Concise Paper: Towards On-board Radiometric Fingerprinting Fully Integrated on an Embedded System

Wenqing Yan*
Uppsala University, Sweden
wenqing.yan@it.uu.se

Mikolai Gütschow*
TU Dresden, Germany
mikolai.guetschow@tu-dresden.de

Thiemo Voigt
Uppsala University
thiemo.voigt@angstrom.uu.se

Christian Rohner
Uppsala University, Sweden
christian.rohner@it.uu.se

ABSTRACT

Radiometric fingerprinting systems leverage unique physical-layer signal characteristics originating from individual hardware imperfections to identify transmitter devices. The pure passive nature of such mechanisms entirely relieves the overhead of identification and authentication operations from the end devices, which fits well with resource-constrained applications such as wireless sensor networks. However, existing systems are limited by the need for specialized hardware and non-trivial computations to extract fingerprinting features, hindering their pervasive deployment. For the first time, we ask the question *whether it is feasible to implement an entire radiometric fingerprinting system on a low-cost and low-power embedded system on chip (SoC)*. We introduce Orf, which demonstrates the feasibility of such a system on an embedded SoC that costs less than 6 dollars. Our experiments in an anechoic chamber show that Orf achieves over 92% average accuracy in identifying one out of 32 different transmitter devices.

CCS CONCEPTS

• **Computer systems organization → Embedded software**; • **Networks → Mobile and wireless security**.

KEYWORDS

Embedded systems, Physical-layer security, Radio frequency fingerprint

1 INTRODUCTION

Recent decades have witnessed a massive increase in applications based on sensors that monitor physical environments and communicate wirelessly. These low-cost sensors are usually small embedded platforms with limited computational capabilities, memory, and communication bandwidth. Authentication and identification operations are challenging for such devices. In response, radiometric fingerprinting has emerged as a pivotal technology. Operating passively, this technique demands no additional resources from end devices, making it suitable for embedded systems. Beyond its use as a complement or replacement to cryptographic approaches, it uniquely identifies devices without relying on bit-level identities, broadening its utility in fields where unobtrusive identification is of the essence.

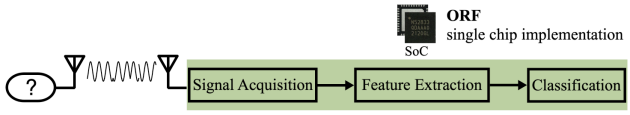

Figure 1: We demonstrate the practicability of implementing an entire radiometric fingerprinting system on a single low-power and low-cost SoC.

At its core, radiometric fingerprinting leverages the inherent transmitter imperfections that are embedded within the radiated signal. These anomalies, unique to each transmitter, form the basis of identification. However, isolating these subtle deviations demands both high-resolution wireless signal acquisition and substantial computational capacity [1–4]. Furthermore, the process requires intricate algorithms capable of effective feature extraction and subsequent classification [5, 6]. Implementing such a system on embedded devices with limited resources poses a demanding task.

Earlier work on radiometric fingerprinting predominantly relied on the use of software-defined radio and dedicated computer processing. This approach, while effective, was primarily restricted by the extensive resources and infrastructure it demanded [7]. In this work, we present and evaluate a comprehensive radiometric fingerprinting chain, from signal acquisition to feature extraction and classification, within the confines of an embedded device. Apart from simplifying the previously complex setup, this also opens up a plethora of novel application scenarios, creating unprecedented opportunities for the use of radiometric fingerprinting in diverse environments where cost, power consumption, and resource constraints are critical factors.

In this study, we present Orf, an on-board radiometric fingerprinting system that achieves practical implementation on a low-cost nRF52833 commercial off-the-shelf (COTS) system on chip (SoC) from Nordic Semiconductor. We summarize the key highlights of the results as follows:

- Orf is the *first demonstration of a complete on-board fingerprinting pipeline* including signal acquisition and feature extraction on such an affordable embedded system.

*Co-primary authors contributed equally to the work.

- The design of Orf *considers constraints in terms of memory and computational resources.* It takes 1.2 s and 12 mJ of energy to fingerprint one frame.
- The overall performance of Orf is demonstrated by achieving an *average accuracy of over 92%* on a dataset collected in an anechoic chamber consisting of 32 devices of four different types.

The code for Orf is available under the MIT license to facilitate further research and community collaboration [1]. It includes application code for data recording and on-board evaluation, scripts to facilitate batch recording, and Python code to analyze the results.

2 RADIOMETRIC FINGERPRINTING ON EMBEDDED SYSTEMS

Manufacturing imperfections and variances in hardware components such as oscillators, amplifiers, capacitors, and resistors result in unique fingerprints embedded in the wireless signal, which can be used to identify transmitter devices.

2.1 Related Work

Implementing a radiometric fingerprinting system on an embedded device involves three stages: accurate signal acquisition, energy-efficient and memory contrained fingerprint feature extraction, and classification to identify or authenticate devices (see Figure 1). Among these, signal acquisition is the primary bottleneck. Previous work estimates the signal by statistical link quality metrics, such as received signal strength indicator (RSSI) and link quality indicator (LQI) provided by radio frequency (RF) transceiver chips with sampling rates of up to 1 MS/s [8]. However, these estimates and the sampling rate is insufficient for extracting meaningful fingerprinting features. Efforts have also been made to facilitate lightweight implementations of feature extraction [9] and classification [10]. Despite these advancements in the individual stage, a complete embedded fingerprinting system has not been successfully demonstrated.

Hua et al. demonstrate fingerprinting based on channel state information (CSI) collected via a WiFi card from a laptop to identify devices [11]. The sampling approach has the potential to be implemented on embedded devices but involves computationally expensive feature extraction operations. Additionally, Jeong et al. reconstruct the physical-layer waveform built on commodity WiFi devices and showcase a fingerprinting application [12]. However, reconstructing the signal waveform from a decoded bit sequence leads to inevitable information loss, thereby limiting fingerprinting performance.

[1] https://codeberg.org/mguetschow/EWSN24-ORF

2.2 Signal Acquisition Constraints

The fingerprinting system puts relatively high requirements on the sampling rate and accuracy of the signal acquisition process. Transient-based features require in general a high sampling rate (order of GS/s [1–3]), while a sampling rate equal or slightly higher to the Nyquist rate is sufficient for steady-state-based features extracted from the modulated part of the signal [1, 5, 9]. Instead, they require high-accuracy I/Q samples. The higher the sampling accuracy, the more accurate the identification and the better the system's scalability. For this reason, fingerprinting typically requires dedicated devices such as oscilloscopes, spectrum analyzers, or software-defined radios for signal acquisition.

The rising support for the Bluetooth direction finding extension (DFE) functionality in COTS devices recently brought the capability of I/Q sampling to embedded systems [13]. DFE requires I/Q sampling to compare the phase shift between signals from antennas separated by a known distance, allowing for the estimation of the received signal's direction. Manufacturers such as Nordic Semiconductor, Texas Instruments, and Silicon Labs have made I/Q samples accessible through software APIs on SoCs. Orf demonstrates the feasibility of configuring it as a general-purpose sampling block for on-air frames and enables continuous I/Q sampling at sufficient sampling rates of up to 8 MS/s. This advancement opens up new possibilities for a wide range of applications that require in-depth analysis of the physical-layer signal, including but not limited to radiometric fingerprinting and RF sensing.

2.3 Embedded System Resource Constraints

The feature extraction and classification stages require non-trivial signal processing to be done in software. In embedded systems, memory—especially random-access memory (RAM)—is a precious resource, and its limited availability restricts the amount of data that can be kept and processed at once. Storing the signal samples, their corresponding intermediate values, and the final fingerprinting profiles has relatively high RAM requirements. On the other hand, the restricted read-only memory (ROM) space limits the type and size of the machine learning model that can be considered for the final classification stage.

Finally, the application of fingerprinting for access control or other online services requires close to real-time performance to handle smooth and efficient authentication and identification. The signal processing, feature extraction, and classification steps all present a certain computational complexity. Despite the increasing support for machine learning on modern SoCs, before this work, it has remained uncertain if an accurate fingerprint classification pipeline can be efficiently implemented on such devices.

3 DESIGN AND SYSTEM

In this section, we present Orf, a comprehensive *on-board radiometric fingerprinting* system, focusing on its architecture and the various components involved. Referring to the system design depicted in Figure 1, we start by discussing the hardware support for signal acquisition. Next, we adopt a resource-efficient pipeline to extract transmitter-specific characteristics that can be used as fingerprints. In the end, we introduce an on-board classification algorithm, which serves as the final stage of Orf.

3.1 I/Q Sampling enabled by DFE

The Bluetooth DFE allows a receiver to determine the direction of a peer device, relying on an antenna array employed either at the transmitter or at the receiver, and a constant tone extension (CTE) after the CRC. To implement Orf, we *customize the radio peripheral configuration of an nRF52833 SoC with Bluetooth DFE support* and re-purpose it as an I/Q sampling block. We select the angle of departure (AoD) mode, which disables antenna switching at the receiver. To reach the sampling rate required to recover fingerprints, oversampling is achieved using manual configuration of the sampling rate, which can be adjusted from 0.25 MS/s up to 8 MS/s. Instead of CTE, we use the option to start sampling at the beginning of the frame payload to sample the information-bearing signal. Furthermore, to obtain continuous I/Q samples, we discard guard and reference periods, thereby skipping the inherent sampling gap.

nRF52833 supports Bluetooth, IEEE 802.15.4, and proprietary radio modes at different data rates. It is feasible to enable continuous I/Q sampling for all these modes. For Orf, *IEEE 802.15.4 is selected as the physical-layer protocol.* We leave the exploration of other protocols to future work. The sampling duration is upper-bounded by a 6-bit field in the `DFECTRL1` control register, which represents multiples of 8 µs slots, to a maximum duration of 504 µs. In Orf, we transmit frames with 14 B random physical layer (PHY) payload. For each frame, a signal clip of length 466.825 µs is collected at 8 MS/s, as depicted in Figure 2, resulting in 3736 I/Q samples. Furthermore, as the nRF52833 is a single-core SoC, both sampling and processing in Orf are conducted on its ARM Cortex-M4 processor.

3.2 Feature Extraction Pipeline

In this section, we present a memory-efficient processing pipeline to extract fingerprinting features from the I/Q samples obtained in the previous stage. Considering the constraints in terms of memory and computational resources, as well as the time requirements of an on-board system implementation, Orf refers to a coherent-receiver architecture

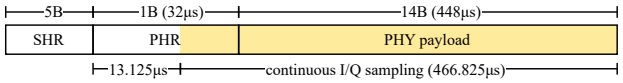

Figure 2: *Frame structure and sampling period.* Continuous I/Q samples are available from the physical header (PHR) with an offset corresponding to the DFE guard and reference period, as well as the first switch slot.

and leverages the side information of the demodulation to extract fingerprinting features for an incoming frame. Figure 3 shows a functional block diagram of such a pipeline. The fingerprinting features are either extracted from by-products or the end-product of successful receiver synchronization. Synchronization using the I/Q samples collected on-board is challenging due to the missing synchronization header (SHR) and the limited number of samples.

Normalization and Matched Filter. To maintain the signal amplitude despite variations in the received signal, an automatic gain control (AGC) is included in the hardware radio peripheral. In order to ensure stability in subsequent processing, Orf normalizes the I/Q samples in-place to their average amplitude. Following the 802.15.4 standard, a half-sine matched filter is employed through convolution to maximize the signal-to-noise ratio (SNR) and to enhance the overall signal quality. For memory-efficiency reasons, we deploy a sliding window approach for the convolution operation, only requiring a very small buffer (64 B) in RAM of a size matching the pulse shape length.

Frequency Synchronization. As the second block of the processing pipeline, the carrier frequency of the transmitter is estimated and recovered. The first fingerprinting feature is defined based on the difference among transmitters' carrier frequencies. Under restrictions of a short signal clip, Orf leverages the fast Fourier transform (FFT) algorithm in conjunction with squaring and interpolation techniques to achieve accurate frequency estimation. The 802.15.4 protocol uses an OQPSK modulation scheme with a rate of 1 MHz. The spectrum of the captured signal is spread between 0 and 1 MHz, making it difficult to estimate the carrier frequency directly. To solve this problem, we square the information-bearing OQPSK signal, which effectively removes the information on the in-phase component. This operation results in a single-tone signal at a frequency of 1 MHz. For a frequency offset $\Delta f \neq 0$, the peak is located at 1 MHz+2Δf. Additionally, Gaussian windowing and interpolation is applied to shape the spectrum, providing a high frequency resolution of less than 1 Hz [14]. The non-linear squaring and windowing transformation of the data requires an additional buffer of size 32.8 kB in RAM.

Phase and Time Synchronization. The remaining fingerprinting features are based on the stable constellation with clear clusters per symbol, which requires successful phase

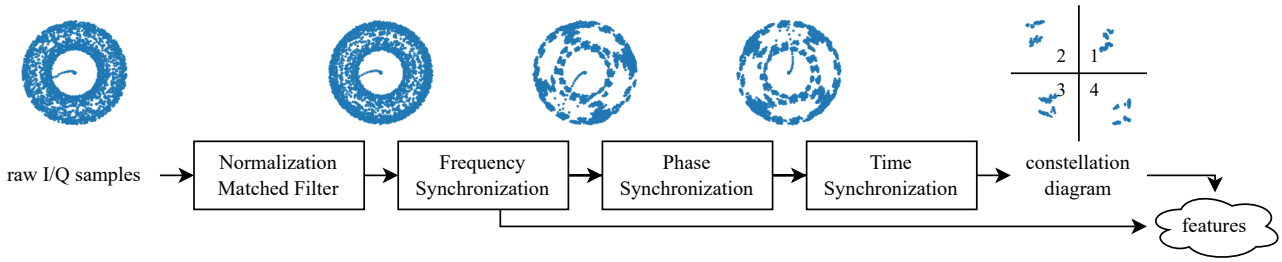

Figure 3: *The architecture of the Orf feature extractor.* The feature extraction pipeline refers to a typical coherent-receiver architecture and is based on a by-product and the result of successful receiver synchronization.

and time synchronization. After the frequency synchronization, the out-of-phase samples are rotated around the origin in the constellation. A conventional method for phase synchronization involves a digital phase lock loop (PLL), which dynamically tracks and compensates for the phase difference sample by sample. However, a PLL—being a closed-loop feedback control system—requires a non-zero acquisition time to achieve phase lock [15]. The lack of the SHR and the comparably short duration of the sampled signal present challenges in obtaining enough stable samples using a PLL. To overcome this challenge, Orf instead employs an iterative approach that traverses possible phase offsets in steps and selects the one closest to the expected constellation with zero phase offset. This approach assumes that the phase offset remains constant throughout the captured signal clip, which is justified thanks to the previous high-resolution frequency-synchronization block.

The remaining 90° phase ambiguity inherent in OQPSK signals is resolved with a similar iterative method by correlating the rotated samples with ideal samples of a known sequence. Conventionally, the SHR is used for this purpose. Since it is not part of the collected signal clip, Orf instead leverages the decoded payload from the SoC receiver module to generate the expected I/Q samples. For both parts of the phase synchronization, only a copy of the original data is rotated to avoid data loss due to rounding errors, reusing the buffer from the previous block.

In the last block, time synchronization is performed using a PLL together with a parabolic interpolator to determine the optimal sampling instant for each symbol. Thanks to successful phase synchronization, the acquisition time of this PLL is significantly shorter. Since this block already showed good results in a standard configuration, it has not been further adapted in this work.

3.3 Feature Overview

Two categories of features are used in Orf: one feature corresponds to a by-product of the receiver synchronization process, while the remaining features are extracted from the constellation diagram obtained after frequency, phase, and time synchronization.

3.3.1 Synchronization-based Features. **Carrier frequency offset (CFO).** Due to the imperfection of hardware oscillators, the actual carrier frequency varies between different transmitters. Using the receiver carrier frequency as reference, the CFO is defined as Δf.

3.3.2 Constellation-based Features. After synchronization, samples resemble a QPSK constellation diagram with one symbol cloud $\{P_i\}$ per quadrant $i \in \{1, 2, 3, 4\}$. The centroid of each cloud P_i is defined as $\hat{p_i}$. We define constellation-based features as follows:

I/Q offset (IQO) and I/Q skew (IQS). Hardware imperfections such as I/Q imbalance result in an asymmetric character of the phase and magnitude errors. IQO quantifies how far the center of the constellation diagram deviates from the origin. In addition, IQS quantifies the effect of phase imbalance, which skews the constellation diagram in the diagonal direction.

$$IQO = \frac{1}{4}\sum_{i=1}^{4}\hat{p_i} \qquad IQS = \frac{\hat{p_2}+\hat{p_4}}{2} - \frac{\hat{p_1}+\hat{p_3}}{2} \tag{1}$$

Constellation cloud shape (CCS). The shape of each symbol cloud is captured by CCS features, which are defined as the maximum magnitude and phase difference among samples in each cloud.

$$\begin{aligned} CCSM_i &= \max|P_i| - \min|P_i| \\ CCSP_i &= \max\angle P_i - \min\angle P_i \end{aligned} \tag{2}$$

Due to their similar shape, the CCS features of diagonally opposite quadrants (1,3 and 2,4) are combined by averaging.

Error vector magnitude (EVM). The error vector of each symbol is defined as $e[k] = s[k] - \hat{p_i}$ to the corresponding cloud centroid $\hat{p_i}$. $s[k]$ is the collected symbol at timestamp k and N is the number of symbols in the collected signal clip.

$$EVM = \sqrt{\frac{\frac{1}{N}\sum_{k}^{N}|e[k]|^2}{\sum_{i=1}^{4}\hat{p_i}}} \tag{3}$$

3.4 On-board Fingerprint Classification

As the last stage of Orf, a random forest classifier (RFC) distinguishes between transmitter devices based on the extracted fingerprinting features. RFC is known for its speed and capability to handle large datasets, and has shown effectiveness in previous research [4]. In our work, we first train

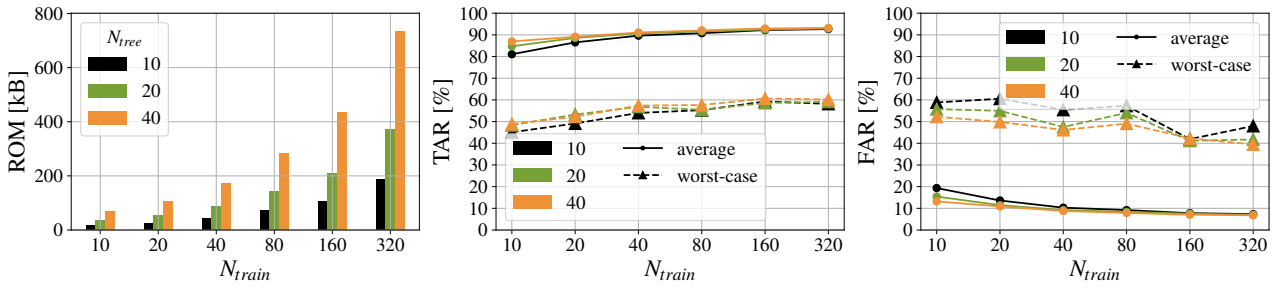

Figure 4: *Memory footprint and classification performance of RFC with different configurations.* Both are positively correlated with N_{tree} and N_{train}. The number of subtrees has a higher impact on the model size, while not significantly improving the classification performance.

the RFC model with pre-collected fingerprint samples on an x86-64 Linux computer using `scikit-learn` in Python. Once the model is well-trained, it is converted to C99 code using `emlearn` [16]. For on-board deployment, the model complexity is upper-bounded by the limited ROM space available on the SoC. Meanwhile, a too simple model may fail to capture the complexity of the data, resulting in poor performance. Therefore, in Section 4.1, we will jointly evaluate the performance and model size of different RFC configurations.

4 EVALUATION

We perform experiments to evaluate Orf from two aspects: the classification performance and the resource consumption in terms of memory occupation, processing time and energy consumption.

Data Collection. We fingerprint four different types of COTS sensor boards, with eight individual transmitters per device type: *TelosB* (Texas Instruments CC2420 radio, integrated PCB antenna), Thunderboard *Gecko* (Silicon Labs EFR32 SoC, ceramic antenna), nRF52840 *DK*, and nRF52840 *Dongle* (both Nordic Semiconductor SoCs, PCB antenna). We collect signals in an anechoic chamber under ideal channel conditions to avoid any impact of the wireless channel. We use one nRF52833 DK board as receiver and place each transmitter at a distance of 1.3 m. We collect 1000 frames per device and conduct fingerprint processing on board.

4.1 Classification Performance

To find an appropriate balance between performance and memory usage, the converted model size in C and the classification performance are jointly evaluated for different RFC model configurations. Figure 4 presents the results, which are obtained using the same test set and training strategy while varying the number of subtrees N_{tree} and the amount N_{train} of frames per transmitter device used during training (the depth of the RFC subtrees depends on the number of training samples). The classification performance is given in terms of true accept rate (TAR) and false accept rate (FAR). In fingerprinting systems, the TAR represents the probability of correctly classifying a device, while FAR is the probability of incorrectly classifying an imposter as a certain device. Both average and worst-case values of TAR and FAR are reported for the entire device set. In this work, we use all extracted features as inputs to the classifier. Further investigation into feature engineering and selection is left for future research.

The results show that the average TAR converges to a maximum value of around 93% with a large training set, which indicates the effectiveness of the selected fingerprinting features. To provide further insights, we consider the configuration $(N_{\text{tree}}, N_{\text{train}}) = (20, 160)$. In this configuration, the average TAR is above 92% with a low average FAR of 7%. Most of the devices are correctly classified with 100% TAR, but three device pairs exhibit similar fingerprints and are highly misclassified. Therefore, the worst-case TAR is only 58% and the worst-case FAR is 46%. Excluding one device out of each pair of devices results in more than 99% average TAR and below 1% FAR. For the task of device type classification, all model configurations accurately classify the transmitter device types, with an average TAR of over 98%.

4.2 Resource Consumption

When discussing the resource consumption of Orf, it is worth noting that the objective of this work is to assess the feasibility of implementing a complete fingerprinting system on a single low-power SoC. Optimization for minimal resource consumption is not the primary focus.

Memory Usage. When deploying a fingerprinting system on an embedded SoC with limited memory, the memory footprint of the application becomes a critical consideration. The chosen nRF52833 SoC provides 512 kB of flash/ROM memory and 128 kB of RAM. The I/Q samples of each incoming frame require 30 kB RAM for storage as `float32_t` values. In the feature extraction pipeline, subsequent blocks can reuse the memory space from previous blocks. By summing the I/Q sample buffer size with the maximum RAM usage in the pipeline, the total RAM space required by Orf is calculated to be 62.8 kB. This is less than half of the available RAM, allowing for practical deployment alongside other functionalities. On the other hand, the size of the classifier model is limited by the available ROM space. The code and static

data for the rest of the Orf pipeline occupy approximately 100 kB, leaving a maximum of 400 kB for the classifier to fit within. Out of various model configurations discussed in the previous section, we choose $(N_{\text{tree}}, N_{\text{train}}) = (20, 160)$ occupying 210.4 kB.

Processing Time and Energy Consumption. The processing time and energy consumption of Orf is investigated using the following experiment: The pipeline is run 100 times, monitoring the current draw using the nRF Power Profiler Kit II in source meter mode with a constant supply voltage of 3 V. The measurements are attributed to individual pipeline blocks, indicated by a pulse on a GPIO pin. Orf takes on average 1.2 s to fingerprint one incoming frame. The energy consumption of the whole pipeline amounts to 12.3 mJ, which is the equivalent of active listening on the channel for roughly 0.6 s.

5 DISCUSSION

Compatibility. Orf demonstrates one fingerprinting system example for IEEE 802.15.4 utilizing the nRF52833 SoC. The implementation of Orf is facilitated by the hardware APIs inherent within the existing DFE functionality. For the nRF52833 platform, I/Q samples can be collected in bluetooth low energy (BLE) mode as well. With access to the physical layer waveform, diverse feature extraction blocks can be implemented. In addition, the DFE functionality has been adopted by other manufacturers, including Texas Instruments and Silicon Labs, encompassing notable chipset series such as CC2642, CC2652, EFR32, BG22, and BG24. Through our correspondence with manufacturers' technical support, it has been affirmed that all these platforms are capable of conducting I/Q sampling at a rate up to 8 MHz. This suggests that Orf can be seamlessly transitioned to such platforms.

Practical System Design Challenges. In this work, we focus on the embedded system aspect and acknowledge that other practical system design aspects align with established research. First, detecting unknown devices is crucial, which can be addressed through advanced classifier design [17]. Second, the fingerprint is prone to environmental impacts like complex multipath channels and co-channel interference. Several works studied and proposed robust signal processing modules to mitigate these effects[4]. Orf can potentially be extended to integrate the aforementioned designs. The primary bottleneck in the current implementation is memory usage, mainly due to the classification model using a simple RFC algorithm in C, which occupies several hundred kB. Strategies such as optimized neural network architectures, model compression techniques, and specialized frameworks can be considered to address this. These optimizations can free up memory for more complex classification models that support robustness. In terms of scalability, an unpublished experimental study using the same features, but an USRP and an off-board processing pipeline achieved approximately 97% TAR with 140 devices. Compared to this study, we anticipate slightly reduced scalability for on-board fingerprinting due to the embedded signal acquisition and simplified feature extraction.

6 CONCLUSION

Orf showcases an on-board radiometric fingerprinting system that is fully integrated on low-cost and low-power COTS hardware. The signal acquisition, processing, and identification of a single frame takes roughly one second and consumes the equivalent energy of active channel monitoring for half a second. Orf achieves an average TAR of over 92% on a challenging dataset with 32 devices.

ACKNOWLEDGMENTS

This work was funded by SSF, the Swedish Foundation for Strategic Research, and the Swedish Research Council.

REFERENCES

[1] Vladimir Brik et al. Wireless device identification with radiometric signatures. In *Proceedings of MobiCom*, 2008.
[2] Boris Danev et al. Transient-based identification of wireless sensor nodes. In *Proceedings of IPSN*, 2009.
[3] Adam C Polak et al. Identifying wireless users via transmitter imperfections. *IEEE Journal on selected areas in communications*, 2011.
[4] Wenqing Yan et al. RRF: A robust radiometric fingerprint system that embraces wireless channel diversity. In *Proceedings of WiSec*, 2022.
[5] Linning Peng et al. Deep learning based RF fingerprint identification using differential constellation trace figure. *IEEE Transactions on Vehicular Technology*, 2019.
[6] Xinyu Zhou et al. A robust radio-frequency fingerprint extraction scheme for practical device recognition. *IEEE Internet of Things Journal*.
[7] Naeimeh Soltanieh et al. A review of radio frequency fingerprinting techniques. *IEEE Journal of Radio Frequency Identification*, 2020.
[8] Carsten Herrmann et al. RSSISpy: Inspecting concurrent transmissions in the wild. In *Proceedings of EWSN*, 2022.
[9] David A Knox et al. Wireless fingerprints inside a wireless sensor network. *ACM Transactions on Sensor Networks*, 2015.
[10] Di Liu et al. RF fingerprint recognition in resource-constrained edge computing. In *Proceedings of ICCWAMTIP*, 2021.
[11] Jingyu Hua et al. Accurate and efficient wireless device fingerprinting using channel state information. In *Proceedings of INFOCOM*, 2018.
[12] Woojae Jeong et al. SDR receiver using commodity WiFi via physical-layer signal reconstruction. In *Proceedings of MobiCom*, 2020.
[13] Martin Woolley. Bluetooth direction finding. Technical report, Bluetooth SIG, 2019.
[14] M Gasior et al. Improving FFT frequency measurement resolution by parabolic and gaussian spectrum interpolation. In *AIP Conference Proceedings*. American Institute of Physics, 2004.
[15] Michael Rice. *Digital communications: a discrete-time approach*, pages 730–731. Pearson Education India, 2009.
[16] Jon Nordby. emlearn: Machine learning inference engine for microcontrollers and embedded devices, 2019.
[17] Saptarshi Hazra et al. PLIO: Physical layer identification using one-shot learning. In *Proceedings of MASS*, 2021.