. (7)) is composed of non-negative real values [19]. The learning architecture is inspired from [31] and consists of a series of N_{conv} convolutional layers followed by selu activation function [32], and by a max-pooling layer. The output of the previous block (in blue and green in Fig. 4) is forward through an attention block and – after being flattened – is processed by N_{dense} dense layers with selu activation function. A final dense layer with softmax activation is used for classification. Alpha-dropout layers are interposed between the dense layers. The attention block is inspired by the spatial attention module in [33]. First, the maximum and the average feature maps are obtained by computing respectively the maximum and the mean of the input feature maps over the channel dimension. Next, the two maps are concatenated and forwarded through a convolutional layer with sigmoid activation function that outputs the weights to attend the input feature maps. Specifically, the attention operation consists in multiplying the input by the computed weights. A skip connection is also implemented by summing the output of the attention block with its input before passing the result to the subsequent dense layers. Thanks to the attention block, the algorithm learns where the most relevant information is located within the feature maps. This allows the network to focus on the relevant regions obtaining a more effective fingerprint.

We performed hyper-parameter evaluation in Section V, and established through experiments that a good set of hyper parameters is $N_{\text{conv}} = 5$ with 128 filters each, and $N_{\text{dense}} = 2$ dense layers with 128 and 64 neurons each. This architecture yields a DNN containing 489,301 trainable parameters, which is relatively small compared to state-of-the-art DNNs. The DeepCSI learning algorithm is trained in an offline fashion by back propagating the cross-entropy loss between the module identifier (ID) predicted by the classifier and the actual one.

IV. Experimental Setup

We evaluate the effectiveness of DeepCSI using off-the-shelf devices and through extensive experimental evaluation. To this end, we set up a Wi-Fi network consisting of one AP (beamformer) and two stations (STAs) (beamformees). The AP was implemented through a Gateworks GW6200 single board computer (SBC) equipped with a Compex WLE1216v5-23 IEEE 802.11ac module, as shown in Fig. 5. Two Netgear Nighthawk X4S AC2600 routers, with $N \in \{1, 2\}$ out of 4 antennas enabled, acted as STAs (beamformees). At the AP, $M = 3$ antennas were used to sound the channel for DL MU-MIMO transmission mode and the STAs were served with $N_{\text{SS}} \in \{1, 2\}$ spatial streams each. Note that implementation specific constraints prevent the use of $M = 4$ for DL MU-MIMO. For the data transmission between the AP and the STAs, we used channel 42, i.e., $f_c = 5.21$ GHz with 80 MHz bandwidth. The number of OFDM sub-channels sounded is $K = 234$ as the mechanism does not consider the 14 control sub-channels and the 8 pilot ones. The AP uses the quantization parameters $b_\phi = 9$ and $b_\psi = 7$ for ϕ and ψ feedback angles, respectively. We generated UDP traffic in the DL direction to induce the AP to trigger the channel sounding mechanism, and collected the angles (ϕ, ψ) that were sent back by the beamformees using the Wireshark network analyzed toolkit [30] running on an off-the-shelf laptop equipped with

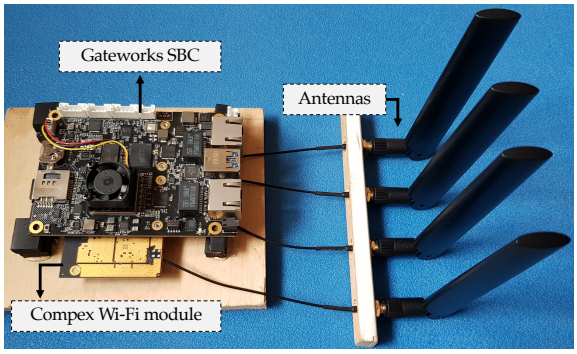


Fig. 5: DL MU-MIMO enabled Wi-Fi AP (beamformer). The Compex WLE1216v5-23 Wi-Fi module was mounted on a Gateworks GW6200 SBC platform. Four antennas were connected to the Wi-Fi module.

an IEEE 802.11ac Wi-Fi card. This allows retrieving the $\tilde{\mathbf{V}}$ matrices associated with each sounding operation, and computing the beamformer fingerprint (see Section III-C).

Two datasets – namely `D1` and `D2` – were collected. As for the former, the STAs were deployed at different positions as depicted in Fig. 6 to generate different beam patterns and different SNR regimes. The number of enabled antennas is $N = 2$ for each beamformee and each of them is served with $N_{\text{SS}} = 2$ spatial streams. Dataset `D1` allows evaluating the performance of `DeepCSI` in different static conditions. The purpose of dataset `D2` is to evaluate the impact of mobility in the beamformer identification. The data were collected while the AP was manually moved following the path described in Fig. 6 by the yellow stars A-B-C-D-B-A, entailing both vertical and horizontal movements. Here, $N = N_{\text{SS}} = 1$ for the first beamformee and $N = N_{\text{SS}} = 2$ for the second. The datasets were collected in two different indoor environments reproducing the same configuration depicted in Fig. 6. This allows evaluating the general applicability of the developed algorithm in recognizing the beamformer in the wild.

We pledge to share our datasets with the community for reproducibility and benchmarking purposes [21].

A. Datasets Structure

The datasets consist of the beamforming feedback angles associated with $N_{\text{modules}} = 10$ different Compex Wi-Fi modules, which are the target of the proposed fingerprinting mechanism. They are collected in two indoor environments where the three entities constituting the experimental Wi-Fi network are placed as shown in Fig. 6 and no obstacles are present between the AP and the STAs. At the AP, the SBC, the antennas and the coaxial cables remain the same across all the considered network setups, by only changing the Compex Wi-Fi module. This ensures that the fingerprint procedure only relies on the hardware imperfections of the Wi-Fi module.

For the static dataset `D1`, we collected 9 different measurements for each Compex module by keeping it fixed in position A and changing the positions of the STAs. Specifically, the beamformees are first placed in front of the beamformer, i.e., with an angle of arrival (AoA) for the direct path of nearly zero degrees, and next moved of multiples of 10 cm respectively to the left and to the right with respect to their

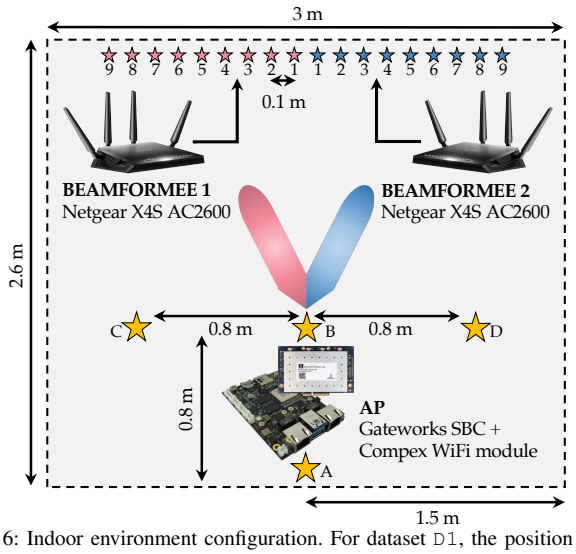


Fig. 6: Indoor environment configuration. For dataset `D1`, the position of the AP remains the same for all the acquisitions (yellow star A). The beamformees are first placed in front of the AP and next, for each new experiment, beamformees 1 and 2 are respectively moved 10 cm to the left and 10 cm to the right. The subsequent positions of the beamformees are marked with red and blue stars respectively and labeled with a number $\in \{1, \ldots, 9\}$. For the dynamic dataset `D2`, the beamformees remain fixed in position 3 while the AP moves following the path described by the yellow stars A-B-C-D-B-A.

initial position (see the colored stars in Fig. 6). The positions of the STAs are maintained fixed for the entire duration of each measurement. These configurations allow obtaining data associated with different beam shaping for the ongoing DL MU-MIMO transmissions. Overall, we collected 90 traces, i.e., 9 traces for each of the 10 Compex Wi-Fi modules.

As for the dynamic dataset `D2`, we collected 11 measurements for each Compex module. Four measurements are collected with the AP fixed in position A. The remaining seven traces are collected while moving the AP following the path described above, i.e., first, the AP is moved 80 cm from position A toward the beamformees reaching position B, next is shifted 80 cm to the left and subsequently 160 cm to the right – up to positions C and D respectively – and finally it is brought back in position A passing from B. The beamformees are kept fixed in position 3. This dataset allows evaluating the performance of `DeepCSI` in the presence of beamformer mobility. Overall, it consists of 11 traces for each of the 10 Compex Wi-Fi modules for a total of 110 traces.

Each trace contains the feedback angles sent by the two beamformees during two minutes of transmission. Such feedbacks can be promptly grouped based on the beamformee identifier by applying a filter on the packets source address.

B. `DeepCSI` Training and Testing Procedure

The `DeepCSI` classifier (see Fig. 4) was trained using different PHY configurations, to evaluate its robustness in correctly identifying the beamformer device (the AP) as the position of the beamformees change – dataset `D1` – and when the beamformer moves within the environment – dataset `D2`.

Table I summarizes the different training/testing sets that were considered for dataset `D1`, where the beamformees positions are depicted in Fig. 6. When the same positions are

set	beamformees position																
	training									testing							
	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
S1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S2	■		■		■	■		■		■		■		■		■		■
S3	■	■	■	■	■										■	■	■	■

TABLE I: Dataset D1, training/testing sets to assess the DeepCSI performance when varying the beamformees position in $\{1, \ldots, 9\}$ (see Fig. 6).

set	measurement group identifier							
	training				testing			
	fix1	fix2	mob1	mob2	fix1	fix2	mob1	mob2
S4			■					■
S5	■	■					■	■
S6			■	■	■	■		

TABLE II: Dataset D2, training/testing sets. 'Fix1' and 'fix2' group two static traces each, i.e., the AP is fixed in position A (see Fig. 6). 'Mob1' and 'mob2' contain respectively four and three mobility traces, i.e., collected while the AP is manually moved following the path detailed in Fig. 6.

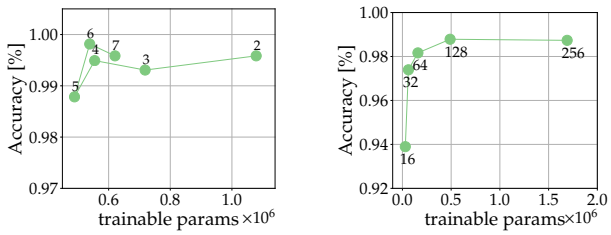
considered in the training and testing phase, the first 80% of the collected data is used for training and validating the model, while the remaining 20% serves as test data. In all cases, the last 20% of training data is used for model validation. As part of our evaluation, we also assess the performance of DeepCSI on $\tilde{\mathbf{V}}$ sub-matrices. This makes it possible to evaluate the impact of using (i) different groups of transmitter antennas and spatial streams, and (ii) different portions of the radio spectrum. For (i), we vary N_{ch} and N_{row}. For (ii), we pick a subset of the K available sub-channels.

The training/test sets considered for dataset D2 are detailed in Table II. For ease of readability, we combined the eleven traces composing the dataset into four groups. 'Fix1' and 'fix2' collect the four traces – two traces each – acquired keeping fixed the position of the AP. The mobility traces – i.e., collected while the AP is manually moved in the environment – are grouped in 'mob1' and 'mob2', where the first group contains four measurements while the remaining three traces compose the second group. Note that the mobility traces encode variations associated with the manual movement of the AP. This implies that the positions taken by the AP during the acquisition of the traces are *approximately* the same due to slight variations in the movements. Moreover, a person is always present in the proximity of the AP to perform the operation, introducing additional variability.

For each configuration, DeepCSI is independently trained on the feedbacks from the two beamformees, obtaining one model for each of them. In this way, we evaluate a realistic usage scenario where each beamformee authenticates the beamformer based on local information, without relying on some other, possibly malicious, entities. The results considering both the beamformees are also reported for completeness.

V. Experimental Results

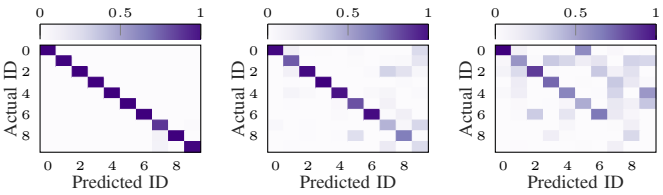
DeepCSI was experimentally evaluated on the Wi-Fi network setups of Tables I and II, assessing the effectiveness of the extracted beamformer fingerprint for different beamformer and beamformees configurations. We first briefly discuss the



(a) DeepCSI accuracy by varying the number of convolutional layers, with 128 filters each, from 2 to 7.

(b) DeepCSI accuracy by using 5 convolutional layers and varying the number of filters in each layer, from 16 to 256.

Fig. 7: DeepCSI accuracy for beamformer 1, on S1 validation data, by varying the DNN parameters.



(a) S1. Accuracy: 98.02%

(b) S2. Accuracy: 75.41%

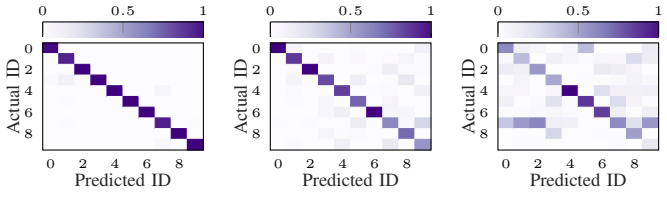
(c) S3. Accuracy: 42.97%

Fig. 8: Confusion matrices for beamformee 1, 3 TX antennas, spatial stream 0. ID in this and in the following plots refers to the AP Wi-Fi module identifier.

DNN hyper parameters selection process and then present the DeepCSI performance by varying the PHY parameters of the MU-MIMO transmission mode. In the first part, the DeepCSI performance are assessed on dataset D1, evaluating the effect of the beamformees' positions. Dataset D2 is considered in the second part to analyze the impact of the beamformer mobility on the device identification accuracy.

DeepCSI hyper parameters selection. Fig. 7a and Fig. 7b respectively evaluate the effect of tuning the number of convolutional layers and filters for the DNN presented in Section III-C. Noticeably, the accuracy remains almost constant when varying the number of layers. Also, the accuracy increases with an increasing number of filters, at a cost of a higher network complexity (i.e., more trainable parameters). As a trade-off between accuracy and complexity, we selected $N_{\text{conv}} = 5$ convolutional layers with 128 filters each and kernel sizes of (1, 7) for the first three layers, (1,5) for the fourth and (1,3) for the last one by using the elbow method [34]. The max-pooling kernels are set to (1, 2) and the alpha-dropout between the three dense layers is applied with retain probability of 0.5 and 0.2, respectively.

DeepCSI performance using different beamformees configurations. Fig. 8 shows the accuracy of DeepCSI in correctly identifying the beamformer among the 10 Compex Wi-Fi modules in the dataset. The results were obtained using the beamforming feedback angles from a single beamformee. The confusion matrices are reported for each of the three training/testing configurations in Table I, where ID refers to the AP module identifier. We notice that the accuracy increases with more spatial diversity in the training data, reaching 98.02% when all the configurations are used at training time (see Fig. 8a for set S1). With sets S2 and S3, the beamformee positions at training and testing times differ. The lowest accuracy is obtained with S3 (worst-case configuration). This



(a) S1. Accuracy: 97.62% (b) S2. Accuracy: 77.38% (c) S3. Accuracy: 47.28%

Fig. 9: Confusion matrices, mixed beamformees, 3 TX ant., spatial stream 0.

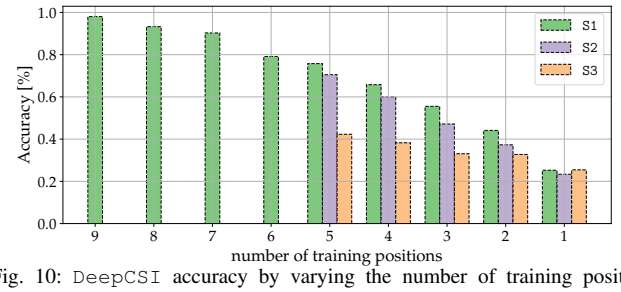
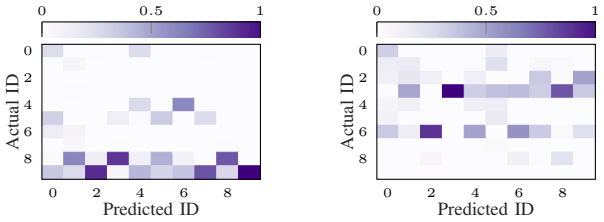


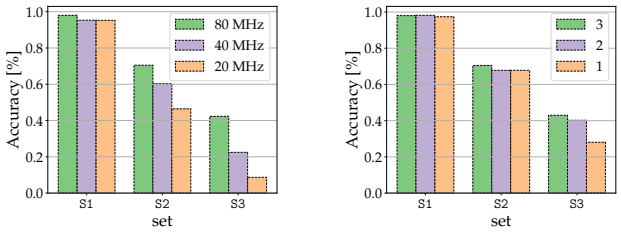
Fig. 10: DeepCSI accuracy by varying the number of training positions from the considered set (see Table I). Set S1 is trained on a maximum of 9 beamformee positions while S2 and S3 on 5.

is because S3 is the set with the largest difference between training and testing positions. The performance improves when going from S3 to S2, as the latter provides DeepCSI with a more balanced set of positions during training, allowing the classifier to fill the knowledge gaps by "interpolating" the patterns learned from adjacent positions. The network reuses information from similar beam patterns leading to an identification accuracy of 75%, even when the beamformee is at a position that was not contained in the training set (see Fig. 8b). The same applies to Fig. 9, where the beamforming feedback angles of both beamformees are used to build the training set. This allows to slightly increase the DeepCSI accuracy on sets S2 and S3. However, using this technique in real-world scenarios poses security concerns associated with the reciprocal trustworthiness of the beamformees in a Wi-Fi network. The impact of the number of beamformee training positions is evaluated in Fig. 10. We report the accuracy obtained by increasing the number of positions used at training time from 1 to 9 for set S1 and from 1 to 5 for sets S2 and S3, according to Table I. In all the cases, the accuracy increases with more beamformee positions in the training data, which confirms that the fingerprint is more effective when high spatial diversity is present in the training data. In Fig. 11, we evaluate the effect of swapping the beamformees used at training and testing times for the same network configuration. We trained DeepCSI with data from a given beamformee and used the trained DNN model to identify the AP module from the $\tilde{\mathbf{V}}$ matrices computed by a different beamformee (for the same AP module). The learned fingerprint in this case performs poorly as matrix $\tilde{\mathbf{V}}$ captures hardware inaccuracies of both devices, i.e., the beamformer (the AP) *and* the beamformee. While a well designed learning architecture can identify with high accuracy the beamformer when the beamformee remains the same at training and testing times, it hardly succeeds when these devices differ. We reasonably believe that in a real-world scenario the impact of this will



(a) Training on beamformee 1 and testing on beamformee 2. Accuracy: 25.86% (b) Training on beamformee 2 and testing on beamformee 1. Accuracy: 25.02%

Fig. 11: Confusion matrices for set S1, training on one beamformee and testing on the other, 3 TX antennas, spatial stream 0.



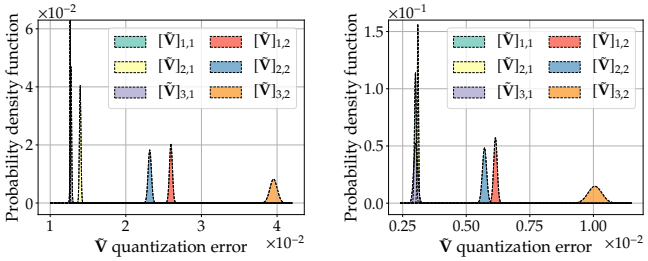
(a) DeepCSI accuracy by varying the channel bandwidth, i.e., selecting respectively $N_{\text{col}} = 234, 110, 54$, out of the $K = 234$ OFDM sub-channels. (b) DeepCSI accuracy by varying the number of transmitter antennas, i.e., selecting respectively $N_{\text{ch}} = 3, 2, 1$ rows of the beamforming feedback matrix $\tilde{\mathbf{V}}$.

Fig. 12: DeepCSI accuracy by varying the channel bandwidth and the number of transmitter antennas, using spatial stream 0.

be even stronger, as the beamformees can be from different vendors and have different hardware configurations.

DeepCSI performance when varying the beamformer transmission parameters. In Fig. 12a we compare the accuracy of DeepCSI when considering different portions of the radio spectrum. According to the IEEE 802.11ac OFDM channels specifications [19], from the 234 sub-channels on an 80 MHz channel, we can group sub-channels belonging to two 40 MHz and four 20 MHz channels. Therefore, from the data collected on the IEEE channel 42 at 80 MHz, we extracted 110 sub-channels for the 40 MHz channel 38 and 54 sub-channels for the 20 MHz channel 36, and assessed the performance of DeepCSI on these subsets. These results prove that the accuracy increases with a larger bandwidth, especially when considering the most challenging configurations S2 and S3. Fig. 12b evaluates the impact of increasing from 1 to 3 the number of transmitter antennas used to compute the fingerprint. Note that the accuracy mainly depends on the number of selected antennas and only weakly depends on their IDs. Thus, we only show results for a single selection pattern out of the possible ones for each number of antennas. The DeepCSI performance remains almost constant on set S1, while the accuracy increases on S2 and S3 going from 1 to 3 transmitter antennas. These results confirm that exploiting to the maximum extent the spatial diversity at the beamformer – by considering all the OFDM sub-channels and transmitter antennas – is key to designing robust RFP algorithms.

DeepCSI performance when changing the reference beamformee spatial stream. To evaluate the effect of changing the DNN input spatial stream on the beamformer fingerprinting accuracy, we consider the impact of the beamforming feedback angles quantization on the columns of $\tilde{\mathbf{V}}$, representing the



(a) $b_\psi = 5$ and $b_\phi = 7$ bit. (b) $b_\psi = 7$ and $b_\phi = 9$ bit.

Fig. 13: PDF of the $\tilde{\mathbf{V}}$ quantization error using the two standard-compliant sets of values for the beamforming feedback angles quantization bits.

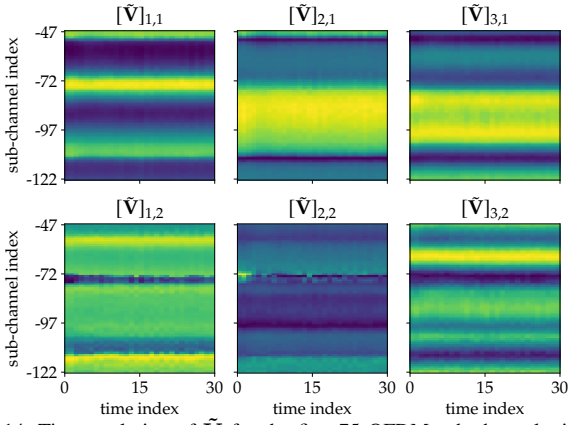
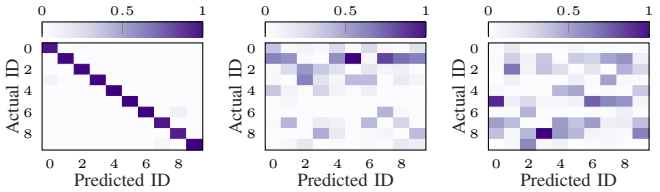


Fig. 14: Time evolution of $\tilde{\mathbf{V}}$ for the first 75 OFDM sub-channels, in static conditions. The columns refer to the transmit antennas while the rows to the spatial streams.

spatial streams dimensions. From Algorithm 1, it follows that the impact of the quantization error increases going from the first to the last reconstructed stream. We verified this fact by simulating an OFDM MU-MIMO channel, considering the ray tracing model of [35]. We obtained the channel matrix $\mathbf{H}$ for 100,000 transmissions in MU-MIMO mode, and we derived $\tilde{\mathbf{V}}$ via SVD. Hence, we computed the q_ϕ and q_ψ quantized angles following Algorithm 1 and using the quantization parameters defined in the standards [19, 20]. These operations are the same performed by the beamformees to generate the feedback. Next, we reconstructed $\tilde{\mathbf{V}}$ from the quantized angles and evaluated the reconstruction error on each combination of transmitter antennas and spatial streams. We plot the probability density functions (PDFs) of the quantization error using ($b_\psi = 5$, $b_\phi = 7$) and ($b_\psi = 7$, $b_\psi = 9$) bits for quantization in Figs. 13a and 13b. We notice the reconstruction of the second column of $\tilde{\mathbf{V}}$, i.e., the second stream, is less accurate than the reconstruction of the first, for all the three transmitter antennas. This is intrinsically related to the construction of the $\mathbf{D}_{k,i}$ and $\mathbf{G}_{k,\ell,i}$ matrices from the quantized angles, and to their combination for the computation of $\tilde{\mathbf{V}}$ (see Eq. (7)). Indeed, the algorithm has a recursive structure by which the quantization error on the first stream propagates to the next ones, leading to worse approximations for the higher order columns of matrix $\tilde{\mathbf{V}}$. The quantization error can also be visualized from our empirical measurements. In Fig. 14, we plot an excerpt of the $\tilde{\mathbf{V}}$ matrix reconstructed by `DeepCSI` from the quantized angles obtained at the beamformee side



(a) `S1`. Accuracy: 97.03% (b) `S2`. Accuracy: 13.32% (c) `S3`. Accuracy: 5.63%

Fig. 15: Confusion matrices, beamformee 1, 3 TX antennas, spatial stream 1.

in static conditions. The quantization error is clearly visible for the second spatial stream (column 2 of matrix $\tilde{\mathbf{V}}$). Thus, the performance of `DeepCSI` decreases when considering as DNN input the data associated with the second spatial stream (Fig. 15) instead of the first one (Fig. 8). While on set `S1` the beamformer can still be identified with high accuracy using data from the second spatial stream, when considering sets `S2` and `S3` – thus reducing the number of training positions – the beamformer fingerprint can no longer be effectively extracted, leading to a considerable drop in the classification accuracy.

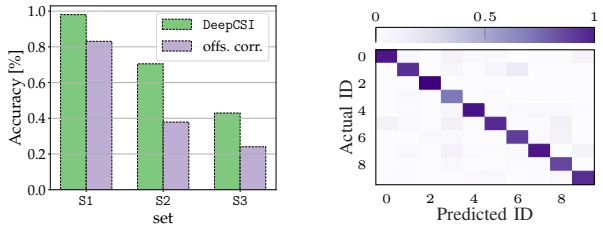
`DeepCSI` performance compared with learning from a processed input. `DeepCSI` learns beamformer-specific features directly from the I/Q samples of matrix $\tilde{\mathbf{V}}$. As an alternative approach, we evaluated the effect of pre-processing such I/Q data before using it as input for the DNN. Specifically, we applied to the beamforming feedback matrices the data cleaning algorithm presented in [36]. The CFR estimated at the beamformee on the NDP – and from which $\tilde{\mathbf{V}}$ is derived – slightly deviates from the theoretical model in Eq. (2) due to hardware imperfections causing undesired phase offsets [37]. Among these imperfections, the most significant are: (i) the carrier frequency offset (CFO), which originates from the difference between the carrier frequency at transmitter and receiver sides; (ii) the sampling frequency offset (SFO), which is due to clocks synchronization error; (iii) the packet detection delay (PDD), i.e., the receiver decoding time; (iv) the phase-locked loop offset (PPO), which is associated with the random generation of the initial signal phase by the phase-locked loop module; and (v) the phase ambiguity (PA), which accounts for the phase difference (multiples of π) among the signals at the transmitter antennas. By considering these contributions, the overall phase offset, $\theta_{\text{offs},k,m,n}$, can be formulated as

$$\theta_{\text{offs},k,m,n} = \theta_{\text{CFO}} - 2\pi k(\tau_{\text{SFO}} + \tau_{\text{PDD}})/T + \theta_{\text{PPO}} + \theta_{\text{PA}}, \quad (9)$$

and, in turn, the CFR estimated at the beamformee during the channel sounding procedure becomes

$$\hat{H}_{k,m,n} = H_{k,m,n} e^{j\theta_{\text{offs},k,m,n}}. \quad (10)$$

Besides the PDD, all the other contributions to Eq. (9) are associated with imperfections at the transmitter device, that in our case is the target of our fingerprinting technique, i.e., the AP. Our key intuition is that also the beamforming feedback matrix $\tilde{\mathbf{V}}$ – derived from $\mathbf{H}$ as discussed in Section III-B – would be affected by the phase offsets (i)-(v). Thus, we may use the offsets cleaning algorithm of [36] to improve its quality. Along this line of reasoning, we evaluate in

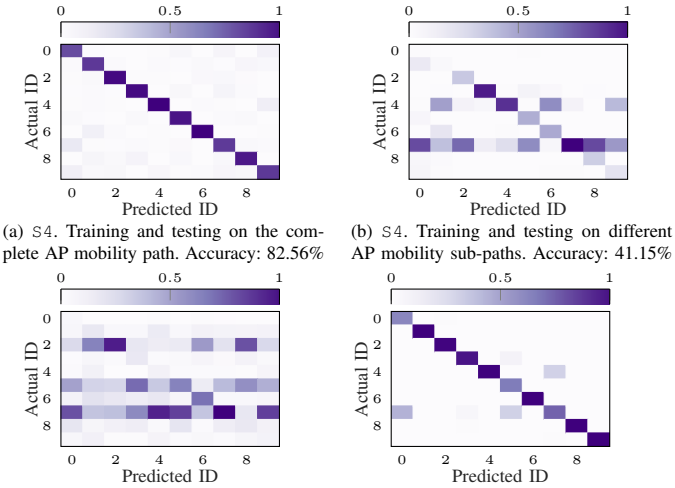


(a) DeepCSI accuracy compared with the one obtained using the processed input. (b) Confusion matrix for S1 after offset correction. Accuracy: 83.10%

Fig. 16: Comparison with the accuracy obtained by learning the fingerprints from the processed version of $\tilde{\mathbf{V}}$, i.e., after applying the offsets correction (offs. corr.) in [36]. Beamformee 1, 3 TX antennas, spatial stream 0.

Fig. 16 the impact of a preliminary offset cleaning phase on matrix $\tilde{\mathbf{V}}$ on the fingerprinting accuracy. DeepCSI (with no offsets cleaning) outperforms its version with the described offset correction capability across all the training/testing sets. In other words, the offsets introduced by the beamformer hardware imperfections are strategic to reliably recognize the device, and any offset cleaning may result in their partial removal, affecting the fingerprinting quality.

DeepCSI performance in the presence of beamformer mobility. The robustness of DeepCSI on beamformer's mobility is evaluated through dataset D2. In Figs. 17a-17b we report the performance of DeepCSI on set S4 (see Table II). Specifically, in Fig. 17a the entire mobility path for both the training and testing sets is considered. We remind that even if the theoretical path is the same for all the measurements, the operation is performed manually and, in turn, the actual shifts undergo uncontrolled variations that reflect on the collected traces. The results show that DeepCSI is able to effectively learn a fingerprint of the AP from the MU-MIMO beamforming feedback matrices even when the AP moves, reaching an accuracy above 80%. The proposed learning architecture allows compensating for the slight variations introduced by the manual shifts of the AP and the presence of the person moving in the vicinity. The fingerprint is less effective when the environmental conditions sharply depart from the training ones. The high-scale modifications on the beamforming feedback matrices – associated with the channel variations – prevent the neural network from effectively capturing the small-scale variations that descend from the hardware imperfections. We show this behaviour in Fig. 17b, where DeepCSI is trained and tested in different mobility conditions. The training and validation phases are performed on the first half of the traces in 'mob1', i.e., the portions related to the sub-path A-B-C-B. The test is executed on the fraction of the traces in 'mob2' collected while the AP spans the segments B-D-B. Overall, the results in Figs. 17a-17b indicate that the higher the variability in the training set, the more likely DeepCSI will learn fingerprints that are robust to changing radio channel conditions. The variability in the training set allows the learning algorithm to properly detect the elements that are in common to the traces and, in turn, identify the hardware-related features. In Figs. 17c-17d we report the performance of DeepCSI on sets S5 and S6. When DeepCSI is trained on the sole static traces – set S5 – the learned fingerprint is not effective in recognizing



(a) S4. Training and testing on the complete AP mobility path. Accuracy: 82.56%

(b) S4. Training and testing on different AP mobility sub-paths. Accuracy: 41.15%

(c) S5. Training on static conditions and test on mobility traces. Accuracy: 20.50%

(d) S6. Training on mobility traces and test on static conditions. Accuracy: 88.12%

Fig. 17: Confusion matrices, beamformee 1, 3 TX antennas, spatial stream 0.

the beamformer when it moves in the environment. On the other hand, once trained on the dynamic traces, DeepCSI is able to correctly identify the AP in static conditions (about 88% of accuracy on set S6). These last results confirm that the diversity in the training set is desirable to obtain a robust algorithm able to generalize over different conditions.

VI. Concluding Remarks

In this paper, we have presented a novel approach to Wi-Fi radio fingerprinting (RFP) which leverages IEEE 802.11-compliant steering matrices to authenticate MU-MIMO Wi-Fi devices. The present work makes the following key advances:

• **We have demonstrated for the first time the feasibility of RFP for MU-MIMO Wi-Fi**. To this end, DeepCSI leverages the beamforming feedback matrices computed by any of the beamformees and transmitted in clear to the beamformer. **We have verified that the matrices are affected by the beamformer hardware imperfections and, in turn, can be used to identify the device**. Moreover, the feedback is not affected by inter-stream and inter-user interference, thus increasing robustness. DeepCSI is independent of the number of beamformees associated with the target beamformer: different beamformer's fingerprints can be computed, one from each beamformee. Conversely from prior work, DeepCSI does not require direct CSI computation and, in turn, can be run on any Wi-Fi device without requiring SDRs.

• **We have performed a massive data collection campaign with off-the-shelf Wi-Fi equipment**, where 10 Wi-Fi radios emit MU-MIMO signals in different positions. Experimental results indicate that DeepCSI is able to correctly identify the transmitter with accuracy above 98%. We have evaluated DeepCSI fingerprinting accuracy by differentiating the set of positions for the devices at training and testing times. Our technique achieves accuracy of 73% when training is performed on a more balanced set of spatial points, which allows the classifier to interpolate the training patterns for the missing points, using those from adjacent training positions.

• **For the first time, we evaluated the proposed RFP technique with moving Wi-Fi devices.** `DeepCSI` reaches an accuracy above 82%, showing the robustness of the learned fingerprint to changing radio channel conditions. **Our results show that the higher the variability in the traffic traces used for the training phase, the higher is the accuracy when the algorithm is used at run-time to identify the devices**. This indicates the need for extensive datasets to train effective RFP algorithms. In this vision, **we pledge to make our contribution by sharing our datasets** [21].

As part of ongoing work, we plan to further investigate the effect of the beam patterns and of the positions of the receivers on the fingerprinting performance, also in the presence of interference from overlapping channels. As a further extension, we intend to investigate lifelong machine learning techniques to accumulate knowledge and improve the device's fingerprint as it moves in the environment.

ACKNOWLEDGEMENT

This material is based upon work supported in part by the National Science Foundation (NSF) under Grant No. CNS-2134973 and CNS-2120447, and by the Italian Ministry of Education, University and Research (MIUR) through the initiative "Departments of Excellence" (Law 232/2016). The views and opinions are those of the authors and do not necessarily reflect those of the funding institutions.

REFERENCES

[1] Federal Communications Commission (FCC), "Spectrum Crunch," https://www.nist.gov/advanced-communications/spectrum-crunch.

[2] Ericsson Incorporated, "Ericsson Interim Mobility Report, June 2020," https://www.ericsson.com/49da93/assets/local/mobility-report/documents/2020/june2020-ericsson-mobility-report.pdf, 2020.

[3] Jamie Davies, Telecoms.com, "FCC finally opens up 3.5 GHz for US telcos," https://telecoms.com/502070/fcc-finally-opens-up-3-5-ghz-for-us-telcos/, 2020.

[4] Federal Communications Commission (FCC), "FCC Opens 6 GHz Band to Wi-Fi and Other Unlicensed Uses," https://www.fcc.gov/document/fcc-opens-6-ghz-band-wi-fi-and-other-unlicensed-uses, 2020.

[5] Wi-Fi Alliance, "Wi-Fi 6E expands Wi-Fi into 6 GHz," https://www.wi-fi.org/file/wi-fi-6e-highlights, 2021.

[6] GSMA.com, "Capacity to Power Innovation: 5G in the 6 GHz Band," https://tinyurl.com/5G-6GHz-Bands, 2021.

[7] J. Horwitz, V. Beat, "Wi-Fi 6E and 5G Will Share 6GHz Spectrum to Supercharge Wireless Data," https://tinyurl.com/wyvmn5c, 2020.

[8] T. Zheng, Z. Sun, and K. Ren, "FID: Function Modeling-based Data-Independent and Channel-Robust Physical-Layer Identification," in *Proc. of IEEE INFOCOM*, 2019.

[9] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a Hybrid RF Fingerprint Extraction and Device Classification Scheme," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 349–360, 2019.

[10] F. Xie, H. Wen, Y. Li, S. Chen, L. Hu, Y. Chen, and H. Song, "Optimized Coherent Integration-Based Radio Frequency Fingerprinting in Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3967–3977, 2018.

[11] Y. Xing, A. Hu, J. Zhang, L. Peng, and G. Li, "On Radio Frequency Fingerprint Identification for DSSS Systems in Low SNR Scenarios," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2326–2329, 2018.

[12] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks," in *Proc. of IEEE INFOCOM*, 2019.

[13] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Fingerprinting Wi-Fi Devices Using Software Defined Radios," in *Proc. of ACM WiSec*, 2016.

[14] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia, "Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting," in *Proc. of IEEE INFOCOM*, 2020.

[15] E. Johnson, "Physical Limitations on Frequency and Power Parameters of Transistors," in *Proc. of IRE International Convention Record*, 1966.

[16] E. H. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtilä, "IEEE 802.11ac: Enhancements for very high throughput WLANs," in *Proc. of IEEE PIMRC*, 2011.

[17] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency WLANs," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 197–216, 2018.

[18] C. Deng, X. Fang, X. Han, X. Wang, L. Yan, R. He, Y. Long, and Y. Guo, "IEEE 802.11be Wi-Fi 7: New challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2136–2166, 2020.

[19] IEEE, "IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz," *IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012)*, 2013.

[20] ——, "IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN," *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)*, 2021.

[21] F. Meneghello, M. Rossi, and F. Restuccia, "DeepCSI – code and datasets," https://github.com/signetlabdei/DeepCSI, 2022.

[22] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device Fingerprinting in Wireless Networks: Challenges and Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.

[23] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms," in *Proc. of ACM MobiHoc*, 2019.

[24] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep Learning Convolutional Neural Networks for Radio Identification," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, 2018.

[25] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep Learning for RF Device Fingerprinting in Cognitive Communication Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 160–167, 2018.

[26] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. Moura, "A Deep Learning Approach to IoT Authentication," in *Proc. of IEEE ICC*, 2018.

[27] S. Gopalakrishnan, M. Cekic, and U. Madhow, "Robust Wireless Fingerprinting via Complex-Valued Neural Networks," in *Proc. of IEEE GLOBECOM*, 2019.

[28] S. D'Oro, F. Restuccia, and T. Melodia, "Can You Fix My Neural Network? Real-Time Adaptive Waveform Synthesis for Resilient Wireless Signal Classification," in *Proc. of IEEE INFOCOM*, 2021.

[29] E. Perahia and R. Stacey, *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*. Cambridge Univ. Press, 2008.

[30] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[31] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Feb 2018.

[32] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," in *Proc. of ACM NIPS*, 2017.

[33] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Proc. of ECCV*, 2018.

[34] D. J. ketchen and C. L. Shook, "The Application of Cluster Analysis in Strategic Management: an Analysis and Critique," *Strategic Management Journal*, vol. 17, pp. 441–458, 1996.

[35] IEEE, "TGac Channel Model Addendum. Version 12. IEEE 802.11-09/0308r12," 2010.

[36] F. Meneghello, D. Garlisi, N. Dal Fabbro, I. Tinnirello, and M. Rossi, "Environment and Person Independent Activity Recognition with a Commodity IEEE 802.11ac Access Point," *arXiv preprint arXiv:2103.09924*, 2021.

[37] H. Zhu, Y. Zhuo, Q. Liu, and S. Chang, "π-Splicer: Perceiving Accurate CSI Phases with Commodity WiFi Devices," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2155–2165, 2018.