Learning-Based RF Fingerprinting for Device Identification using Amplitude-Phase Spectrograms

Abdullahi Mohammad, *Member, IEEE,* Mateen Ashraf, *Member, IEEE,* Mikko Valkama, *Fellow, IEEE* and Bo Tan, *Member, IEEE,*

2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall) | 979-8-3503-2928-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/VTC2023-Fall60731.2023.10333639

***Abstract*—Radio frequency fingerprinting (RFF), a technique based on specific transmitter hardware impairments, has emerged as an effective solution for wireless device identification. In this paper, we present a flexible deep CNN-LSTM for RF feature extraction capable of handling inputs with varying lengths. We construct a channel-independent spectrogram by exploiting the amplitude and phase information of the received RF signals, ensuring the extractor's resilience to channel variations. To evaluate the performance of the proposed approach, we utilize the open-source LoRa dataset consisting of 60 commercial off-the-shelf LoRa devices and a USRP N210 software-defined radio platform. The experimental results show that classifiers perform better when trained with RF templates generated from amplitude-phase spectrogram than amplitude-only spectrogram. This is due to the additional information present in the amplitude-phase channel-independent spectrogram.**

***Index Terms*—RF Fingerprinting, Spectrogram, Deep Neural Networks, Device Identification, Feature Extraction.**

I. Introduction

BEYOND 5G or 6G system is the platform to support the ultra-dense and massive connectivity on the radio frequency (RF) spectrum in the machine-type communications [1]. This increased number of connected devices has inspired manufacturers to develop smaller, cheaper devices with higher energy efficiency. These devices employ wireless communication protocols such as long-range wide-area networks (LoRaWAN) and Narrowband Internet of things (NB-IoT) that can transmit data over long distances while consuming very little energy [2]–[4]. From the security perspective, however, these devices are prone to attacks. Typically, IoT wireless protocols use cryptographic techniques, which depend on digital signatures to ensure network security [5]. The drawback of cryptographic approaches is that they cannot accommodate longer-length digital signatures to guarantee the security of the data transmission due to the limited physical layer bandwidth and hardware computational resources [6], [7]. Several efficacious invasions, such as spoofing and packet sniffing attacks [2], [8], rogue device access [8], network jamming [9], encryption cracking, etc. [7] have been reported lately to target these vulnerable protocols.

Radio frequency fingerprinting (RFF) is a noncryptographic technique that exploits the device's unique intrinsic characteristics in the RF signal resulting from the hardware impairments of RF circuits [7]. The RFF has been regarded as an alternative authentication strategy at the physical layer of wireless communications over the digital information such as media access control (MAC) address or the cryptographic key to infer the identity of a radio device. It has surfaced as a practical solution for certifying IoT devices. Extensive research has been conducted on RFF to enhance the security in different radio systems such as spectrum sharing system [10], cognitive-communication networks [2], the IoT [11], 5G and open radio access networks [12], [13], linear frequency modulation radar [14], [15].

Currently, research on RFF-based identification (RFFI) can be divided into two categories: conventional RFF extractor-based and learning-based approaches. The former category uses custom-designed RFF extractors to capture hardware imperfections, such as power spectral density [16], beam pattern [17], IQ imbalance, carrier frequency offset (CFO), power amplifier non-linearity [4], etc. However, these approaches heavily rely on the quality of the feature extraction algorithms and require an in-depth understanding of the RF circuits involved. Additionally, some hardware characteristics are interrelated, which makes it challenging to extract each feature individually. Conversely, learning-based methods utilize classification neural networks (NNs) to directly process raw signals and deduce the device's identity without requiring explicit feature engineering. These approaches have gained significant attention in recent years due to their ability to learn relevant features from the data automatically [2]–[4], [6], [7].

A learning approach is proposed in [2] to detect unique and repeatable signatures to address the device identification and verification challenges in cognitive radio networks using a convolutional neural network. RF signal modelling of hardware impairments and CNN-based RFFI protocol is presented for device authentication in [4]. While the impairment model is extended to the receiver side, it lacks domain generalization. To address this, a parametric impairment model is developed [6] to optimize the signal representation of RF fingerprints for learning algorithms. Zhang *et al.* [3] introduces an adaptive RF fingerprints fusion network (ARFNet) that utilizes a dual attention convolution layer that learns and fuses distinctive features in RF signals from various transmitters in a data-driven manner, aiming to obtain more discriminant features for improved recognition performance of the wireless device.

In these works, the impact of wireless channels on the quality of RFF is not considered, and the techniques are not scalable because they can only classify devices present during training. The paper [7] proposes a deep learning-based RFFI framework that is both scalable and resilient to channel variations to train an NN that can extract RFFs from previously unseen devices and enable efficient enrollment and maintenance of an RFF database. The framework leverages channel-independent features based on the received signal amplitude value and data augmentation techniques to address the impact of wireless channel distortion on RFF feature extraction. However, the phase component of the RF signal is neglected in the above works.

In this work, we propose a deep learning RF feature extractor that leverages the combined effects of amplitude and phase information spectrograms to enhance the model's robustness to channel variations, a novel approach not previously explored. The architecture employs a CNN Long Short-Term Memory (CNN-LSTM) network for RFF feature extraction. CNN layers are utilized to extract features from input data, and LSTMs are used to support sequence prediction to handle inputs with varying lengths. The paper's major contributions are summarized as follows:

- We introduce the phase unwrapping technique to exploit the phase of the received RF signal to extract additional information.
- To enhance the robustness of the RF extractor model to channel variations, we construct a channel-independent spectrogram that combines amplitude and phase information.
- To support sequence prediction to handle inputs with varying lengths, we use the CNN-LSTM network to build an RFF feature extraction model.

II. System Description and Signal Capture

We build our work on LoRa devices becasue of their wide adoption in many IoT applications. Also, the low-power and long communication range leads to significant signal attenuation. Therefore, its characterized by sufficient hardware impairments, which makes it suitable for RFFI. In this work, we adopt the dataset of [7] [1], where the transmitted LoRa baseband signal $x(t)$ undergoes a sequence of signal processing (up-conversion) via RF circuits, such as oscillator and power amplifier, etc. These components possess distinct impairments; their combined impact is injected into the RF signal and is defined by $\mathcal{F}(\cdot)$. Subsequently, the signal propagates through the wireless channel and is received by the receiver, and is expressed as:

$$y(t) = h(t) * \mathcal{F}\left(x(t)\right) + n_0(t) \tag{1}$$

where $h(t)$ is the wireless channel response, $*$ is convolution operation $\mathcal{F}(\cdot)$ represents the overall effect of the hardware impairments and $n_0(t)$ is the additive white Gaussian noise. At the receiver, $y(t)$ is converted into digital samples using an analog-digital converter (ADC) at a sampling interval of T_s, defined as $y[n]$.

[1]https://ieee-dataport.org/open-access/lorarffidataset

A. Preprocessing

The received signal must be preprocessed by applying synchronization, CFO compensation, and normalization to fulfill the essential criteria of RFFI. The synchronization aligns receiver and transmitter time, while CFO compensation corrects frequency differences between transmitted carrier and receiver's local oscillator. The reader is referred to [7], [18] for the details of these methods. It is commonly believed that LoRa transmissions undergo flat fading [7]. However, experiments reveal that the wireless channel introduces considerable distortion to the LoRa signal [18]. Furthermore, the non-stationary nature of the LoRa signal requires analysis in the time-frequency domain. The Short-time Fourier transform (STFT) is an effective algorithm for time-frequency analysis, capable of revealing the signal's temproal-frequency features.

III. Proposed Method

To mitigate the effect of the wireless channel on the received signal, STFT is performed on it by dividing $y[n]$ into K segments of N samples to obtain the channel-independent spectrogram based on signal amplitude and phase information.

A. Phase Channel Independent Spectrogram

The STFT produces time-frequency representation of the signal, decomposing it into its constituent frequency components over time [19]. Signal analysis involving STFT has always been presented as a power spectrogram using amplitude value, where only the amplitude of the STFT is considered [20]. However, phase, which is an essential source of information, is left out in such an approach. The phase information can be obtained from the frequency bin computed from the phase difference between each time slice of the STFT, showing the deviation in the instantaneous frequency of each spectral component. [21]. In the discrete-time domain, the received signal could be broken into frames presented in the magnitude and phase for each point in time and frequency as its equivalent STFT expressed as:

$$Y_{m,k} = \sum_{n=0}^{N-1} y[n]w[n - mL]e^{-j\pi \frac{k}{N} n} \tag{2}$$

where $Y_{m,k}$ is the STFT element of the complex matrix $\mathbf{Y} \in \mathbb{C}^{N \times K}$. $w[n]$ denotes the spectral window function, $k = 1, 2, \cdots, K$, $m = 1, 2, \cdots, N$ and L is the hope size.

B. Phase Unwrapping

The phase information within STFT of a signal is noisy and can be affected by phase noise, CFO, and sampling time offset. Techniques, such as frequency estimation, phase correction, cyclic prefix, blind estimation, etc. [4]

are commonly used to compensate CFO. One way to avoid phase noise is to use only the signal amplitude. This proves promising in mitigating the channel distortion, but it leaves out the phase component of the signal. In this work, phase unwrapping is not mainly used to combat CFO but may improve CFO compensation accuracy and provide additional information about the received signal. The phase values of the STFT are usually wrapped within the range $[-\pi, \pi]$ or $[0, 2\pi]$. The phase unwrapping technique is typically applied to the phase values obtained from STFT of a signal.

The resulting phase from the STFT is wrapped and obtained from the relation

$$\bar{\theta}_{m,k} = \text{atan}\left(\Im(Y_{m,k}), \Re(Y_{m,k})\right) \tag{3}$$

Suppose $P(\omega, t)$ be the wrapped phase value at frequency bin ω and time frame t, and $U(\omega, t)$ be the unwrapped phase value. Then the unwrapped phase can be obtained iteratively using the following expression

$$U(\omega, t) = P(\omega, t) + 2\pi C(\omega, t) \tag{4}$$

where $C(\omega, t)$ is the unwrapping integer variable that keeps track of the accumulated phase jumps, and is determined based on the neighboring phase values as follows:

$$C(\omega, t) = \text{floor}\left(\frac{U(\omega, t-1) - P(\omega, t)}{2\pi}\right) + C(\omega, t-1) \tag{5}$$

Moreover, (4) can be elegantly written as

$$\theta_{m,k} = \bar{\theta}_{m,k} + \beta_{m,k} 2\pi; \forall\ \beta_{m,k} \in \mathbb{Z} \tag{6}$$

Here, $\beta_{m,k}$ represents the number of rotations or turns completed in the circle and is found recursively such that

$$\beta_{m,k} = \begin{cases} \beta_{m-1,k} & \text{for } -\pi < \bar{\theta}_{m,k} - \bar{\theta}_{m-1,k} \leq \pi \\ 1 + \beta_{m-1,k} & \text{for } \bar{\theta}_{m,k} - \bar{\theta}_{m-1,k} \leq -\pi \text{ and } \beta_{1,k} = 0 \\ \beta_{m-1,k} - 1 & \text{for } \pi < \bar{\theta}_{m,k} - \bar{\theta}_{m-1,k} \end{cases} \tag{7}$$

For simplicity, we write the phase unwrapping as

$$\theta_{m,k} = \Xi\left(\bar{\theta}_{m,k}\right) \tag{8}$$

where the symbol $\Xi(\cdot)$ is the unwrapped function operator.

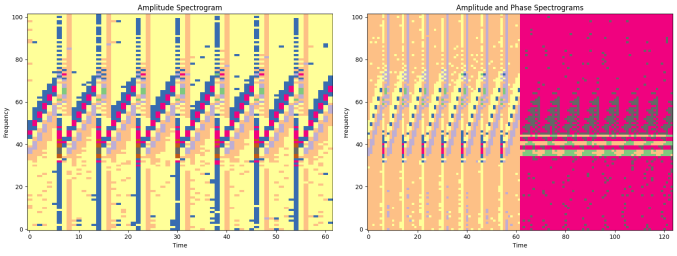
The collective impact of hardware distortion on the transmitted signal in the frequency domain can be represented by the STFT complex matrix as follows

$$\mathbf{Y} = \bar{\mathbf{H}} \odot F(\bar{\mathbf{X}}) \tag{9}$$

where:

$$\bar{\mathbf{H}} = \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,K} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,K} \\ \vdots & \cdots & \ddots & \vdots \\ H_{N,1} & H_{N,2} & \cdots & H_{N,K} \end{bmatrix} \tag{10a}$$

$$F(\bar{\mathbf{X}}) = \begin{bmatrix} F(X_{1,1}) & F(H_{1,2}) & \cdots & F(H_{1,K}) \\ F(H_{2,1}) & F(H_{2,2}) & \cdots & H_{2,K} \\ \vdots & \cdots & \ddots & \vdots \\ F(H_{N,1}) & F(H_{N,2}) & \cdots & F(H_{N,K}) \end{bmatrix} \tag{10b}$$



(a) (b)

Fig. 1: (a). Amplitude Channel independent spectrogram of LoRa preambles; (b). Combined Amplitude-Phase Channel independent spectrogram of LoRa preambles

Here, $\bar{\mathbf{X}}$ is the transmitted baseband signal and $\bar{\mathbf{H}}$ denotes the channel frequency response experienced by it. The $\odot$ and $F(\cdot)$ represent the element-wise product and the overall hardware impairments associated with the transmitted signal in the frequency domain, respectively.

We use the amplitude of $\mathbf{Y}$ to reveal the amplitude information of the signal. For ease of analysis, it is often assumed that the wireless channel remains unchanged within a short period. Therefore, the k-th column of $\mathbf{Y}$ is divided element-wise by its $(k-1)$-th to mitigate the channel-related information. However, exploiting amplitude and phase information ensures that every essential signal part is included. Accordingly, the amplitude information of the channel-independent spectrogram is given by

$$\mathbf{A}_{\text{spect}} = 10\log_{10}\left(\left|\mathbf{Y}_K \oslash \mathbf{Y}_{K-1}\right|^2\right) \tag{11}$$

where $\oslash$ is element-wise division.

We use the resulting STFT complex matrix $\mathbf{Y}$, (3), (6), (7) and (8) to obtain the unwrapped phase spectrogram matrix ($\mathbf{\Theta}_{\text{spect}}$).

$$\mathbf{\Theta}_{\text{spect}} = \Xi\left(\bar{\mathbf{\Theta}}\right) \tag{12}$$

where $\bar{\mathbf{\Theta}} = \text{atan}\left(\Im(\mathbf{Y}), \Re(\mathbf{Y})\right)$. By combining the channel-independent spectrograms of amplitude and phase information, we reconstruct the signal without the channel-related information distortions as follows

$$\mathbf{Y}_{\mathbf{A},\mathbf{\Theta}} = \mathbf{A}_{\text{spect}} \cdot e^{j(\mathbf{\Theta}_{\text{spect}})} \tag{13}$$

Finally, the combined effects of the amplitude and phase information channel-independent spectrograms is given by

$$\mathbf{\Gamma}_{\mathbf{A},\mathbf{\Theta}} = \text{concat}\{\mathbf{A}_{\text{spect}}, \mathbf{\Theta}_{\text{spect}}\} \tag{14}$$

Fig. 1(a) shows the channel-independent spectrogram of the LoRa preambles obtained from signal amplitude. The amplitude spectrogram can be observed to contain. Fig. 1(b) displays the channel-independent spectrograms, capturing both joint amplitude and phase information. The spectrogram is divided into two parts: the left-hand side represents the amplitude, while the right-hand side represents the phase. The spectrogram is partitioned into two. The left-hand side represents the amplitude, and to the right is the phase. It can be observed that the phase spectrogram, using the phase of the signal from its STFT, provides additional information which complements that offered by the amplitude (power) spectrogram.

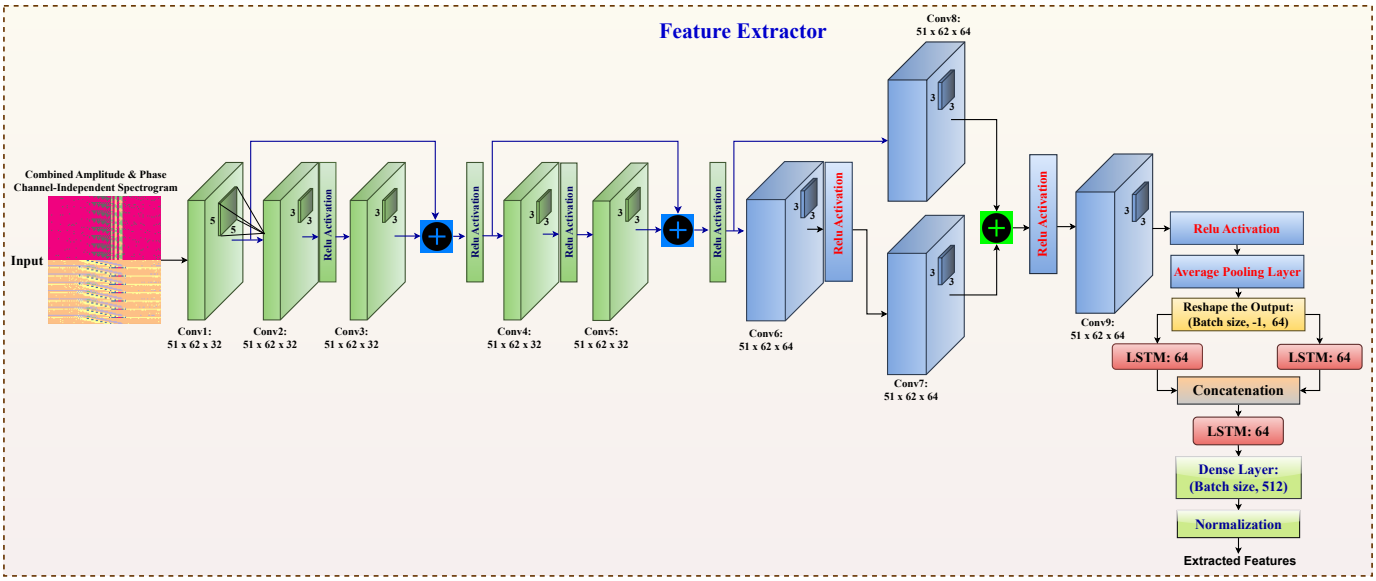


Fig. 2: Feature Extractor Architecture

C. RFFI Model Architecture

The proposed feature extractor is designed with CNN and LSTM layers, as shown in Fig. 2. We use CNN layers with skip connections to address varnishing gradients and promote feature reuse to improve performance and efficiency. The LSTM offers parameter sharing across different time steps, which reduces the number of parameters, making the network computationally more efficient. It also provides flexibility with respect to input and output data sizes. The RFF extractor as shown in Fig. 2 has 9 convolutional layers with strides of 2 and 1 average pooling layer applied to the last convolutional layer. The first layer has 5×5 filters with 32 channels. The second to fifth layers use 3×3 filters and 32 channels. The sixth to ninth layers utilize 3×3 filters with 64 channels. A rectified linear unit (ReLu) activation and the same padding are used for all the convolutional layers. The output of the last convolutional layer is connected to two LSTM layers, each having 64 sequence lengths. Their outputs are merged and connected to the third LSTM layer with 64 sequence lengths. Finally, the last LSTM layer is connected to a dense layer of size 512, which is L2 normalized to learn better features.

D. Dataset, and Feature Extractor Training

The dataset contains 30000 samples generated from 60 commercial off-the-shelf LoRa devices under test (DUTs) as RF emitters and USRP N210 software defined radio (SDR) platform as the receiver [7]. The model of DUTs used were labeled as flows: Pycom LoPy4 $(1-45)$. As explained in [7], 500 packets were collected from each of DUTs $1-30$ in a residential room with a line of sight (LoS) between the DUT and the receiver. Data augmentation was performed on the dataset to make the data robust to wireless channel variations. A fraction of 10% of the training dataset is reserved for validation. The RFF extractor is trained using the Adam optimizer with an initial learning rate of 0.001. The learning rate is adjusted to decrease by a factor of 0.2 whenever the validation loss does not improve for ten consecutive epochs.

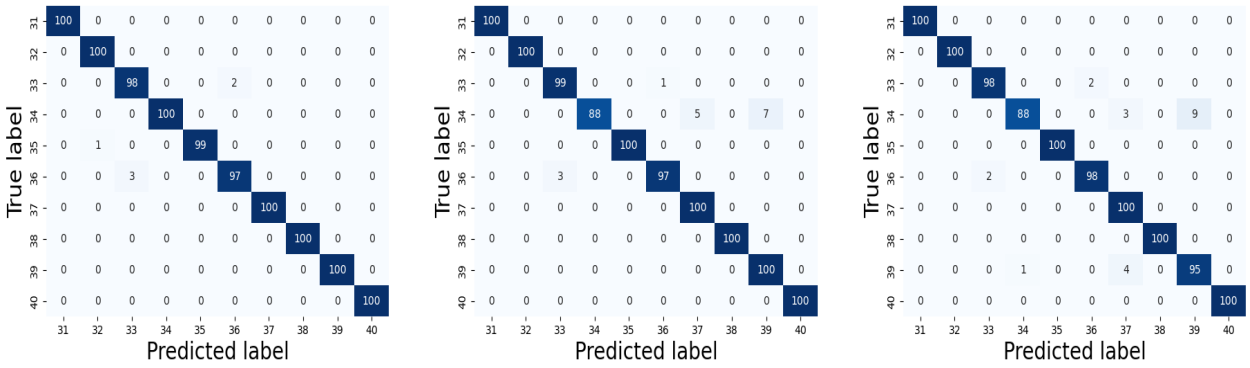
The DUTs numbered $31-40$ are considered legitimate, while DUTs numbered $41-45$ are regarded as rogue devices (specifically, LoPy4 devices). To assess the extractor's effectiveness on unseen devices, we use additional DUTs, numbered $31-40$, which belong to the same manufacturer (LoPy4) as the ones used for training. A triplet loss function is used as a training metric to train the RF extractor because it encourages the extractor to learn discriminative embeddings that capture the underlying semantic information of the data, enabling better template extraction [22]. A triplet loss function is given by

$$\mathcal{L}_{triplet} = \frac{1}{N}\sum_{i=1}^{N}\Big(\|g(x_i^a)-g(x_i^p)\|^2-\|g(x_i^a)-g(x_i^n)\|^2+\varepsilon\Big) \tag{15}$$

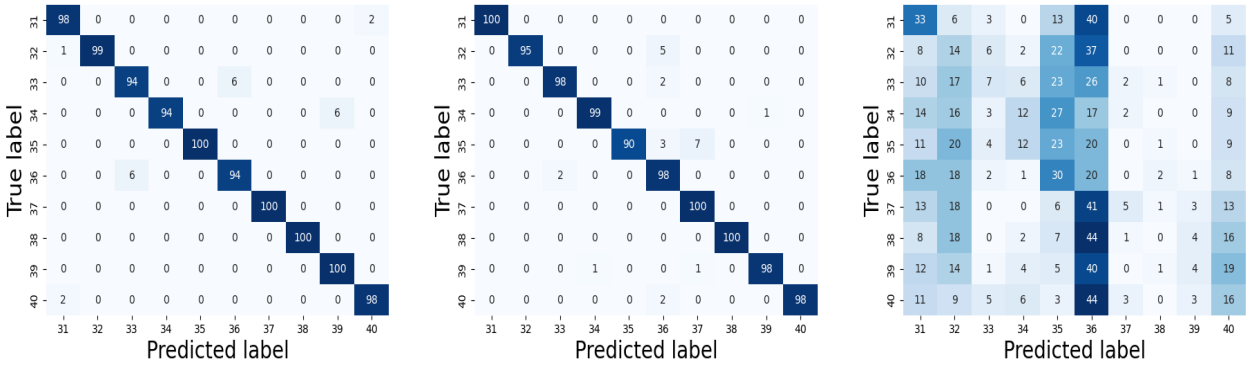
where $g(x)$ accepts x of the i-th input samples, the superscripts a, p and n denote the anchor, positive and negative samples, and ε is a predefined parameter that defines the margin between positive and negative samples. The goal of (15) is to minimize the first term (distance between similar data points) and maximize the second term (distance between dissimilar data points), and ε acts as a threshold.

E. Device Enrollment and Identification

The extractor is mainly used to extract and store the RFF templates in a database. We train various machine learning classifiers, such as k-nearest neighbor (k-NN), random forest, gradient boosting (XGBoot), decision tree, naive bayes, and support vector machine (SVM) to enroll the devices in the database, where these classifiers will essentially memorize all the training samples.



(a) k-NN (b) Random Forest (c) XGBoot



(d) Decision Tree (e) SVM (f) Naive Bayes

Fig. 3: Classification results on LoRa devices

From the perspective of machine learning, device identification is treated as a classification problem, aiming to deduce the specific label of a transmitter that sent the received packets. The system works by outputting a label previously enrolled in the RFF database using any machine learning algorithms mentioned in section A. The RFF of the received packet is first extracted. Then a machine learning algorithm is trained on the extracted templates to deduce the labels that closely match the ones stored in the database based on the prediction probability.

IV. Experimental Evaluation and Results

The parameters used for the experimental simulation are summarized in Table I.

TABLE I: Simulation settings

Parameters	Values
Training Samples (500 packets/device)	30000
Validation Samples	10% of the training samples
Batch Size	32
Enrolment Samples (100 packets/device)	6000
Authentication Samples (100 packets/device)	6000
SINR range	10dB - 80dB
Bandwidth	125 KHz
L (STFT window size),	128
N, K based on the formulation in [7]	256, 63
Signal overlap	128 μs
Triplet loss function margin	0.2
Number of neighbors in k-NN	15

Before device classification is performed, legitimate devices must send several packets for enrolment (100 packets). This involves training a classifier on the extracted RF templates from the RF extractor. In this case, the classifier memorizes all the training samples. The device classification is formed by evaluating the trained classifier on the packets from other devices (unseen devices) to predict the infer of their identity.

Fig. 3 shows the different classifiers' confusion matrixes. It can be seen that k-NN and Random Forest perform pretty well in identifying the devices by correctly classifying the packets from them. They perform better than other classifiers in this task. For instance, k-NN (see Fig. 3(a)) correctly classifies all the packets from the DUTs except those from DUT-33, DUT-35, and DUT-36. Similarly, it can be observed in Fig. 3(b) that random forest misclassifies DUTs 33, 34, and 36. Fig. 3(c) shows that XGBoost also performs well with a 0.05% performance gap below k-NN and Random Forest. While the Naive Bayes (Fig. 3(f)) has failed, the Decision Tree and SVM (Fig. 3(d) and Fig. 3(e)) have comparatively good performance, but the former surpasses the latter.

Fig. 4 illustrates the classification accuracies of different classifiers trained using extracted RF templates based on amplitude and combined amplitude-phase channel-independent spectrograms. It is evident that the classifiers

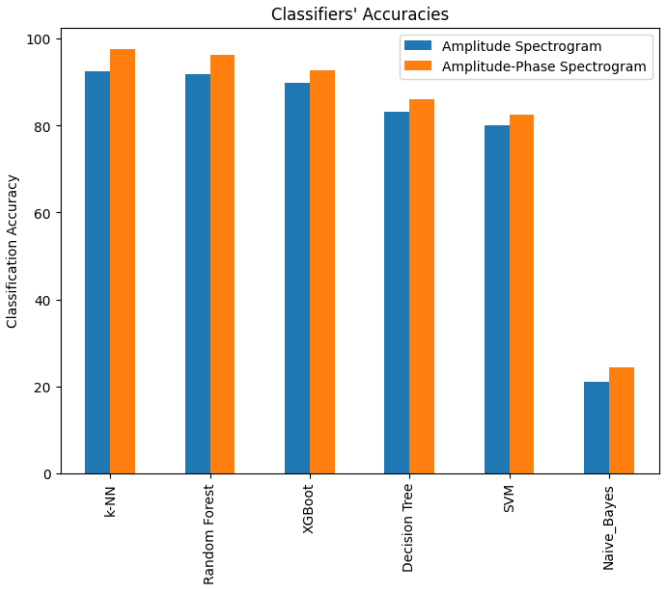


Fig. 4: Classification accuracies based on amplitude-phase and amplitude channel-independent spectrograms

demonstrate superior performance when trained with the extracted RF features obtained from both amplitude-phase spectrograms, owing to the additional information contained in the phase component of the RF signal.

V. Conclusion

This paper proposes a flexible deep-learning framework for RF that uses the unique features in the RF waveform due to the hardware impairments and imperfections for device identification. In addition to CNN layers for effective RF feature extraction, we leverage the SLTM layer to make the learning-based RFF model flexible to inputs of varying lengths. We design a channel-independent spectrogram that combines amplitude and phase information to overcome channel effects and enhance system robustness to distortions. We observe that machine learning classifiers perform better when trained on RF features extracted from the combined amplitude and phase spectrograms.

References

[1] A. Jagannath, J. Jagannath, and P. S. P. V. Kumar, "A comprehensive survey on radio frequency (RF) fingerprinting: Traditional approaches, deep learning, and open challenges," *Computer Networks*, vol. 219, p. 109455, 2022.

[2] 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.

[3] W. Zhang, W. Zhao, X. Tan, L. Shao, and C. Ran, "Adaptive rf fingerprints fusion via dual attention convolutions," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 181–25 195, 2022.

[4] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio frequency fingerprint identification for narrowband systems, modelling and classification," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3974–3987, 2021.

[5] 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 *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 646–655.

[6] S. Rajendran and Z. Sun, "Rf impairment model-based iot physical-layer identification for enhanced domain generalization," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1285–1299, 2022.

[7] G. Shen, J. Zhang, A. Marshall, and J. R. Cavallaro, "Towards scalable and channel-robust radio frequency fingerprint identification for lora," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 774–787, 2022.

[8] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[9] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of radio frequency fingerprinting techniques," *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222–233, 2020.

[10] J.-M. Park, J. H. Reed, A. Beex, T. C. Clancy, V. Kumar, and B. Bahrak, "Security and enforcement in spectrum sharing," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 270–281, 2014.

[11] S. Rajendran, Z. Sun, F. Lin, and K. Ren, "Injecting reliable radio frequency fingerprints using metasurface for the internet of things," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1896–1911, 2020.

[12] G. Reus-Muns, D. Jaisinghani, K. Sankhe, and K. R. Chowdhury, "Trust in 5g open rans through machine learning: Rf fingerprinting on the powder pawr platform," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[13] Z. Xu, A. Qu, and K. An, "Coalitional game based joint beamforming and power control for physical layer security enhancement in cognitive iot networks," *China Communications*, vol. 18, no. 12, pp. 139–150, 2021.

[14] Y. Xing, A. Hu, J. Zhang, J. Yu, G. Li, and T. Wang, "Design of a robust radio-frequency fingerprint identification scheme for multimode LFM radar," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 581–10 593, 2020.

[15] H. Yuan, Y. Yan, G. Zhang, and Z. Bao, "Payload symbol-based nonlinear rf fingerprint for wireless qpsk-ofdm devices," *China Communications*, vol. 20, no. 6, pp. 240–248, 2023.

[16] W. Wang, Z. Sun, S. Piao, B. Zhu, and K. Ren, "Wireless physical-layer identification: Modeling and validation," *IEEE transactions on information forensics and security*, vol. 11, no. 9, pp. 2091–2106, 2016.

[17] S. Balakrishnan, S. Gupta, A. Bhuyan, P. Wang, D. Koutsonikolas, and Z. Sun, "Physical layer identification based on spatial–temporal beam features for millimeter-wave wireless networks," *IEEE transactions on information forensics and security*, vol. 15, pp. 1831–1845, 2019.

[18] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for lora using deep learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2604–2616, 2021.

[19] H. Al-Nashi, "Phase unwrapping of digital signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1693–1702, 1989.

[20] F. Léonard, "Phase spectrogram and frequency spectrogram as new diagnostic tools," *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 125–137, 2007.

[21] R. M. Parry and I. Essa, "Incorporating phase information for source separation via spectrogram factorization," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 2. IEEE, 2007, pp. II–661.

[22] W. Ge, "Deep metric learning with hierarchical triplet loss," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 269–285.