Neural Network-Driven Estimation of Hardware Impairments for Robust Wireless Device Identification

Haytham Albousayri [1] **Bechir Hamdaoui** [1 2] **Weng-Keen Wong** [1]

Abstract

We proposed a neural network (NN)-based framework for the joint estimation of hardware impairments in wireless devices. We validated the approach using real-world measurements from Bluetooth Low Energy (BLE) and WiFi devices. Experimental results show that the estimated impairments exhibit a stationary distribution after the warm-up phase, highlighting their stability and potential as device-specific fingerprints. Moreover, we demonstrated that these impairments provide a lightweight yet robust alternative to raw IQ samples for RF fingerprinting (RFFP), achieving an average improvement of 54% in classification accuracy across different domains.

Keywords: Wireless device identification, hardware impairment estimation, neural network-based optimization.

1. Introduction

Recent studies have shown that hardware-induced imperfections can serve as unique, device-specific fingerprints and signatures, extractable from received radio frequency (RF) signals, thereby enabling security features such as network device identification and authentication (Elmaghbub & Hamdaoui, 2023; del Arroyo et al., 2024). These RF-based security mechanisms are especially promising for low-end IoT devices and networks, where traditional cryptographic approaches may be too resource-intensive. As a result, estimating and leveraging hardware impairments for device fingerprinting has recently become a prominent area of research focus (Elmaghbub & Hamdaoui, 2023; del Arroyo et al., 2024; Givehchian et al., 2022).

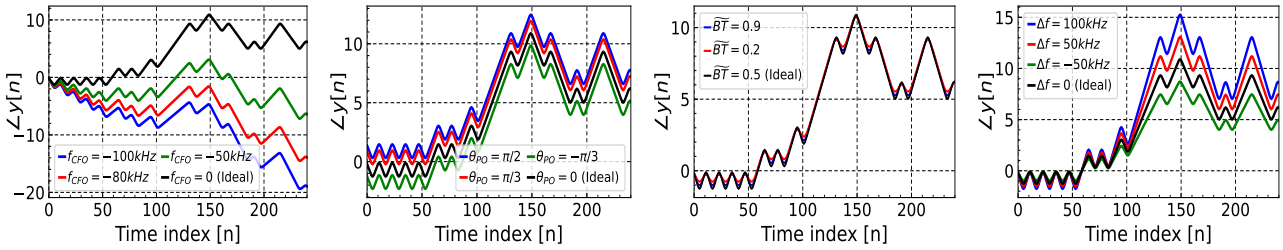
Early works centered their effort on developing analytical models of these impairments. For instance, Polak et al. (Polak & Goeckel, 2015) proposed statistical methods to estimate the CFO and phase noise, emphasizing their temporal consistency and device specificity. Elmaghbub et al. (Elmaghbub & Hamdaoui, 2023) examined the impact of domain shifts on RF fingerprints and found that WiFi signal envelopes distorted by CFO can be used as reliable identifiers. Del Vecchio et al. (del Arroyo et al., 2024) showed that the distribution of IQ imbalance varies across devices and can serve as a distinctive fingerprinting feature. Building on this, recent studies have demonstrated that a vector of estimated hardware impairments can serve as a robust and lightweight device identifier. For instance, Elmaghbub et al. (Elmaghbub & Hamdaoui, 2024a) successfully identified WiFi devices using impairments measured with a signal analyzer. The authors studied the warm-up phase effect on the fingerprints, but did not study the domain adaptation problem. Givehchian et al. (Givehchian et al., 2022) performed a physical-layer tracking attack—essentially a form of RF fingerprinting—by jointly estimating several hardware impairments of the target devices. To address the resulting non-convex optimization problem, the authors proposed an iterative approach, which, however, often converges to suboptimal solutions.

Motivated by these challenges, we propose a framework for jointly estimating hardware impairments directly from received RF signals. Specifically, we design a neural network (NN)-based model to estimate these hardware impairments by solving the underlying optimization problem. Experimental results show that our method offers a more stable and efficient optimization process, reaching lower loss values in fewer steps compared to conventional iterative techniques. In addition, visual analysis shows that the estimated impairments exhibit stable distributions, making them effective compressed features and improving device identification accuracy by 54% compared to models using raw signals as their inputs. Moreover, we showed that these estimated impairments provide a lightweight yet robust alternative to raw IQ samples for RF fingerprinting and device identification, achieving an average improvement of 54% in classification accuracy.

The remainder of this paper is organized as follows. Section 2 introduces key RF hardware impairments and describes their impact on received IQ signals. Section 3

[1]Oregon State University, Corvallis, OR, USA [2]Hamad Bin Khalifa University, Doha, Qatar. Correspondence to: Haytham Albousayri <albousah@oregonstate.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).



(a) CFO (b) PO (c) BT (d) MFD

Figure 1. Impairments' impact on the instantaneous IQ phase of BLE/GFSK signals.

presents the problem formulation and describes the proposed neural-network framework for solving the formulated optimization problem. Section 4 evaluates the accuracy and robustness of the proposed framework using WiFi and BLE data. Finally, Section 5 concludes the paper and summarizes the main findings.

2. The Impact of Device Hardware Impairments on IQ Signals

Radio Frequency (RF) hardware impairments refer to the inherent imperfections and non-ideal behaviors of the RF hardware components of communication devices. These impairments arise from manufacturing variations, component tolerances and environmental influences, collectively affecting signal integrity and communication performance. Although RF impairments distort transmitted signals and reduce overall system efficiency, it also plays a crucial role in security applications such as device identification and authentication. Understanding, modeling, and mitigating these hardware imperfections are needed for designing robust wireless communication systems and maintaining reliable connectivity. In this section, we study and demonstrate via simulations the impact of different hardware impairment values on the IQ signal behavior. Our simulator implemented the BLE PHY layer specifications as indicated in (BT-Spec-2023) and adopted the GFSK 1M PHY mode described in Section 3.1.1. An arbitrary 41-bit sequence, sampled at 6 MS/s, resulting in a signal duration of 246 samples, was used to generate all the presented graphs. In this work, we studied the following key impairments.

2.1. Carrier Frequency Offset (f_{CFO})

The carrier frequency offset (CFO) represents the deviation of the carrier frequency from its nominal frequency value and is typically caused by factors like local oscillator inaccuracies and Doppler shift. In Figure 1(a), we show the impact of CFO values on the phase of received IQ of Bluetooth signals. Observe that CFO introduces a slope in the signal's phase, with different CFO values resulting in varying slopes.

2.2. Phase Offset (θ_{PO})

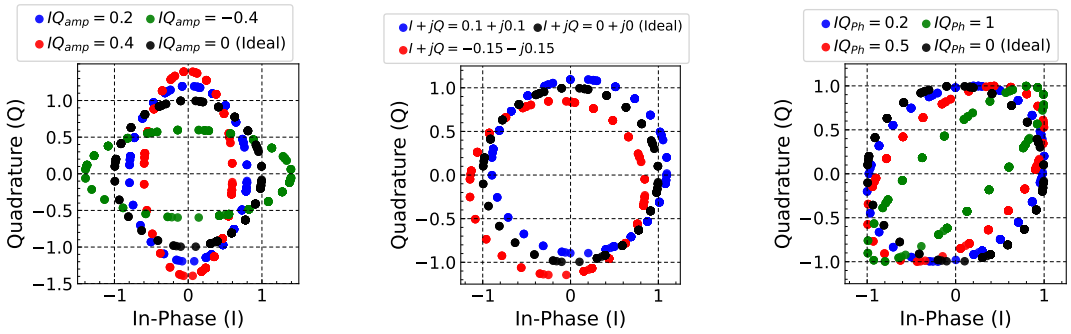
Phase offset (PO) is another form of distortion arising from both device-specific and channel-specific factors. While PO is generally treated as a nuisance parameter rather than a reliable device identifier, its accurate estimation is critical, as it can significantly affect the estimation of other impairments—particularly due to its strong sensitivity to channel conditions (Xu & Kan, 2023). Figure 1(b) illustrates the phase of IQ signals under varying phase offset values, where different θ_{PO} values result in vertical shifts in the signal phase.

2.3. Bandwidth Duration (BT) Product

BT characterizes the smoothness of the Gaussian filter and is typically set to a standard value; e.g., 0.5 in the Bluetooth specification (BT-Spec-2023). In practical systems, however, the filter's 3-dB bandwidth (B) may deviate from this nominal value, leading to a distorted version, denoted by $\widetilde{BT}$. These deviations have been shown to function as distinctive device-specific identifiers (Zhang et al., 2025). Figure 1(c) illustrates the impact of varying BT values on the system behavior, where smaller values lead to flatter pulse shapes.

2.4. Maximum Frequency Deviation (MFD) Offset

MFD is a frequency modulation-specific parameter that represents the difference between the maximum positive frequency and the central frequency, determined by the modulation index. In BLE, the modulation index is defined to be between 0.45 and 0.55, with an ideal value of 0.5, resulting in a frequency deviation of 250 kHz (BT-Spec-2023). However, hardware imperfections can affect this deviation adding an error Δf to the optimal value and leading to a distorted signal, thereby contributing to the overall device fingerprint. Figure 1(d) shows how the addition of Δf to the ideal MFD impacts the instantaneous phase of the signal, creating a separation that be used for device identification.



(a) IQ amplitude imbalance (b) DC offset (c) IQ phase imbalance

Figure 2. Constellation diagrams with the impact of IQ imbalance and DC offset on BLE/GFSK signals.

2.5. IQ Amplitude Imbalance (IQ_{Amp}), IQ Phase Imbalance (IQ_{Phase}) and IQ Offset (I_{DC} and Q_{DC})

IQ imbalance arises when there is an amplitude and/or phase mismatch between the in-phase (I) and quadrature (Q) components of a transmitter or receiver, typically resulting from hardware manufacturing errors and variations (Schuchert et al., 2001). Figure 2(a) and 2(c) show how the amplitude and phase imbalances affect the constellation diagram, which ideally forms a unit circle. We can see that higher imbalances leads to an elliptic distortion. IQ offset, on the other hand, occurs when the IQ origin shifts from its intended position, often appearing as a constant DC offset or carrier feedthrough in the modulated signal. This misalignment is also caused by hardware imperfections, such as imbalances in the analog and digital processing chains, and can degrade signal quality if not properly corrected. Figure 2(b) shows the effect of different DC offset values on the IQ the constellation diagram.

3. The Proposed Neural Network-Based Impairments Estimation Framework

Estimating and compensating for hardware impairments has always been crucial for reliable communication. However, recent research efforts have also leveraged such device-specific impairments to fingerprint and identify wireless devices to support various security application like automated network authentication. Most existing systems estimate hardware impairments using method-of-moments (MoM) techniques because they are simple and fast. However, these methods are often less accurate than approaches that learn by minimizing a loss function. In this work, we propose a loss-based method that directly learns to estimate hardware impairments from RF signals.

3.1. The Impairment Estimation Formulation

Let $y_G[n; \hat{\Theta}]$ denote a synthetic (generated) baseband signal distorted by the impairment vector $\hat{\Theta}$. Let $y_M[n]$ represent the complex baseband signal captured from device transmissions, for which we aim to estimate the underlying impairments. Here, $n = 0, 1, \ldots, N$, where N denotes the total number of signal samples. We assume that both signals, $y_M[n]$ and $y_G[n; \hat{\Theta}]$, convey the same bit sequence $d[n]$. Omitting the time variable n for simplicity and without loss of generality, the impairments estimation problem can be formulated as:

$$\min_{\hat{\Theta}} \ \mathcal{L}\left(f(y_M), f(y_G(\hat{\Theta}))\right) \tag{1}$$

subject to $IQ_{Phase} \in [-\pi, \pi]$ and $\theta_{PO} \in [-\pi, \pi]$ where $\mathcal{L}(\cdot, \cdot)$ denotes the estimation loss function, and $f(\cdot)$ is a non-linear transformation function that maps the input signal to an alternative representation, such as phase, magnitude, or other signal-specific representations. We found that selecting a transformation function $f(\cdot)$ that unfolds the signals as a direct function of the binary bit sequence d significantly improves convergence; more on this will be said later.
Since both signals share the same bit sequence d, the generated signal $y_G(\hat{\Theta})$ can be expressed as:

$$y_G(\hat{\Theta}) = \mathbf{Mod}\left(\mathbf{Dem}\left(y_M\right), \hat{\Theta}\right)$$

where $\mathbf{Dem}(\cdot)$ denotes a demodulation function that takes a measured signal and returns its corresponding baseband bit sequence d, and $\mathbf{Mod}(d, \hat{\Theta})$ is a modulation function that takes the bit sequence d and an impairments vector $\hat{\Theta}$ and generates the corresponding baseband modulated signal.

In this work, we validated our proposed estimation approach using both WiFi and BLE signals. To perform this validation, we implemented the signal generation and modulation procedures in Python. For completeness, we next provide details on the $\mathbf{Mod}(\cdot, \cdot)$ and $\mathbf{Dem}(\cdot)$ functions used for each of the WiFi and BLE communication technologies.

3.1.1. THE $\mathbf{Mod}(\cdot,\cdot)$ FUNCTION BLOCK

WiFi/DSSS Signal Generation: Given the baseband data $d[n]$ and the predefined DSSS spread code $c[n]$, the baseband spread signal can be expressed as $g[n] = d[n] \cdot c[n]$. Assuming Gaussian pulse-shaping filter (Linz & Hendrickson, 1996), we can then write $h[n] = \frac{\sqrt{\pi}}{a} e^{-\pi^2 n^2/a^2}$ where $a = (1/\widetilde{BT})\sqrt{ln(2)/2}$ is a parameter related to 3-dB bandwidth. Again, $\widetilde{BT}$ represents bandwidth duration product under hardware impairments. The pulse-shaped baseband signal is then given by $x[n] = g[n] * h[n]$, where $*$ represents the convolution operation. The generated WiFi signal distorted by hardware impairments can be modeled as (del Arroyo et al., 2024):

$$\tilde{y}[n] = (\tilde{y}_I[n] + j\tilde{y}_Q[n])\, e^{j2\pi f_{\text{CFO}} n} \tag{2}$$

with $\tilde{y}_I[n] = G_I\, x[n] \cos(\theta_{PO} + IQ_{Phase}/2) + I_{DC}$ and $\tilde{y}_Q[n] = G_Q\, x[n] \sin(\theta_{PO} - IQ_{Phase}/2) + Q_{DC}$. Where f_{CFO}, θ_{PO}, I_{DC}, Q_{DC} and IQ_{Phase} are again the carrier frequency offset, phase offset, In-phase component of DC-offset, Quadrature component of DC-offset, and phase imbalance between the In-phase and Quadrature components, respectively. G_I and G_Q represent the in-phase and quadrature gains introduced by the IQ amplitude imbalance IQ_{Amp}, and are defined as $G_I = 10^{\,IQ_{Amp}/40}$ and $G_Q = 10^{-IQ_{Amp}/40}$. The modulation function block for WiFi can then be implemented as $\mathbf{Mod}_{WiFi}(d[n], \hat{\Theta}) \triangleq y_G[n; \hat{\Theta}] = \tilde{y}[n]$ where $\tilde{y}[n]$ is as defined in Eq. (2) and $\hat{\Theta} = \{\widetilde{BT}, f_{CFO}, \theta_{PO}, I_{DC}, Q_{DC}, IQ_{Phase}, IQ_{Amp}\}$.

BLE/GFSK Signal Generation: Given the baseband data $d[n]$ and the impulse response of a Gaussian pulse-shaping filter $h[n]$, the Gaussian filtered pulse stream can be expressed as $g[n] = d[n] * h[n]$. The instantaneous angular shift function can then be expressed as discrete version of running integral $\phi[n] = 2\pi\,(f_m + \Delta f) \sum_{k=0}^{n} g[k]\; T_S$, where f_m, Δf and T_S represent the optimal peak frequency deviation, the offset from the optimal value of f_m and the sampling interval, respectively. Figure 3 illustrates the BLE modulation steps, starting from $d[n]$ and ending at $y_G[n]$, with all impairment values set to zero. The received distorted signal can be expressed as (Givehchian et al., 2022):

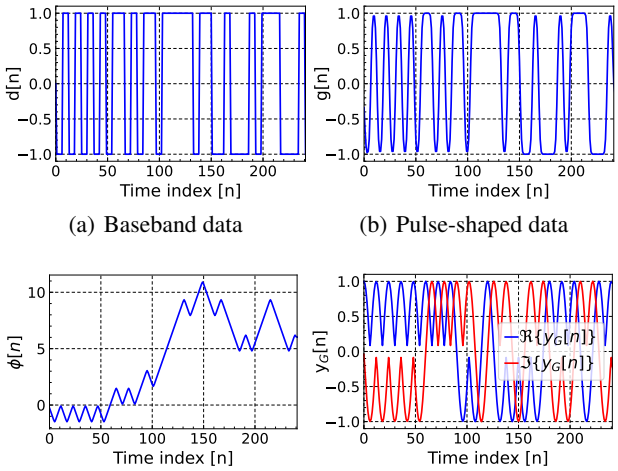
$$\tilde{y}[n] = (\tilde{y}_I[n] + j\tilde{y}_Q[n])\, e^{j(2\pi f_{\text{CFO}} n + \theta_{PO})} \tag{3}$$

with $\tilde{y}_I[n] = (1 - IQ_{Amp}) \cos(\phi[n] - IQ_{Phase}/2) + I_{DC}$ and $\tilde{y}_Q[n] = (1 + IQ_{Amp}) \sin(\phi[n] + IQ_{Phase}/2) + Q_{DC}$. As with the WiFi case, the BLE modulation function can similarly be implemented as $\mathbf{Mod}_{BLE}(d[n], \hat{\Theta}) = \tilde{y}[n]$ where $\tilde{y}[n]$ is as defined in Eq. (3) and $\hat{\Theta} = \{\widetilde{BT}, \Delta f, f_{CFO}, \theta_{PO}, I_{DC}, Q_{DC}, IQ_{Phase}, IQ_{Amp}\}$.

3.1.2. THE $\mathbf{Dem}(\cdot)$ FUNCTION BLOCK

Demodulation, the process of retrieving the baseband bit sequence from a received signal, is essentially the inverse of the signal modulation steps described earlier. While demodulating a signal under high levels of impairment-induced distortion can lead to errors, it remains feasible when the impairments are sufficiently small. In the case of WiFi, given a captured signal $y_M[n]$, we extract the unwrapped phase $\angle y_M[n]$ to recover the spread code sequences. Figure 4(a) shows the normalized phase of the WiFi preamble, where the red dashed line separates the original data bits, which alternate between 0 and 1. Each data bit is represented by a fixed 11-chip spreading code. The captured signal exhibits noticeable distortions, including a negative phase offset.

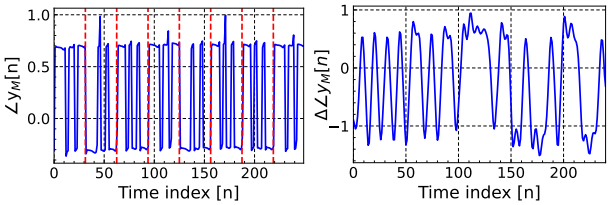
In BLE, the original bit sequence can be recovered by computing the discrete-time derivative (i.e., the first difference) of the unwrapped phase of $y_M[n]$, denoted as $\Delta_n \angle y_M[n]$. Figure 4(b) shows the normalized $\Delta_n \angle y_M[n]$ of BLE signal, which carries the same sequence as the one shown in Figure 3. When the signal is highly distorted and produces bit errors, a practical solution is for both ends to agree on a predefined bit sequence, thereby bypassing the need for the demodulation block. Alternatively, lightweight impairment estimation/compensation techniques can be applied for bit recovery. However, such impairments can be noisy and inaccurate, thus unsuitable for RFFP tasks.



(a) Baseband data (b) Pulse-shaped data

(c) Angle modulated data (d) Complex baseband signal

Figure 3. Steps of the BLE signal modulation process.



(a) WiFi (b) BLE

Figure 4. Signal representation for demodulation blocks

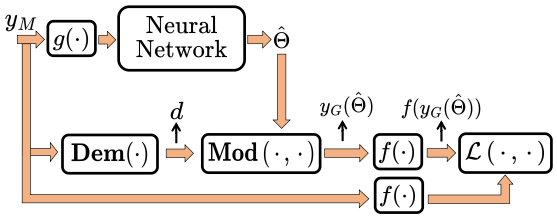


Figure 5. The proposed NN-based estimation framework.

3.2. The Proposed NN-based Optimization Framework

Solving (1) is challenging due to the non-convex nature of the problem (Givehchian et al., 2022). Thanks to their non-linear activation functions and robust optimizers, neural networks (NNs) have shown a stable estimation and good convergence when it comes to solving non-convex complex optimization problems (Kawaguchi, 2016; Nikbakht et al., 2020). Motivated by this, we opted for using neural networks to minimize the loss between $f(y_M)$ and $f(y_G(\hat{\Theta}))$. Since directly using the raw signals as input to the neural network often leads to suboptimal performance (Elmaghbub & Hamdaoui, 2023), we introduce a transformation function $g(\cdot)$ that projects the raw signal into alternative representations that are more suitable for the impairments estimation task. After passing y_M through $g(\cdot)$, the results are used as an input to the neural network to output the estimated impairments, $\hat{\Theta}$, as shown in Fig. 5. y_M is also passed to $\mathbf{Dem}(\cdot)$ to output a baseband bit sequence, which, along with the output $\hat{\Theta}$ of the NN, are then fed to the modulation block, $\mathbf{Mod}(\cdot,\cdot)$, to produce $y_G(\hat{\Theta})$ and finally $f(y_G(\hat{\Theta}))$. The NN is updated by minimizing the loss between transformed signals $f(\cdot)$–instead of the raw signals–to address the multiple-solution ambiguity caused by the inherent sinusoidal structure of BLE/GFSK baseband signals (McKilliam et al., 2010).

Finally, we satisfied the objective constraints by utilizing scaled and shifted versions of a $\tanh(\cdot)$ activation function from each neuron of the output layer. We found that the L1 loss consistently yielded better performance than the L2 loss, and therefore adopt it as our loss function $\mathcal{L}(\cdot,\cdot)$. To ensure stability, we applied L2 regularization with coefficient λ, leading to the following modified optimization

$$\min_{\hat{\Theta}} \mathcal{L}\left(f(y_M), f(y_G(\hat{\Theta}))\right) + \lambda \sum_{i=1}^{W} w_i^2 \tag{4}$$

where w_i denotes the the i-th NN parameter and W is the total number of NN parameters.

4. Performance Evaluation and Analysis

We validate the accuracy of the proposed estimation framework and assess its fingerprinting effectiveness using WiFi and BLE datasets. After updating the weights using Eq. (4), we evaluate the achievable performances using the L2 error directly between the signals y_M and $y_G(\hat{\Theta})$, denoted as $\mathbf{MSE}(y_M, y_G(\hat{\Theta}))$. Optimization terminates when this error converges or falls below a predefined threshold ϵ. In addition to that, we define the *matching score* S, as the percentage of generated samples that closely match the corresponding measured samples, where a generated sample is considered a match if its relative absolute error with respect to the measured sample is less than the threshold τ. Formally,

$$S = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left(\left|\frac{y_G[n;\hat{\Theta}] - y_M[n]}{y_M[n]}\right| < \tau\right) \times 100\% \tag{5}$$

where N denoting the total number of samples and τ representing the relative error threshold used to determine a match. $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition within it is satisfied and 0 otherwise. The proposed NN used for estimation includes 5 CNN blocks, each followed by MaxPooling layers and LeakyReLU activation function, and 2 Fully connected NN, each followed by ReLU activation function, the rest of the model parameters are summarized in Table 1.

Table 1. Proposed framework architecture parameters

Parameter	**Value**
Number of Filters (F)	$\{64, 64, 96, 128, 96\}$
Kernel Sizes (H)	$\{48, 18, 98, 106, 98\}$
Number of Neurons (N)	$\{1024, 512\}$
Initial learning rate	1.84×10^{-4}
λ	8.3×10^{-4}
τ	0.15
Optimizer	AdamW

4.1. Proposed NN-based Method vs. Iterative Method

We begin our evaluation by comparing the proposed optimization approach with baseline methods, specifically iterative optimization techniques. For this comparison, Gradient Descent (GD) (Rumelhart et al., 1986) and Adam method (Kingma, 2014) were employed to solve the minimization problem defined in Eq. (4). Both methods were evaluated using proper learning rates across 50 different random initializations using distinct seed values to ensure robustness and fairness in the comparison. Figure 6 presents the $\mathbf{MSE}$ loss along with a confidence interval representing one standard deviation, plotted against the number of update steps (epochs). The results clearly demonstrate that the proposed model consistently outperforms the baseline iterative methods. Notably, our approach converges to a loss on the order of 10^{-6} within approximately 70–110 steps, whereas the iterative methods require around 2500 steps to reach convergence—if they converge at all. Traditional iterative optimization methods are limited by their sequential loss

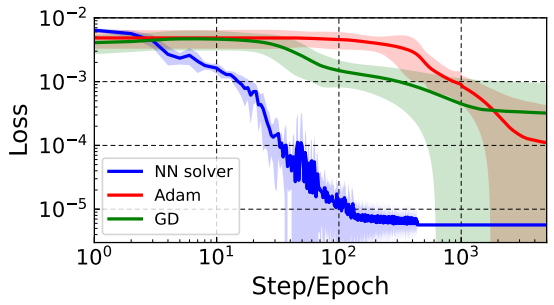


Figure 6. MSE loss in estimating BLE impairments

computation, where each step depends on the previous one. In contrast, neural network-based solvers, despite computing loss sequentially, mitigate this bottleneck by computing a more accurate loss per update by leveraging large numbers of highly parallelized parameters, fully utilizing GPU hardware and resulting in fewer update-steps overall.

4.2. Estimation Validation Results

We validate the accuracy of the proposed NN–based estimation model using WiFi signals by comparing the hardware impairment estimates generated by our model with ground-truth measurements obtained from a Keysight N9030B PXA signal analyzer. For this, we leverage the WiFi 802.11b RF Fingerprints with Hardware Impairments Dataset introduced in (Elmaghbub & Hamdaoui, 2024a), which provides recorded WiFi signals alongside corresponding impairment values measured using the Keysight analyzer.

For the generated WiFi signals, we used the same bit sequences and spreading codes described in Sec. 3.1.1 to generate BPSK/DSSS signals. Without loss of generality, we set the transformation functions $g(\cdot)$ and $f(\cdot)$ defined in Sec. 3.2 as the identity, i.e., $g(z) = f(z) = z$. All other parameters were retained as previously defined.

In our evaluation, we primarily focused on comparing the CFO measured via Signal Analyzer (SigAn), denoted as $f_{\text{CFO}}^{\text{SigAn}}$, and our estimated CFO, denoted as $f_{\text{CFO}}^{\text{Estimated}}$. After jointly estimating all hardware impairments using our proposed method, we generated the synthetic signal $y_G[n; \hat{\Theta}]$ twice: once using $f_{\text{CFO}}^{\text{Estimated}}$ to get $y_G^{\text{Estimated}}[n; \hat{\Theta}]$, obtained from our model, and once using $f_{\text{CFO}}^{\text{SigAn}}$ to get $y_G^{\text{SigAn}}[n; \hat{\Theta}]$, while keeping all other impairments fixed.

Table 2 presents $f_{\text{CFO}}^{\text{SigAn}}$ and $f_{\text{CFO}}^{\text{Estimated}}$ values, along with their corresponding average mean squared error (MSE) and matching score of their generated waveforms, evaluated during both the stable and warm-up phases of transmission. As shown, estimated CFOs derived from stable-phase signals closely match the measured values, whereas those obtained during the warm-up phase exhibit higher discrepancies. Overall, the table demonstrates that our estimated CFO values achieve superior performance in terms of their average MSE and *matching score* across all the evaluated cases.

Table 2. Comparison of SigAn and estimated CFO values f_{CFO} (Hz), along with their associated average **MSE** $[10^{-5}]$ and average matching score [%], at $N = 2000$.

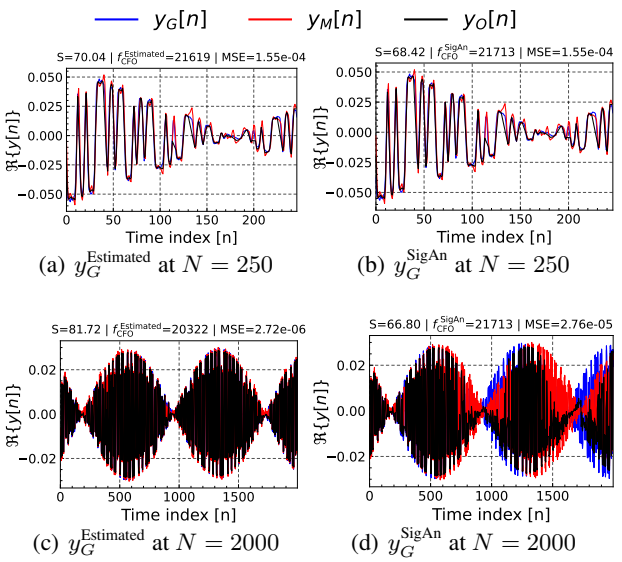
	Signal Analyzer		**Ours**	
	$f_{\text{CFO}}^{\text{SigAn}}$	**MSE**$\downarrow$ ($S\uparrow$)	$f_{\text{CFO}}^{\text{Estimated}}$	**MSE**$\downarrow$ ($S\uparrow$)
Stable	12481.7	0.58 (75.2)	12361.7$\pm$55.0	**0.56 (77.1)**
	12510.6	4.52 (**69.5**)	12538.8 $\pm$20.7	**4.49 (69.5)**
	12776.9	3.34 (70.1)	12624.5$\pm$35.2	**3.30 (70.2)**
	13099.3	1.23 (71.7)	12695.1$\pm$24.8	**1.02 (77.3)**
Warming-up	15286.8	5.17 (63.3)	14141.5$\pm$2.33	**3.55 (71.9)**
	17633.5	6.00 (59.9)	16381.5$\pm$14.9	**4.07 (70.2)**
	19069.0	8.26 (60.5)	17798.9$\pm$42.6	**6.40 (69.3)**
	20766.5	9.66 (57.6)	19013.6$\pm$65.6	**6.04 (67.1)**
	22125.1	8.49 (59.7)	20731.3$\pm$42.7	**6.19 (69.3)**

As the impairments can be highly unstable during the warm-up phase (Elmaghbub & Hamdaoui, 2024a), we studied the impact of the size N on the estimated impairments during both the warm-up and stable periods. Figure 7 shows the In-phase component of the generated signals $y_G^{\text{SigAn}}[n]$ and $y_G^{\text{Estimated}}[n]$ based on the estimated and measured values of CFO, respectively, for $N = 250$ and $N = 2000$. The signal $y_O[n]$ is measuring the overlaps between the captured signal $y_M[n]$ and the generated one, using the *matching score* defined in Eq. (5). The figure shows that for small N, the estimated CFO closely matches the value measured by the signal analyzer, giving approximately similar **MSE** of 1.55×10^{-4} and around 1.62% improvement on S. On the other hand, when N is large, the gap between $f_{\text{CFO}}^{\text{Estimated}}$ and $f_{\text{CFO}}^{\text{SigAn}}$ increases, introducing a larger mismatch between the generated signals. This leads to differences in both **MSE** and S values, where our estimated CFO achieves $S = 81.72\%$ and $\mathbf{MSE} = 2.72 \times 10^{-6}$, compared to the measured CFO that results in $S = 66.80\%$ and $\mathbf{MSE} = 2.76 \times 10^{-5}$.

4.3. Fingerprinting Accuracy Results

After demonstrating the effectiveness of our estimation framework, we now leverage the estimated impairments to perform lightweight RF fingerprint classification. Specifically, we compared classification performance using the estimated impairment vector $\hat{\Theta}$ as input to the machine learning classifier against using normalized raw IQ samples.

For this assessment, we used BLE signals that we collected in our lab. The BLE dataset comprises stable BLE frames captured from multiple devices under diverse scenarios, locations, receiver setups, and frequency channels. Wireless data were collected at two locations with device-to-receiver



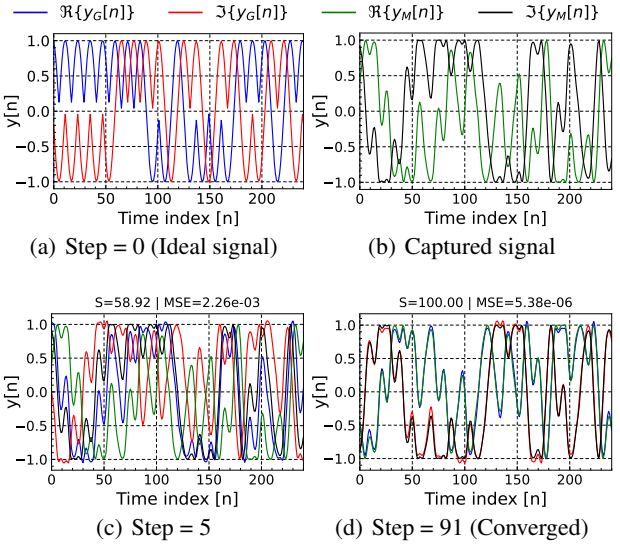
(a) $y_G^{\text{Estimated}}$ at $N = 250$ (b) y_G^{SigAn} at $N = 250$

(c) $y_G^{\text{Estimated}}$ at $N = 2000$ (d) y_G^{SigAn} at $N = 2000$

Figure 7. Impact of the number of samples, N, on the the estimated impairments during the warm-up period

spacings of 1m and 2m, while maintaining a fixed receiver and a BLE channel. For wired data collection, the same setup was used, but with varying frequency channels. Without loss of generality, Channel 1 (Ch1), Channel 2 (Ch2), and Channel 14 (Ch14) were selected for this study. Additionally, a second receiver was employed to collect wired data on Ch1. Prior to each data collection session, all devices were powered on and allowed a 6-minute warm-up period to ensure hardware stabilization (Elmaghbub & Hamdaoui, 2024b). Data were collected using GNU Radio, capturing raw IQ samples with a 2MHz bandwidth and a 6MS/s sampling rate. The receiver gain was set to 29dB for wireless and 8dB for wired measurements.

To obtain a reliable estimate of $\hat{\Theta}$, we used $N = 250$ samples for estimation—approximately three times the length of a BLE preamble—and set the stopping condition to $\epsilon = 6 \times 10^{-6}$. To resolve the ambiguity caused by encoding information in sinusoidal GFSK signals, the transformation functions $g(z)$ and $f(z)$ presented in Sec. 3.2 are defined as $f(z[n]) = g(z[n]) + C_D \Delta_n g(z[n])$ with $g(z[n]) = \text{unwrap}(\angle z[n])$, Δ_n being the first order discrete time derivative and $C_D = 10$ being a constant that regulates the impact of the derivative. The time derivative term is included based on the observation that it directly relates to CFO. In particular, when $C_D \to \infty$, the estimator ignores static impairments such as Phase Offset (PO) and focuses on dynamic ones like CFO. In the other hand, when $C_D = 0$, all impairments are considered, but this might place less emphasis on critical ones like CFO (Elmaghbub & Hamdaoui, 2023). Figure 8 illustrates how a BLE generated signal progressively aligns with the measured signal after several iterations, resulting in an accurate estimate of $\hat{\Theta}$ using our proposed method.



(a) Step = 0 (Ideal signal) (b) Captured signal

(c) Step = 5 (d) Step = 91 (Converged)

Figure 8. The optimization process of our proposed framework

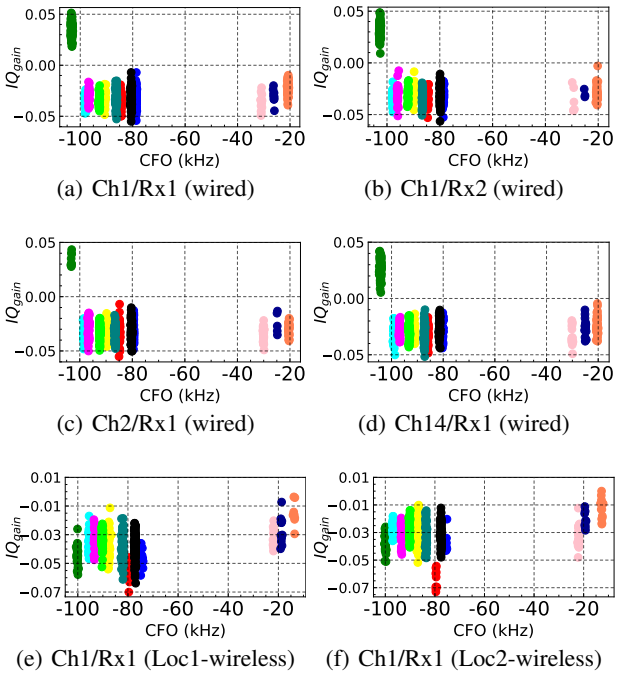
For consistency and fair comparison, the classifier using raw IQ samples adopted the same architecture presented in Table 1. In contrast, the classifier using $\hat{\Theta}$ employed a lightweight two-layer fully connected neural network with 32 neurons in each layer. Both classifiers share a common output layer consisting of 12 neurons, corresponding to the 12 target devices. The overall dataset consists of approximately 2,400 frames per device and was split into training, testing, and validation sets with ratios of 78.4%, 20%, and 1.6%, respectively.

4.3.1. Robustness to Varying Receivers

We first demonstrate device separability through visualization of the estimated impairments, and then assess our proposed method's ability in performing device classification using the estimated values of $\hat{\Theta}$.

Figs. 9(a) and 9(b) show the estimated values of CFO and IQ_{gain} for 12 devices, each represented by a distinct color, when signals are sampled by Receiver 1 and Receiver 2, respectively. IQ_{gain} is defined as $\frac{1+IQ_{Amp}}{1-IQ_{Amp}}$. The figure clearly indicates that each device exhibits a unique distribution that remains consistent across different receivers.

Fig. 10 shows the classification accuracy when training is performed on data collected by Rx1 or by Rx2 and tested on data collected by Rx1 or Rx2. Our proposed approach demonstrates stronger overall generalization across different receivers. It achieves an accuracy of 97.22% when trained and tested on Rx1 data and 88.52% when trained on Rx1 data and tested on Rx2 data—outperforming the raw IQ-based classifier by approximately 66%. Similarly, when trained and tested on data from Rx2, our approach obtains around 87% accuracy and 81.5% when tested on Rx1, yielding a 45% improvement over the raw IQ-based classifier.



(a) Ch1/Rx1 (wired) (b) Ch1/Rx2 (wired)

(c) Ch2/Rx1 (wired) (d) Ch14/Rx1 (wired)

(e) Ch1/Rx1 (Loc1-wireless) (f) Ch1/Rx1 (Loc2-wireless)

Figure 9. Estimated IQ_{gain} vs CFO in various domains, where each point represents the average of five independent estimates

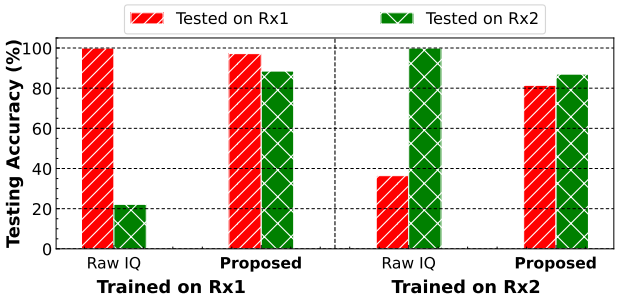


Figure 10. Classification results for 12 BLE devices

Note that there was a slight drop in classification accuracy when the model was trained and tested on the same receiver. This is expected, as compressing the RF fingerprinting information from a raw IQ signal of size $N = 1850 \times 2$ (2 for I&Q) into a compact vector of size $|\hat{\Theta}| = 8$ inevitably leads to some information loss during the process.

4.3.2. Robustness to Varying Environment

We now evaluate the proposed method's ability to perform fingerprinting under varying environmental conditions and across different frequency channels—a known challenge when using raw IQ-based fingerprinting models (Fu et al., 2023; Hamdaoui & Elmaghbub, 2022).

Figs. 9(c) and 9(d) present the distributions of IQ_{gain} and f_{CFO} for 12 devices transmitting over different BLE frequency channels. A trend similar to that observed across different receivers (Figs. 9(a–b)) is evident, indicating a generally stable distribution of the estimated impairments when signals are transmitted over wired connections.

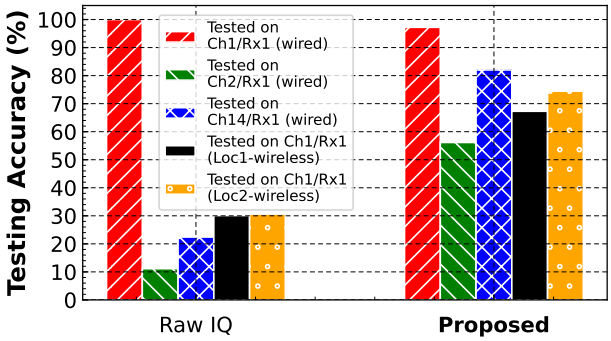


Figure 11. Classification results for 12 BLE devices when the model is trained on wired data sent over Ch1/Rx1

On the other hand, Figs. 9(e) and 9(f), which depict wireless transmissions across different locations, indicate that the wireless environment introduces both shifts and dispersion in the estimated impairments. Notably, IQ_{gain} appears to be more sensitive to these environmental variations. For example, one device (dark green), which previously exhibited a large positive outlier during wired transmission, no longer shows this behavior under wireless conditions. Additionally, the f_{CFO} values for all devices exhibit a slight rightward shift—approximately 5 kHz—while preserving their relative ordering and spacing.

Figure 11 shows the classification accuracy when the model is trained on wired data transmitted over Channel 1 and tested on the previously introduced scenarios. Despite some dispersion observed in the estimated impairments under wireless conditions, the proposed impairment-based classifier achieves accuracies of 67.3% and 74.5% when tested on wireless signals from Loc1 and Loc2, respectively—representing an improvement of approximately 40% over raw IQ-based fingerprinting methods.

Furthermore, when evaluating the model on data from different frequency channels, the proposed method continues to outperform the raw IQ-based approach. Specifically, it achieves 56% accuracy versus 11.13% on Channel 2, and 82% versus 22.4% on Channel 14.

5. Conclusion

We proposed a neural network-based framework for estimating hardware impairments of RF-based wireless communication devices. Validation of the proposed estimation framework and evaluation of its fingerprinting accuracy were conducted using real-world WiFi and BLE measurements. Results demonstrate that the impairments estimated by our model are highly accurate and can serve as robust device fingerprints—outperforming more complex deep learning-

based RF fingerprinting approaches that rely on raw IQ signals. Future work could explore using a larger model to predict impairments for unseen devices, allowing for real-world deployment to enhance signal quality and extract impairment-based signatures to enable open-set device authentication for network access authorization.

Impact Statement

This work presents a neural network-based approach for accurately estimating hardware impairments in wireless devices. The resulting estimates can enhance communication performance and serve as lightweight features for RF fingerprinting-based device identification. While this technique strengthens wireless security by enabling physical layer-based authentication and intrusion detection, it also raises potential privacy concerns if deployed without consent. To avoid misuse, we recommend deploying this method within established ethical and regulatory boundaries

References

BT-Spec-2023. Bluetooth core specification version 6.0, 2023. URL https://www.bluetooth.com/specifications/bluetooth-core-specification/. Accessed: February 03, 2025.

del Arroyo, J. A. G., Borghetti, B. J., and Temple, M. A. Fingerprint extraction through distortion reconstruction (fedr): A cnn-based approach to rf fingerprinting. *IEEE Transactions on Information Forensics and Security*, 2024.

Elmaghbub, A. and Hamdaoui, B. Eps: distinguishable iq data representation for domain-adaptation learning of device fingerprints. *arXiv preprint arXiv:2308.04467*, 2023.

Elmaghbub, A. and Hamdaoui, B. No blind spots: On the resiliency of device fingerprints to hardware warm-up through sequential transfer learning. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 134–144, 2024a.

Elmaghbub, A. and Hamdaoui, B. Distinguishable iq feature representation for domain-adaptation learning of wifi device. *IEEE Transactions on Machine Learning in Communications and Networking*, 2024b.

Fu, H., Peng, L., Liu, M., and Hu, A. Deep learning based rf fingerprint identification with channel effects mitigation. *IEEE Open Journal of the Communications Society*, 2023.

Givehchian, H., Bhaskar, N., Herrera, E. R., Soto, H. R. L., Dameff, C., Bharadia, D., and Schulman, A. Evaluating physical-layer ble location tracking attacks on mobile devices. In *2022 IEEE symposium on security and privacy (SP)*, pp. 1690–1704. IEEE, 2022.

Hamdaoui, B. and Elmaghbub, A. Uncovering the portability limitation of deep learning-based wireless device fingerprints. *arXiv preprint arXiv:2211.07687*, 2022.

Kawaguchi, K. Deep learning without poor local minima. *Advances in neural information processing systems*, 29, 2016.

Kingma, D. P. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Linz, A. and Hendrickson, A. Efficient implementation of an iq gmsk modulator. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43 (1):14–23, 1996.

McKilliam, R. G., Quinn, B. G., Clarkson, I. V. L., and Moran, B. Frequency estimation by phase unwrapping. *IEEE transactions on signal processing*, 58(6): 2953–2963, 2010.

Nikbakht, R., Jonsson, A., and Lozano, A. Unsupervised learning for parametric optimization. *IEEE Communications Letters*, 25(3):678–681, 2020.

Polak, A. C. and Goeckel, D. L. Wireless device identification based on rf oscillator imperfections. *IEEE Transactions on Information Forensics and Security*, 10(12): 2492–2501, 2015.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Schuchert, A., Hasholzner, R., and Antoine, P. A novel iq imbalance compensation scheme for the reception of ofdm signals. *IEEE Transactions on Consumer Electronics*, 47(3):313–318, 2001.

Xu, G. and Kan, E. C. Phase offset calibration in multichannel radio-frequency transceivers. *IEEE Journal of Microwaves*, 2023.

Zhang, J., Zheng, X., Liu, Q., and Lin, R. Radio frequency fingerprint identification of gmsk modulated signals based on eye diagram traces deviation. *IEEE Transactions on Vehicular Technology*, 2025.