POSTER: Robust Deep-learning-based Radio Fingerprinting with Fine-Tuning

Haipeng Li
University of Cincinnati, USA
li2hp@mail.uc.edu

Chenggang Wang
University of Cincinnati, USA
wang2c9@mail.uc.edu

Nirnimesh Ghose
University of Nebraska-Lincoln
nghose@unl.edu

Boyang Wang
University of Cincinnati, USA
boyang.wang@uc.edu

ABSTRACT

Minute hardware imperfections in the radio-frequency circuitry of a wireless device can be leveraged as a unique fingerprint. Radio fingerprinting is a way of distinguishing a device from others of the same type at the physical layer by utilizing these hardware imperfections. Recent studies proposed to utilize deep learning over raw I/Q data for the purpose of radio fingerprinting and achieve high accuracy. Unfortunately, deep-learning-based radio fingerprinting is not robust over I/Q data across different days due to significant changes in wireless channels. This study proposes to leverage fine-tuning to improve the robustness of radio fingerprinting in a cross-day scenario, where training and test I/Q data are from different days. Our experimental results suggest that transfer learning is a promising approach for robust deep-learning-based radio fingerprinting in practice.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

ACM Reference Format:
Haipeng Li, Chenggang Wang, Nirnimesh Ghose, and Boyang Wang. 2021. POSTER: Robust Deep-learning-based Radio Fingerprinting with Fine-Tuning. In *14th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '21), June 28–July 2, 2021, Abu Dhabi, United Arab Emirates.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3448300.3468253

1 INTRODUCTION

Background. Due to the imperfection of radio-frequency circuitry in manufacturing, a wireless device can carry hardware imperfections when it transmits wireless data. These hardware imperfections include I/Q imbalance, phase noise, frequency offset, sampling offset, and harmonic distortions [7]. *Radio Fingerprinting* is a way of distinguishing a device from others of the same type over I/Q data by taking advantage of these hardware imperfections. Specifically, given I/Q data collected on a receiver side, a receiver decides which transmitter it is in radio fingerprinting.

Recent studies [3, 4, 6, 7] in radio fingerprinting utilize deep neural networks and obtain promising results. Al-Shawabka et. al. [4] collected large-scale datasets from 20 USRPs over several days and evaluated three different convolutional neural networks over their datasets as well as private WiFi and ADS-B datasets provided by DARPA. Two studies [3, 6] showed that complex-valued deep neural networks can outperform real-valued deep neural networks in radio fingerprinting.

Limitations in Current Studies. Despite the promising results reported in recent studies, deep-learning-based radio fingerprinting is ***not robust***. Specifically, if a neural network is trained based on I/Q data from one day and tested based on data from a different day, the accuracy can drop dramatically [4, 6, 7]. The poor performance of *cross-day* radio fingerprinting is because the conditions of the indoor wireless channels change significantly across two different days. The performance will be much worse for the fast-changing outdoor channels. These changes can affect I/Q data, and therefore significantly affects a neural network's capability of identifying hardware imperfections of each transmitter from I/Q data. Recollecting large amounts of I/Q data from a new day and retraining the entire neural network is a straightforward approach to regain high accuracy. However, it is time-consuming and not scalable.

Our Main Idea. To improve the robustness of deep-learning-based radio fingerprinting in a cross-day scenario, we propose to leverage *fine-tuning*, which is a transfer learning technique. Specifically, given a neural network trained with I/Q data from a previous day, we re-tune the weights of the last layer of the neural network with a small amount of I/Q data collected from a new day. The weights of the other layers of the neural network are frozen during fine-tuning. After fine-tuning, the updated neural network is used to perform radio fingerprinting over I/Q data from the new day.

The intuition behind our approach is that the high-level features learned from the same group of transmitters by the first few layers of a neural network can still be used even I/Q data are from a new day. The last layer is tuned to derive the best possible accuracy for radio fingerprinting over I/Q data of the new day. By leveraging fine-tuning, our approach does not need to recollect large-scale data or retrain the entire neural network.

Our Results and Findings. The contributions of this study can be summarized as below.

(1) We build a testbed, including 1 receiver and 5 transmitters, by using Software Defined Radio with HackRF Ones. We collect a dataset of 5 transmitters across two different days.

WiSec '21, June 28–July 2, 2021, Abu Dhabi, United Arab Emirates
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8349-3/21/06.
https://doi.org/10.1145/3448300.3468253

(2) We validate that deep-learning-based radio fingerprinting can derive good accuracy (e.g., 82%) in the same-day scenario and perform poorly (e.g., 29%) in the cross-day scenario.
(3) By leveraging fine-tuning with a small number of I/Q data from a new day, we can significantly regain the accuracy (e.g., 76%) in the cross-day scenario.

Reproducibility. We make our source code and dataset publicly available [1] to the community so that others can reproduce our results and further expand the research.

2 PERFORMANCE EVALUATION

Testbed Setup. We setup a testbed with 1 receiver and 5 transmitters as shown in Fig. 1. Each receiver/transmitter is a HackRF One (with ANT500 antenna) running with GNU Radio. We leverage the open-source GNU Radio code from [2] to establish WiFi transmissions (IEEE 802.11 a/g) with BPSK 1/2 modulation scheme between the receiver and transmitter. Additional details of the code can be found in [5]. We customized some parameters to work with our hardware. Specifically, we captured the I/Q data at 2.45 GHz center frequency with 2MHz bandwidth and a 2MHz sampling rate.

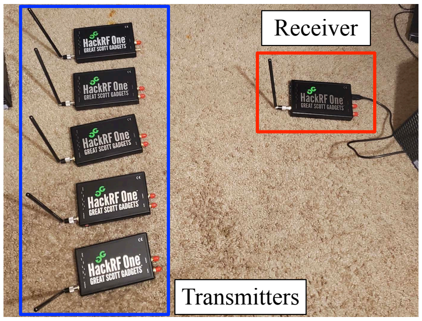

Figure 1: Our testbed with 1 receiver and 5 transmitters. Each receiver/transmitter is a HackRF One running GNU Radio.

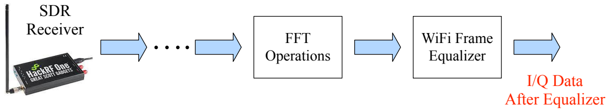

Figure 2: I/Q data collection in GNU Radio at the Receiver. We collected I/Q data after WiFi Frame Equalizer.

During the data collection, the receiver remains the same and is on all the time. Only one transmitter is on each time. The transmitters were about 3 feet away from the receiver. The devices are static, i.e., we did not move the receiver or transmitters when we were collecting data. The data collection was running in an indoor environment for two days. On Day 1, we collected 3 transmissions from each transmitter, where each transmission lasted for 30 seconds. Between each transmission, the transmitter was idle for 15 seconds. All the transmitters sent the same data repeatedly. As suggested in [4], we recorded I/Q data after WiFi Frame Equalizer at the receiver side (as shown in Fig. 2) for the purpose of radio fingerprinting. We repeated the same process with our testbed on Day 2. We refer to these I/Q data collected directly from our testbed as raw I/Q data.

Our Dataset. Based on the raw I/Q data we recorded, we form a dataset for the purpose of machine learning by following the same approach introduced in [4]. Specifically, for raw I/Q data from Day 1, we randomly select 100,000 I/Q traces from all three transmissions per transmitter. Each I/Q trace includes consecutive 288 I/Q samples, where the in-phase part and quadrature part are considered as two separate channels. In essence, each I/Q trace is a 2-dimensional time-series data with a length of 288. Each I/Q trace is considered as one input sample for a deep neural network. Overall, we obtain 500,000 I/Q traces from Day 1. We repeat the same process to obtain 500,000 I/Q traces from Day 2.

INFOCOM20 Dataset. We also leverage a subset of a public dataset collected in [4]. We denote it as INFOCOM20 Dataset in this paper. The INFOCOM20 dataset was collected with USRPs. We leverage the data from their "Setup 1" [4], which consists of I/Q data after WiFi Frame Equalizer. We only choose five devices (device 1 to device 5) with 100,000 I/Q traces per device for easy comparison with our dataset. Overall, we use 500,000 I/Q traces from Day 1 and 500,000 I/Q traces from Day 2 of INFOCOM20 dataset. Their raw I/Q data includes 20 devices with 10 transmissions per device per day for 2 days. More details can be found in [4].

Neural Networks. We examine three neural networks, including Homegrown, Baseline, and RestNet-50, described in [4]. All three are Convolutional Neural Networks. Homegrown is a simple one consisting of 2 convolutional layers and 2 fully connected layers. The baseline includes 5 repeated blocks with 2 convolutional layers and 1 max-pooling layer per block. RestNet-50 includes 50 layers. More details can be found in [4]. We implemented the three neural networks according to the description in [4].

ML Evaluation Setting. We use Keras as the front end and Tensorflow as the back end. We run all the experiments on a Linux machine with Ubuntu 18.04, 2.8 GHz CPU, 64 GB RAM, and an Nvidia Titan RTX GPU. Given a dataset, unless specified, we use 72% for training, 8% for validation, and 20% for testing. We perform 5-fold cross-validations. We train each neural network for at most 100 epochs or stop the training earlier if the training accuracy does not improve for 10 consecutive epochs.

Experiment 1: Same-Day Radio Fingerprinting. We first examine the performance of radio fingerprinting in the same-day scenario, where the training, validation, and test are from the same day. All the accuracy results are reported based on test data. As shown in Table 1, all the three models obtain good accuracy, which is much higher than random guessing. Specifically, Homegrown can achieve more than 82% accuracy on our data from Day 2. The three models also obtain reasonable accuracy over INFOCOM20 dataset as shown in Table 2.

Note that, unlike neural networks for image recognition which can easily achieve more than 95% accuracy over millions of training samples, neural networks for radio fingerprinting are much harder to learn as the conditions of wireless channels keep changing in practice. These changes affect the I/Q data collected at the receiver as well as the features learned by neural networks. The recent studies [3, 4, 6, 7] in deep-learning-based radio fingerprinting normally obtain 50%~85% accuracy depending on the neural networks, testbeds, and datasets.

About Overfitting. One important thing we would like to point out is that, for radio fingerprinting, a deep neural network can be easily over fitted, where the validation accuracy can be extremely high (e.g., greater than 97% in our experiments) but the test accuracy

Table 1: Average accuracy (%) of Same-Day Radio Fingerprinting (our dataset; 5 devices)

	Homegrown	Baseline	RestNet-50
Day 1	60.0	61.5	60.0
Day 2	82.1	64.4	62.4

Table 2: Average accuracy (%) of Same-Day Radio Fingerprinting (INFOCOM20 dataset; 5 devices)

	Homegrown	Baseline	RestNet-50
Day 1	73.8	53.6	69.8
Day 2	97.9	37.2	63.3

Table 3: Average accuracy (%) of Cross-Day Radio Fingerprinting (training and validation: Day 1; test: Day 2)

	Homegrown	Baseline	RestNet-50
Our dataset	29.6	29.9	37.5
INFOCOM20 dataset	19.1	21.0	22.1

is much lower (e.g., the ones we reported in the tables). This is likely because, for a neural network, it is easier to learn wireless channel features rather than the hardware imperfections of a device. To obtain reliable results for radio fingerprinting, it is important to have validation data and also perform cross-validations.

Experiment 2: Straightforward Cross-Day Radio Fingerprinting. We evaluate the performance of radio fingerprinting in a cross-day scenario, where the training and validation data are from the same day (e.g., Day 1) but the test data are from a later day (e.g., Day 2). All the accuracy results are reported based on test data.

As we can observe from Table 3, the accuracy of all the three neural networks over the two datasets drop significantly to or close to the level of random guessing (i.e., 20%). This suggests that the trained neural networks are not robust in the cross-day scenario. This observation is consistent with results in recent studies [4, 6, 7].

Experiment 3: Cross-Day Radio Fingerprinting with Fine-Tuning. In this experiment, we still examine the performance of radio fingerprinting in a cross-day scenario. Different from the last experiment, fine-tuning is performed, where a neural network is trained based on data from Day 1 but fine-tuned with data from Day 2 before testing. Specifically, we tune the weights of the last layer of each neural network trained by Day 1 with a small amount of data from Day 2. Given data from Day 2, we use parameter N to denote the number of traces per class/transmitter we use for fine-tuning. When $N = 0$ in our tables, it is equivalent to straightforward cross-day radio fingerprinting without using fine-tuning. The number of traces per transmitter for testing from Day 2 is still 20,000 (i.e., 20% of the dataset from Day 2) as in the previous experiment.

Our results in Table 4 and Table 5 show that for Homegrown and Baseline, fine-tuning can indeed improve the accuracy of radio fingerprinting in the cross-day scenario without the need of retraining the entire neural network over large-scale I/Q data. For example, with $N = 1,600$, Homegrown can increase accuracy from 29.6% to 76.6% over our dataset and from 19.1% to 67.9% over INFOCOM20 dataset in the cross-day scenario.

Table 4: Average accuracy (%) of Cross-Day Radio Fingerprinting with Fine-Tuning (our dataset; 5 devices; training and validation: Day 1; test: Day 2)

N	0	50	100	200	400	800	1,600
Homegrown	29.6	35.3	44.6	50.2	54.5	66.3	76.6
Baseline	29.9	54.9	58.3	58.4	58.6	60.0	61.1
RestNet-50	37.5	26.5	33.9	30.8	34.9	36.1	36.2

Table 5: Average accuracy (%) of Cross-Day Radio Fingerprinting with Fine-Tuning (INFOCOM20 dataset; 5 devices; training and validation: Day 1; test: Day 2)

N	0	50	100	200	400	800	1,600
Homegrown	19.1	23.9	31.6	38.7	42.3	54.2	67.9
Baseline	21.0	34.9	36.5	37.2	38.1	40.9	43.0
RestNet-50	22.1	25.6	25.6	25.2	26.2	24.0	25.3

For RestNet-50, fine-tuning is, however, not effective in terms of improving the accuracy in the cross-day scenario. This is likely because RestNet-50 is very deep (i.e., including 50 layers) and we only fine-tuned the one last layer. Fine-tuning more layers in RestNet-50 could be helpful. We will leave it as our future work.

3 CONCLUSION AND FUTURE WORK

We demonstrate that fine-tuning can be an effective but lightweight way to improve the robustness of radio fingerprinting as wireless channels changes in practice. There are several directions we can further improve in our future work. (1) We plan to expand our testbed with a much greater number of devices (e.g., 30~50). (2) More advanced transfer learning methods, such as triplet networks, can be explored. Data augmentation can be also leveraged to boost the performance of transfer learning in the context of radio fingerprinting. (3) How to perform transfer learning over complex-valued neural networks will also be an interesting direction to investigate to promote the robustness of radio fingerprinting.

Acknowledgement. The authors would like to thank the anonymous reviewers for their insightful comments. We would like to also thank Amani Al-Shawabka and Dr. Francesco Restuccia for explaining the details of their experiments in [4]. The UC authors were partially supported by National Science Foundation (CNS-1947913).

REFERENCES

[1] [n.d.]. https://github.com/SmartHomePrivacyProject/RadioFingerprinting

[2] [n.d.]. IEEE 802.11 a/g/p - WiFi. https://www.wime-project.net/features/

[3] I. Agadakos, N. Agadakos, J. Polakis, and M. R. Amer. 2020. Chameleons' Oblivion: Complex-Valued Deep Nerual Networks for Protocol-Agnostic RF Device Fingerprinting. In *Proc. of IEEE Euro S&P'20.*

[4] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia. 2020. Exposing the Fingerprint: Dissecting the Impact of the Wirelss Channel on Radio Fingerprinting. In *Proc. of IEEE INFOCOM'20.*

[5] B. Bloessl, M. Segata, C. Sommer, and F. Dressler. 2013. An IEEE 802.11a/g/p OFDM receiver for GNU radio. In *Proc. of the 2nd workshop on Software radio implementation forum (SRIF'13).*

[6] M. Cekic, S. Gopalakrishnan, and U. Madhow. [n.d.]. Wireless Fingerprinting via Deep Learning: The Impact of Confounding Factors. ([n. d.]). https://arxiv.org/pdf/2002.10791.pdf.

[7] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia. 2019. DeepRadioID: Real-time Channel-Resilient Optimization of Deep Learning-Based Radio Fingerprinting Algorithms. In *Proc. of ACM MobiHoc'19.*