An Investigation of Power Amplifier Feature for Deep Learning Based RF Fingerprint Identification

Wentao Jing*, Linning Peng*†, Junqing Zhang‡, and Hua Fu*†
* School of Cyber Science and Engineering, Southeast University, Nanjing, China
† Purple Mountain Laboratories for Network and Communication Security, Nanjing, China
‡ Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, United Kingdom
Email:{wentaojing, pengln, hfu}@seu.edu.cn, junqing.zhang@liverpool.ac.uk

***Abstract*—Radio frequency fingerprint identification (RFFI) employs unique hardware impairments for device authentication. Power amplifier (PA) nonlinearity is an important hardware impairment feature, which has been widely used for RFFI. In this paper, we modeled and estimated PA parameters and studied their effects on RFFI. In particular, a memory polynomial model was used to capture the detailed features of PA, involving polynomial order, memory depth and coefficient matrix. These parameters are twisted and we designed grid search and deep learning-based algorithms to find optimal parameters, with a tradeoff of computational complexity and RFFI classification accuracy. We created a testbed consisting of 60 IEEE 802.15.4 devices and a universal software radio peripheral (USRP) X310 software-defined radio (SDR) platform as the receiver. The PA function of our devices can be disabled when needed, which allowed us to generate signals with and without PA features. Experimental results demonstrated that our proposed algorithms can estimate the PA parameters. Simulated signals with the estimated PA parameters were also demonstrated to retain the PA features accurately.**

***Index Terms*—Radio frequency fingerprint identification, deep learning, power amplifier.**

I. Introduction

Device authentication is the first step to ensure wireless security, which authenticates device identities when they want to join the network. Traditionally, it is handled by cryptographic approaches [1]. However, these approaches tend to be computationally expensive and may not be suitable for low-cost internet of things (IoT) devices that usually have limited computational resources [2]. In addition, device addresses such as media access control (MAC) addresses are usually used for device identification. However, these addresses can be easily altered. Therefore, there is an urgent need for lightweight device authentication for IoT.

Radio frequency fingerprint identification (RFFI) emerged as a promising candidate [3]. RFFI relies on hardware impairments that result from imperfections during the manufacturing process. These impairments are unique and stable, which can be used as device identifiers, in a similar way as biometric fingerprints. RFFI protocols can be implemented at the receiver end and do not require any modification to the IoT transmitter. Hence, it can be readily applied to any IoT systems including legacy ones. Therefore, RFFI has been widely studied with WiFi [4], [5], ZigBee [6], long-term evolution (LTE) [7], and LoRa [8]–[10], etc.

Lining Peng is the corresponding author. This work was supported in part by the National Natural Science Foundation of China under Grant 62171120.

Deep learning has been widely used in RFFI due to its excellent ability to automatically extract features [3], [11], [12]. Deep learning significantly pushes the boundary of RFFI since it was first adopted in the community [8]. However, deep learning-based RFFI treats the hardware impairments as black box and does not distinguish their individual contributions. Studying their contribution to RFFI is significant, based on which we can categorize these features into irrelevant features, weak features, and strong features.

There are various features of signals from wireless devices such as carrier frequency offset (CFO), I/Q gain imbalance, and power amplifier (PA) feature [13]. Among these impairments, PA nonlinearity has been widely used for RFFI. Fu *et al.* [14] proposed an RFFI method to eliminate the channel effect with nonlinear features with memory effect, which gets 92.92% average identification accuracy in 22 IEEE 802.11 devices. Besides, Li *et al.* [15] utilized eight PAs to study RFF and achieved 92% of average classification accuracy when signal to noise ratio (SNR) was 10 dB. Furthermore, Li *et al.* [16] used a physical unclonable function (PUF) to dynamically tune the PA feature, which enlarged the RFF feature differences among the wireless devices. Therefore, according to previous studies, the PA feature is an essential feature in RFFI, which is valuable in further research.

There are many PA models including both memoryless models and models with memory effects, such as Saleh model [17], Wiener-Hammerstein [18], memory polynomial model [19], and generalized memory polynomial model [20]. There are existing works to model PA behavior with the memoryless nonlinear model for RFFI [13]. While memoryless models are simple, they may not capture all the PA features. Instead, the memory effect appears on real PAs. Therefore, it is meaningful to build a PA model with the memory effect and estimate related parameters.

In this paper, we performed a comprehensive modelling and estimation of PA parameters for RFFI and carried out experimental evaluation. The memory polynomial model was used, consisting of the polynomial order, memory depth, and coefficient matrix. We designed algorithms to estimate their values and used these values to simulate/generate new signals. We created a testbed using 60 IEEE 802.15.4 and a universal

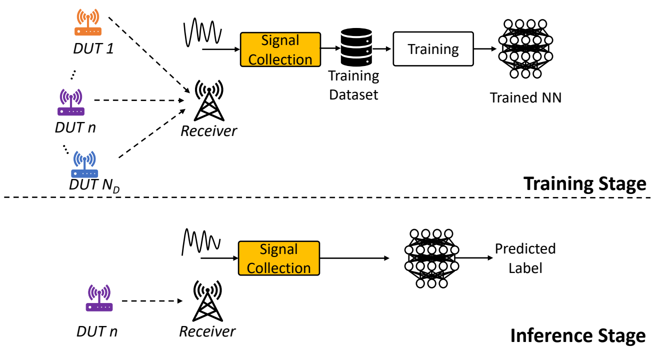


Fig. 1. Deep learning-based RFFI.

software radio peripheral (USRP) X310 software-defined radio (SDR) platform. Our experimental results demonstrated that our algorithms can replicate the PA features accurately. Our contributions are listed as follows:

- We adopted the memory polynomial model with polynomial order, memory depth, and coefficient matrix. We used a grid search to estimate the polynomial order. We then employed a deep learning-based RFFI model to obtain the optimal memory depth. The coefficient can then be calculated accordingly.
- Signals with and without PA features were collected, which were used to estimate the PA parameters and generate/simulate new signals with PA features. This was feasible as the PA of the selected IEEE 802.15.4 devices can be disabled.
- We carried out a comprehensive experimental evaluation. When using the simulated signals as training dataset and real collected signals as test dataset, the deep learning model can achieve 99.92% classification accuracy, which demonstrated that the simulated signals retain the PA features accurately.

The remainder of this paper is organized as follows. Section II briefly introduces RFFI and PA model. In Section III, the estimation algorithms are elaborated. The experimental setup and results are given in Section IV and Section V, respectively. Section VI concludes the paper.

II. RFFI Primer and Power Amplifier Model

A. RFFI

RFFI utilizes the hardware impairments from the manufacturing imperfections of devices for identification. As shown in Fig. 1, there are N_D devices under test (DUTs) and a receiver. A deep learning-based RFFI approach involves training and inference stages. In the training stage, a receiver constructs a training dataset with signals from all DUTs. A deep learning model is then trained. In the inference stage, the trained deep learning model will make a prediction of the device identity based on the received signal.

B. PA Model

PA has abundant nonlinear features which are suitable for RFFI. There are many PA models such as Saleh model [17], Wiener-Hammerstein [18], memory polynomial model [19], and generalized memory polynomial model [20]. With the consideration of the memory effect and complexity of various PA models, we utilize the memory polynomial model as the PA model with lower parameters of complexity and memory effect, which can be described by

$$y[n] = \sum_{k=1}^{K} \sum_{m=0}^{M} a_{k,m} x[n-m] \left|x[n-m]\right|^{k-1}, \tag{1}$$

where $y[n]$ is the output of the signal passed through the PA, $x[n]$ is the input signal to the PA with a length of N. K and M represent the polynomial order and the memory depth, respectively. $a_{k,m}$ is the Volterra kernel of the PA, and these form a Volterra kernel vector $\mathbf{a} \in \mathbb{C}^{K(M+1)\times 1}$, which can be described by

$$\mathbf{a} = [\mathbf{a}_1, \cdots, \mathbf{a}_K]^T \tag{2}$$

with

$$\mathbf{a}_k = [a_{k,0}, \cdots, a_{k,M}]^T. \tag{3}$$

III. Estimation of PA Parameters

A. Overview

As can be shown in (1), there are three parameters need to be determined, $\{K, M, \mathbf{a}\}$, which will be estimated in this section. The estimated parameters are given as $\{\hat{K}, \hat{M}, \tilde{\mathbf{a}}\}$.

The values of $\hat{K}$, $\hat{M}$, and $\tilde{\mathbf{a}}$ are twisted with each other. In the following two sections, we will explain how to estimate $\hat{K}$ and $\hat{M}$. We will first estimate $\hat{K}$ by iterating various M values. Once $\hat{K}$ is estimated, it is fixed. We can then focus on estimating $\hat{M}$.

Given K and M, the corresponding $\tilde{\mathbf{a}}$ can be calculated by the least squares method as

$$\tilde{\mathbf{a}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\tilde{\mathbf{y}}, \tag{4}$$

where $\tilde{\mathbf{y}} \in \mathbb{C}^{(N-M)\times 1}$ is the vector of the collected output signals with PA features. $\mathbf{X} \in \mathbb{C}^{(N-M)\times K(M+1)}$ is the input signal matrix, which can be described by

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_K \end{bmatrix}, \tag{5}$$

where $\mathbf{X}_k \in \mathbb{C}^{(N-M)\times(M+1)}$, and $\mathbf{X}_k$ can be described by

$$\mathbf{X}_k = \begin{bmatrix} \chi_{M+1}^k & \cdots & \chi_1^k \\ \vdots & \ddots & \vdots \\ \chi_N^k & \cdots & \chi_{N-M}^k \end{bmatrix}, N > M+1 \tag{6}$$

with

$$\chi_n^k = x[n] \left|x[n]\right|^{k-1}. \tag{7}$$

B. Estimation of Polynomial Order K

We adopted a method to grid search the proper K and M [21]. Firstly, it defines K_i and M_j, where i and j represent the index of the different K and M, respectively, in the grid search process. We can calculate $\tilde{\mathbf{a}}^{ij} \in \mathbb{C}^{K_i(M_j+1)\times 1}$ by (4), and $\mathbf{X}^{ij} \in \mathbb{C}^{(N-M_j)\times K_i(M_j+1)}$ by (5). Then, we generate simulated frame signal $\bar{\mathbf{y}}^{ij}$, which can be described by

$$\bar{\mathbf{y}}^{ij} = \mathbf{X}^{ij}\tilde{\mathbf{a}}^{ij}, \tag{8}$$

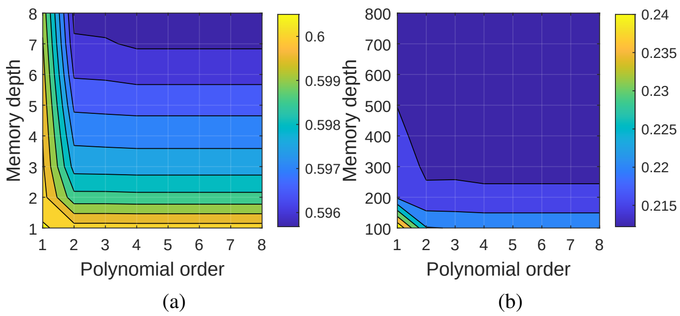


Fig. 2. Values of RMS error distribution against K and M. (a) Device a with memory depth from 1 to 8. (b) Device b with memory depth from 100 to 800 (with a step of 100).

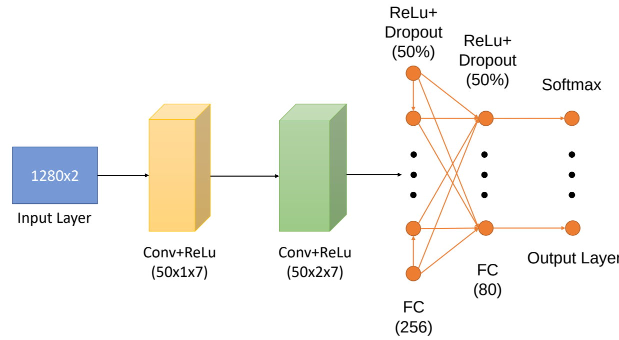


Fig. 3. CNN architecture.

where $\bar{\mathbf{y}}^{ij} \in \mathbb{C}^{(N-M_j)\times 1}$, which is the simulation for $\tilde{\mathbf{y}}$ with K_i and M_j.

According to the $\bar{\mathbf{y}}^{ij}$, we can calculate the difference vector $\mathbf{e}^{ij}$. The n^{th} element of $\mathbf{e}^{ij}$, denoted as $e^{ij}[n]$, can be described by

$$e^{ij}[n] = \frac{\mid \tilde{y}[n] - \bar{y}^{ij}[n] \mid}{\mid \tilde{y}[n] \mid}, \tag{9}$$

where $\tilde{y}[n]$ and $\bar{y}^{ij}[n]$ are the n-th element of $\tilde{\mathbf{y}}$ and $\bar{\mathbf{y}}^{ij}$ respectively. $e^{ij}[n]$ represents differences between $\tilde{y}[n]$ and $\bar{y}^{ij}[n]$ in the time domain.

Then, we computes error of Root-Mean-Square (RMS) for $\mathbf{e}^{ij}$, which can be described by

$$r^{ij} = \sqrt{\frac{1}{N-M_j} \sum_{n=1}^{N-M_j} \left(e^{ij}[n]\right)^2}, \tag{10}$$

With this method, we can get r with respect to K_i and M_j. Fig. 2 exemplifies values of RMS errors.

Then, we utilize multiple frames from multiple devices to generate the results that the value of RMS error distribution against K and M. By these results, we utilize dimensionality reduction algorithm to generate a result that RMS related to polynomial order, including (11), (12), and (13).

$$\mathbf{R}_q = \begin{bmatrix} r_q^{11} & \cdots & r_q^{I1} \\ \vdots & \ddots & \vdots \\ r_q^{1J} & \cdots & r_q^{IJ} \end{bmatrix}, \tag{11}$$

where $\mathbf{R}_q$ is the result of RMS distribution against K and M. q is the index of the RMS distribution result generated from the q-th collected signal frame. $M_j \in [M_1, M_J]$, $K_i \in [K_1, K_I]$, and M_1 and M_J are 1 and 8 respectively in Fig. 2a, M_1 and M_J are 100 and 800 respectively in Fig. 2b, and K_1 and K_I are 1 and 8 in Fig. 2.

We then sum each column, given as

$$r_q^i = \sum_{j=1}^{J} r_q^{ij} \tag{12}$$

where $\mathbf{r}_q^i$ is the sum of i-th column in $\mathbf{R}_q$. In this way, we can focus on the effect of K for now.

Finally, we utilize a vector $\bar{\mathbf{r}}$ to present the mean of the each $\mathbf{r}_q^i$ with Q frames, which can be described by

$$\bar{\mathbf{r}} = \left[\frac{1}{Q} \sum_{q=1}^{Q} r_q^1, \ldots, \frac{1}{Q} \sum_{q=1}^{Q} r_q^I \right]. \tag{13}$$

According to the result that the trend of $\bar{\mathbf{r}}$ related to polynomial order, we can get the measured optimal K, namely $\hat{K}$.

C. Estimation of Memory Depth M

We used a different method to estimate M. Specifically, we utilize (8) with predetermined polynomial order $\hat{K}$ and varying memory depth M_l to generate simulated frame $\bar{\mathbf{y}}$, where l is the index of the varying memory depth. Considering there are N_D devices, we generate N_f frames for each device. These frames are then used to train a CNN model for RFFI. We utilize the accuracy of the CNN-based identification to evaluate the performance of simulation with varying parameters for exploring proper parameters. Because the purpose of this paper is to generate simulated frames using the estimated PA parameters for RFFI, using this approach can ensure that the generated frames can replicate the intrinsic PA features.

A CNN architecture proposed by [22] is used in this paper, which is illustrated in Fig. 3. The input is I/Q samples with a size of 1280×2. The CNN model consists of two 2-D convolutional layers followed by a batch normalization (BN) layer and a maxpooling layer, and the last one of them is connected to two fully connected layers. ReLu function is adopted to activate convolutional layers and fully connected layers, while the softmax function is utilized at the end of the structure. Each fully connected layers followed by a 50% dropout layer to prevent overfitting.

Given $\hat{K}$ and M_l, the CNN accuracy is expressed as

$$\Theta = \left\{ \theta(\hat{K}, M_l) \mid l \in [1, L] \right\}, \tag{14}$$

We can then search the expected identification accuracy in the set Θ, namely the expected performance of simulation, and find the corresponding M_l, which is estimated optimal M, namely $\hat{M}$. The expected performance of the simulation is generated from the identification accuracy with real frames, which is approaching accuracy in reality.

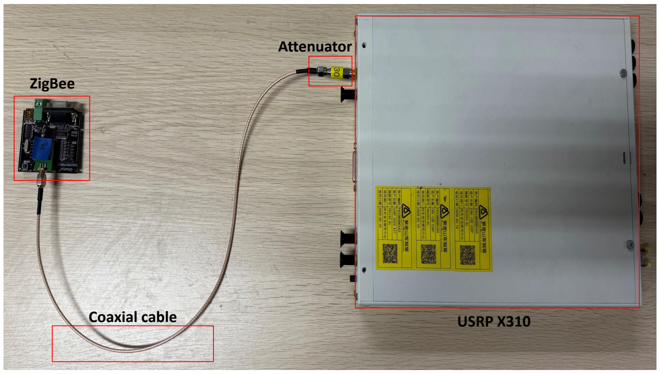


Fig. 4. Signal collection scenario.

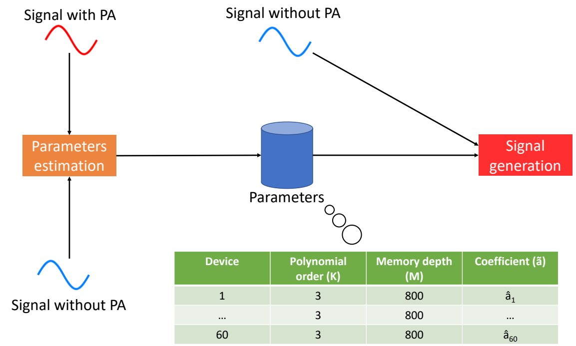


Fig. 5. Estimation of PA parameters and signal generation.

D. *Estimation of Coefficients* $\tilde{\mathbf{a}}$

Once the optimal polynomial order $\hat{K}$ and optimal memory depth $\hat{M}$ are obtained, we can utilize (4) to calculate the corresponding Volterra kernel vector $\tilde{\mathbf{a}}$. We measure multiple signals for each device, so the optimal $\tilde{\mathbf{a}}$ is selected which corresponds to the minimum error of RMS with $\tilde{\mathbf{y}}$.

IV. Experimental Setup

This paper used IEEE 802.15.4 as a case study. Specifically, we employed $N_D = 60$ commercial-off-the-shelf (COTS) devices and Universal Software Radio Peripheral (USRP) software-defined radio (SDR) X310 as the receiver platform, as shown in Fig. 4.

- All devices comply with the IEEE 802.15.4 Physical layer (PHY) protocol. They send signal frames that of physical service data unit (PSDU) is full of 0, so all collected frames keep the same data.
- The USRP X310 platform is interfaced using GNU radio. Once the signals are collected, they are transferred to a PC for signal processing using Matlab, which includes synchronization, carrier phase offset compensation, carrier frequency offset compensation, and normalization. The sampling rate in the collection scenario is 10 MS/s.

Fig. 5 portrays the PA estimation and signal generation. As observed in (1), both the input and output signals of the PA are required. The devices that we selected allowed us to disable the PA manually, which can produce signals without PA features. When the PA is enabled, signals with PA features can be generated. In particular, we configured the device with the maximum transmission power, which is 15-level in the setting.

In order to focus on the PA features and avoid channel effects, we designed a cabled experiment for signal collection, shown in Fig. 4. We directly connected USRP X310 and the ZigBee devices using a radio frequency (RF) coaxial cable and an attenuator. When the PA is disabled, we set USRP X310 with a high receive gain. A detailed information is illustrated in Table I.

We collected 60 signal frames with PA and without PA, respectively, for each device. A segment of the signal frame with PA is shown in Fig. 6. There are small fluctuations in these signals, but the differences are not significant, which demonstrates that PA features are stable within a certain period of time.

TABLE I
PARAMETERS OF SIGNALS COLLECTION SCENARIO

PA Status	RF Attenuator Value	USRP Receive Gain
PA disabled	30 dB	75 dB
15 level (Maximum)	60 dB	65 dB

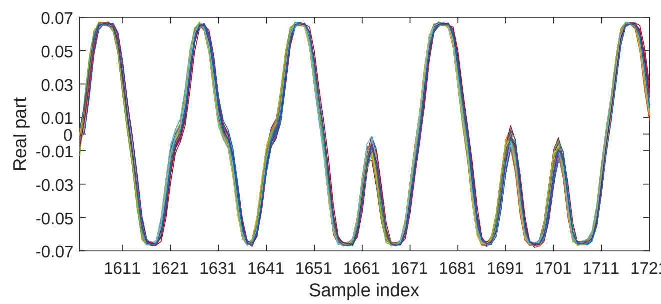


Fig. 6. 50 Signals from the same device.

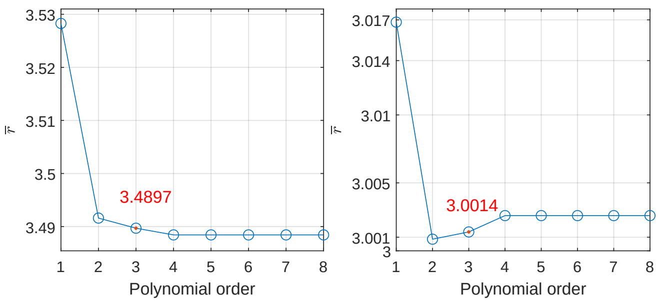
V. Experimental Results

We carried out multiple experiments to search for the optimal parameters, including polynomial order K, memory depth M, and Volterra kernel vector $\tilde{\mathbf{a}}$.

A. *Polynomial Order* K

We utilized 10 frames from each device to estimate K, totaling 600 frames. We used the entire part of the frame to take grid search by (8), (9), and (10), which can get 600 values of RMS error distribution against K and M. Therefore, we can use dimensionality reduction algorithm including (11), (12), and (13) with values of RMS error distributions.

In Fig. 7, we plot trends of $\overline{\mathbf{r}}$ versus memory depth. It can be observed that the values of $\overline{\mathbf{b}}$ do not decrease significantly when K is larger than 3. Meanwhile, considering the computational complexity caused by the increase of K, we chose $\hat{K}$ as 3 in the following of this paper.



(a) (b)

Fig. 7. Relationship between $\bar{\mathbf{r}}$ and polynomial order with (a) memory depth from 1 to 8 and (b) memory depth from 100 to 800.

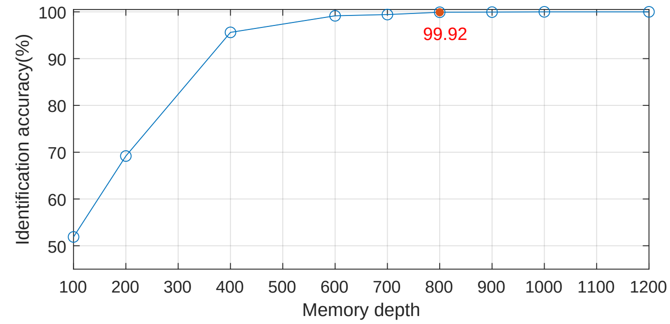
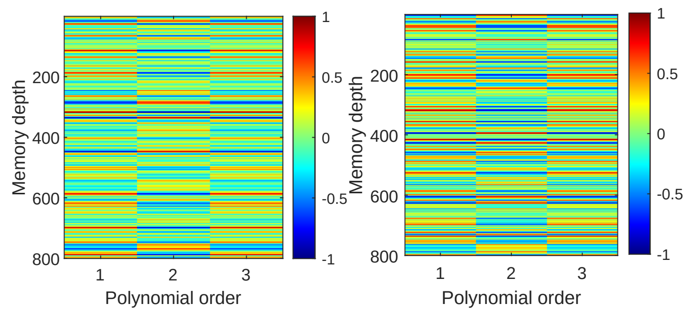


Fig. 8. Identification accuracy with 60 devices and varying memory depth.



(a) (b)

Fig. 9. Parameters distribution of $\hat{\mathbf{a}}$ between two devices. (a) Device a. (b) Device b.

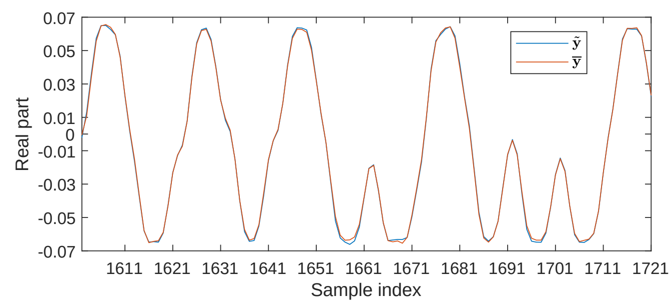


Fig. 10. Comparison between $\bar{y}[n]$ and $\tilde{y}[n]$.

B. Memory Depth M

We used 50 signal frames from each device to estimate M. We simulated frames with fixed $\hat{K} = 3$ and varying M from 100 to 1200 with a step of 100 by (8). By this way, we generated massive datasets of simulated frames for evaluation.

We selected 16 real frames for each of the 60 devices. Meanwhile, we simulated 16 real frames from the collected 50 frames for each device, and enlarged them to 800 frames as the training set, 192 frames as a validation set with adding noise of the 40 dB SNR for each device. We selected 30 frames from the rest of the real frames that do not overlap with the simulated frames in the training set and validation set. These 30 frames are enlarged to 120 frames per device by augmenting noise of the 40 dB SNR, which forms the test set. In this way, we can evaluate whether the simulated frames possess the PA features.

After the signals are generated/simulated, we used the signal segment from samples 1601 to 2881 as input to CNN. As the signal has both real and imaginary parts, the segmented IQ samples have a size of 1280×2.

We set the initial learning as 0.00001, the optimizer as Adam, and the maximum epochs as 250 to make the model converge better. In addition, the mini-batch size is 256, the validation frequency is 50, and each experiment takes 11 times training and tests for the median as a result. Besides, we use the early stop method and dropout layer to prevent overfitting.

The experimental results are demonstrated in Fig. 8. We selected $\hat{M}$ as 800, with a consideration of accuracy and computational complexity.

C. Coefficient Matrix $\tilde{\mathbf{a}}$

We utilized 50 collected real frames for each device. We used (4) to calculate $\hat{\mathbf{a}}$ with $\hat{K} = 3$ and $\hat{M} = 800$. Then, we utilize (8) to generate simulated frames and calculate RMS between real frames and simulated frames by (9) and (10). Searching the frames with the lowest RMS, we can find the optimal Volterra kernel vector $\hat{\mathbf{a}}$ for each device to represent the device.

Besides, we convert the Volterra kernel vector $\hat{\mathbf{a}}$ to a matrix with a size of $(\hat{M}+1) \times \hat{K}$, i.e., 801×3. The color map of $\hat{\mathbf{a}}$ of two devices are depicted in Fig. 9. For a better illustration effect, we only showed the real part and scale values in $\hat{\mathbf{a}}$ to the range [-1, 1]. The distribution of values close to 1 and -1 is different between the two devices, and most values are around 0.

D. Performance of Signal Generation

According to $\hat{\mathbf{a}}$, $\hat{K}$, $\hat{M}$ measured from 60 IEEE 802.15.4 device, we can use (8) to generate a simulated signal. As exemplified in Fig. 10, the generated waveform, $\bar{y}[n]$, matches with the real collected signal, $\tilde{y}[n]$, very well.

We constructed a few datasets for evaluation, as summarized in Table II. The subscripts α and β indicate the two datasets consisting of frames 1-16 and frames 17-46 out of the 60 collected signals, respectively. Besides, the superscripts, $real$ and sim, indicate real frames and simulated frames, respectively.

TABLE II
DATASET CONSTRUCTION

Datasets	Frame index range	Type
$\mathcal{D}_{\alpha}^{sim}$	1-16	Simulated frames
$\mathcal{D}_{\alpha}^{real}$	1-16	Real frames
$\mathcal{D}_{\beta}^{real}$	17-46	Real frames

TABLE III
EXPERIMENTAL RESULTS OF PERFORMANCE ANALYSIS

Index	Training and validation set	Test set	Median of accuracy
1	$\mathcal{D}_{\alpha}^{real}$	$\mathcal{D}_{\beta}^{real}$	100.00%
2	$\mathcal{D}_{\alpha}^{sim}$	$\mathcal{D}_{\alpha}^{real}$	100.00%
3	$\mathcal{D}_{\alpha}^{sim}$	$\mathcal{D}_{\beta}^{real}$	99.92%

We carried out three different evaluations and the results are given in Table III. For each evaluation, we performed 11 experiments and used the median value. Test 1 achieves 100% accuracy with CNN-based identification, which is valid as the SNR of the real collected signals is very high. Test 2 also achieves 100% accuracy, when the training and test datasets are simulated frames and real frames, respectively. They are based on the same frames 1-16. Finally, Test 3 is the most challenging scenario, as the training dataset is simulated but the frames between training and test datasets are not overlapping. The accuracy is still as high as 99.92%, which indicates that our approach can replicate the PA features accurately.

VI. CONCLUSION

In this paper, we modeled and estimated the PA parameters for the purpose of RFFI and carried out experimental evaluation using IEEE 802.15.4 devices. The memory polynomial model of PA was adopted, consisting of polynomial order, memory depth, and Volterra kernel vector. We used grid search to estimate the polynomial order. We then simulated the signals with the generated PA features and designed a CNN-based RFFI model to find the optimal memory depth. A testbed was created including 60 IEEE 802.15.4 devices and a USRP X310 SDR platform as the receiver. The PA of the device can be disabled which allowed us to generate signals both with and without PA features. Our experimental results demonstrated that our proposed algorithms can replicate the features of PA accurately. Specifically, when we used simulated signals as training dataset and real collected signals as test dataset, a classification accuracy as high as 99.92% can be obtained, under high SNR scenarios (40 dB).

REFERENCES

[1] P. Kushwaha, H. Sonkar, F. Altaf, and S. Maity, "A brief survey of challenge–response authentication mechanisms," *ICT Analysis and Applications: Proceedings of ICT4SD 2020, Volume 2*, pp. 573–581, 2021.

[2] N. Potlapally, S. Ravi, A. Raghunathan, and N. Jha, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," *IEEE Transactions on Mobile Computing*, vol. 5, no. 2, pp. 128–143, 2006.

[3] L. Xie, L. Peng, J. Zhang, and A. Hu, "Radio frequency fingerprint identification for internet of things: A survey," *Security and Safety*, vol. 3, p. 2023022, 2024.

[4] R. Kong and H. Chen, "DeepCRF: Deep learning-enhanced CSI-based RF fingerprinting for channel-resilient WiFi device identification," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 264–278, 2025.

[5] L. Xie, L. Peng, and J. Zhang, "Towards robust RF fingerprint identification using spectral regrowth and carrier frequency offset," in *Proc. IEEE INFOCOM*, 2025.

[6] J. Shi, L. Peng, H. Fu, and A. Hu, "Robust rf fingerprint extraction based on cyclic shift characteristic," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 19 218–19 233, Nov. 2023.

[7] L. Peng, H. Peng, H. Fu, and M. Liu, "Channel-robust radio frequency fingerprint identification for cellular uplink lte devices," *IEEE Internet Things J.*, 2024.

[8] P. Robyns, E. Marin, W. Lamotte, P. Quax, D. Singelée, and B. Preneel, "Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning," in *Proc. ACM Conf. Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2017, pp. 58–63.

[9] A. Al-Shawabka, P. Pietraski, S. B. Pattar, F. Restuccia, and T. Melodia, "DeepLoRa: Fingerprinting LoRa devices at scale through deep learning and data augmentation," in *Proc. Int. Symp. Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, Shanghai, China, Jul. 2021, pp. 251–260.

[10] G. Shen, J. Zhang, A. Marshall, and J. R. Cavallaro, "Towards scalable and channel-robust radio frequency fingerprint identification for LoRa," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 774–787, 2022.

[11] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "Deep learning for rf fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

[12] T. Zhao, X. Wang, and S. Mao, "Cross-domain, scalable, and interpretable rf device fingerprinting," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, 2024, pp. 2099–2108.

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

[14] H. Fu, Y. Sun, L. Peng, and M. Liu, "Channel-resilient RF fingerprint identification based on nonlinear features with memory effect," *IEEE Commun. Lett.*, vol. 28, no. 4, pp. 798–802, Apr. 2024.

[15] Y. Li, Y. Ding, J. Zhang, G. Goussetis, and S. K. Podilchak, "Radio frequency fingerprinting exploiting non-linear memory effect," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 4, pp. 1618–1631, Dec. 2022.

[16] Y. Li, K. Xu, J. Zhang, C. Gu, Y. Ding, G. Goussetis, and S. Podilchak, "PUF-assisted radio frequency fingerprinting exploiting power amplifier active load-pulling," *IEEE Trans. Inf. Forensics Secur.*, 2024.

[17] W.-L. Mao, W.-Z. Chen, C.-T. Yao, S.-H. Chen, and C.-C. Wang, "Digital predistortion technique using kernel adaptive algorithm for nonlinear power amplifier," in *2022 IET International Conference on Engineering Technologies and Applications (IET-ICETA)*, 2022, pp. 1–2.

[18] F. Abedi, "Digital pre-distorter system based on memoryless hammerstein model for high power amplifier impairments," *Arabian Journal for Science and Engineering*, vol. 49, no. 5, pp. 6419–6428, 2024.

[19] G. Mohiuddin, M. M. Shaikh, S. A. Dahri, F. Panhwar, K. A. Memon, and N. Madina, "Characterization of rf power amplifier for narrow and wide band memory polynomial implementations."

[20] D. Wang, Y. Lei, and L. Zeng, "A pruning method of the generalized memory polynomial model for power amplifiers based on the lasso regression," in *2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)*, 2023, pp. 182–185.

[21] Power amplifier characterization. 2024. [Online]. Available: https://ww2.mathworks.cn/help/comm/ug/power-amplifier-characterization.html

[22] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized radio classification through convolutional neural networks," in *Proc. IEEE INFOCOM*, Paris, France, 2019, pp. 370–378.