A Noise-Robust Radio Frequency Fingerprint Identification Scheme for Internet of Things Devices

Yuexiu Xing*,†, Xiaoxing Chen†, Junqing Zhang‡, Aiqun Hu§, Dengyin Zhang*,
*School of Internet of Things, Nanjing University of Posts and Telecommunications, 210003 Nanjing, China.
†Jiang Su Yi Tong High-tech Company Limited. 215500 Changshu, China.
‡Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom.
§School of Information Science and Engineering, Southeast University, 210096 Nanjing, China.

***Abstract*—Radio frequency fingerprint (RFF) identification is a potentially effective technique to address the authentication security of Internet of Things (IoT) devices. Since the complex working environment and limited resources of IoT devices, noise is non-negligible in RFF identification of IoT devices. It is a challenge to suppress the noise without damaging the RFF information. In this paper, we propose a robust RFF identification scheme, which consists of a frequency point selection (FPS) based denoising algorithm, and a convolutional neural network (CNN) classifier. The FPS algorithm performs denoising by filtering out all the frequency components that are independent of the RFF. The CNN is designed with a dynamically decreasing learning rate to accelerate learning and obtain optimal identification performance. Experiments were conducted with 54 ZigBee devices to evaluate the performance of the proposed scheme under three different RFF identification scenarios. The results show that the FPS algorithm brings the highest accuracy improvement of about 25% when the training signal-to-noise ratio (SNR) is hybrid and the test SNR is 0 dB.**

***Index Terms*—Radio frequency fingerprint, denoise, ZigBee, CNN**

I. Introduction

THE number of Internet of Things (IoT) devices is increasing exponentially in recent years, and their security is receiving increasing attention [1]. IoT devices are mainly connected via wireless. However, the broadcast nature of wireless communication makes it possible for any malicious user to access the network, which makes device authentication a challenge. Traditional device authentication is usually based on the cryptographic scheme, such as the preshared key and MAC/IP address. Cryptography involves key management, which is challenging for most IoT devices because they are low-cost, resource-limited, and remotely distributed. MAC/IP address is at risk of being easily spoofed [2].Therefore, a new lightweight and secure device authentication scheme is urgently needed.

Radio frequency fingerprint (RFF) is a promising physical layer security technique for device authentication. During manufacturing, each device has unique hardware imperfections due to the electronic component tolerances. As a result, the signals emitted by a device carry corresponding hardware characteristics. These characteristics are tiny and do not compromise normal communication operations; they are also hard to tamper with. Thus, they can be extracted to identify devices, named RFF. The RFF-based device identification operations are all performed at the receiver, which means the transmitter has no additional computational or power overhead. It is therefore particularly suitable for IoT devices.

Dengyin Zhang (zhangdy@njupt.edu.cn) is the corresponding author.

RFF-based device identification consists of two phases, i.e., training and identification [3]. In the training phase, the authenticator extracts RFF features from the signal of each legitimate device to form an RFF library, in which the RFF features and the device identity correspond to each other. In the identification phase, the authenticator receives signals from unknown devices and extracts RFF features using the same method as in the training phase. The device identity is then inferred by comparison with the RFF library. However, in practical communications, noise is inevitable and will interfere with RFF feature extraction and recognition, which has been experimentally demonstrated [4], [5].

The current research efforts on RFF noise suppression can be divided into two categories: signal processing-based and deep learning-based approaches. The signal processing-based approach first denoises the received signal while retaining the RFF information, and then performs RFF feature extraction and identification. Xie *et al.* proposed a noise reduction algorithm based on coherent integration, which mitigates the impact of noise on RFFs by coherent integration of multiple signals [6]. Xing *et al.* effectively improved the RFF identification performance of the direct sequence spread spectrum (DSSS) signal at a low signal-to-noise ratio (SNR) by signal superposition [7]. The two works are both based on the principle that RFF is signal-dependent and the noise is random, hence, the noise can be reduced by the coherent superposition of multiple identical symbols, which however limits its applications. Shen *et al.* [8] investigated the multi-packet-based RFF denoising method, which has a security risk since it needs multiple transmissions to make decisions. We need more general and secure approaches. Deep learning-based methods attempt to automatically learn the invariant part of the signal, i.e. RFF, from a large amount of noisy data. Wu *et al.* proposed a convolutional neural network (CNN) with dynamic shrinkage thresholding to improve device recognition accuracy at low SNR [9]. Yu *et al.* designed a deep learning-based Denoising Autoencoders to extract RFF features of signals [10]. However, as the noise increases, the tiny RFFs

will be swamped, which leads to the network cannot learn a stable RFF.

In order to address the above challenges, this paper proposed a robust RFF identification scheme. Specifically, we designed a signal processing-based RFF denoising algorithm, i.e. frequency point selection (FPS) based denoising algorithm, which performs noise suppression on the spectrum of the received signal. The main contributions are summarized as follows:

- We propose an RFF denoising method by searching and filtering noise frequency points in the signal spectrum named the FPS algorithm. Here, RFF is modelled as a filter that acts on the transmitted signal, which in theory does not introduce new frequency components to the transmitted signal. Therefore, the frequency component of a signal with RFF should be a subset of the frequency component of its standard signal. Thus, the FPS algorithm first generates a standard signal and extracts its all frequency components as RFF frequency points. The amplitude of all non-RFF frequency points in the received signal spectrum is then set to zero to obtain a denoised signal. Although there is noise residue at the RFF frequency points, the noise is effectively suppressed. Then, CNN with two convolutional layers is designed for RFF feature extraction and identification.
- We carried out extensive experimental evaluation with 12,238 packets collected from 54 ZigBee devices with SNRs of about 30 dB. Then, multiple datasets with SNRs ranging from 0 dB to 25 dB were created by manually adding additive white Gaussian noise (AWGN). The performance of the proposed RFF identification scheme is evaluated in three RFF identification scenarios. First, when the training SNR and the test SNR are equal, the highest RFF identification accuracy reaches up to 98.82% (SNR = 25 dB). In addition, the maximum performance improvement brought by the FPS algorithm is 21.45% (SNR = 0 dB). Second, when the SNR of the training set is fixed and the test set was variable, the FPS algorithm shows excellent noise robustness. Finally, when the training set has a hybrid SNR, FPS algorithm introduces an average accuracy improvement of 13% (from 0 dB to 10 dB).

The remainder of this paper is organized as follows. Section II introduces the structure of the IEEE 802.15.4 protocol. Section III gives the details of the robust RFF identification scheme. Section IV shows the experimental setup and performance evaluation. Finally, Section V concludes this paper.

II. Preliminary: IEEE 802.15.4 Protocol

In this paper, the ZigBee device (CC2530), which implements IEEE 802.15.4 protocol in the physical layer (PHY), is chosen as a case study to verify the proposed noise robust RFF identification scheme.

The specific IEEE 802.15.4 frame format is shown in Fig. 1, which contains three parts: the synchronization header (SHR), the PHY header, and the PHY service data unit (PSDU). The SHR consists of a preamble and a start frame delimiter (SFD).

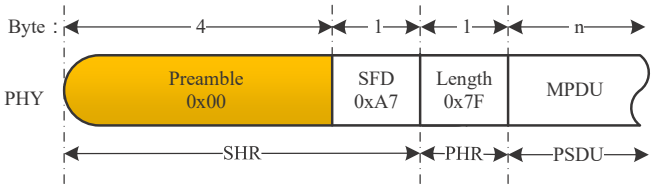

Fig. 1. Physical layer frame format of IEEE 802.14.5.

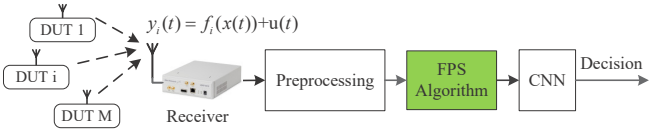

Fig. 2. System architecture of robust RFF identification.

The preamble consists of four 0x00s and the SFD is fixed at one byte 0xA7. The PHY header defines the number of bytes in the MAC protocol data unit (MPDU), which is set as 0x7F in our experiment. In this paper, we have chosen the preamble as the region of interest for device identification.

III. Noise-Robust RFF Identification Scheme

The authentication scenario studied in this paper is one in which a receiver identifies multiple transmitters. Therefore, the receiver's RFF is the same in all signals and can be neglected. As illustrated in Fig. 2, we propose a robust RFF identification scheme that consists of signal preprocessing, the FPS algorithm, and a CNN.

Consider that channel variation has less influence on the RFF identification performance of narrow-band signals [11], [12], and the channel is not the focus of our paper, the received baseband signal from the i-th device under test (DUT) can be modelled as

$$y_i(t) = f_i(x(t)) + u(t), i = 1, 2, \cdots, M, \quad (1)$$

where $x(t)$ denotes the transmitted signal, $f_i(x(t))$ is the signal after RFF distortion, in which the RFF of the transmitter is collectively represented as $f(\cdot)$, $u(t)$ is the noise, and M is the number of legitimate DUTs.

A. Preprocessing

Before RFF identification, signal preprocessing is necessary, including normalization and carrier frequency offset (CFO) compensation.

1) Normalization: The power of the received signal is related to the power of the transmitter and the transmission distance. Moreover, the signal power variations will impair the accuracy of RFF identification. It is therefore necessary to normalize the signal to unit power, which is achieved by dividing all signal samples by the root mean square (RMS) of the amplitude signal.

Algorithm 1 FPS Algorithm

Input:

$y_i'(n)$, the noisy signal;
$x(n)$, the standard signal;
γ, the threshold.

Output:

$\widehat{y}_i(n)$, The denoised signal.

1: Transform $x(n)$ to the frequency domain by the discrete Fourier transform (DFT) $X(k) = \mathcal{DFT}(x(n)), k = 1, 2, \cdots, N$.
2: Set $k = 1$;
3: **while** $k \leq N$ **do**
4: **if** $X(k) \geq \gamma$ **then**
5: Set the denoising vector $\Theta(k) = 1$;
6: **else**
7: Set the denoising vector $\Theta(k) = 0$;
8: **end if**
9: $k = k + 1$;
10: **end while**
11: Transform $y_i'(n)$ to the frequency domain by the DFT $Y_i(k) = \mathcal{DFT}(\widehat{y}_i(n)), k = 1, 2, \cdots, N$.
12: Get the denoised signal spectrum $\mathbb{Y}_i(k) = Y_i(k)\,\Theta(k)$;
13: **return** The denoised signal by inverse discrete Fourier transform (IDFT) $\widehat{y}_i(n) = \mathcal{IDFT}(\mathbb{Y}_i(k))$;

2) CFO Compensation: Because an oscillator is subject to temperature drift [13], the CFO is not extracted as an RFF feature in this paper. Before the feature extraction, the CFO compensation is performed, whose details can be found in [11].

After preprocessing, the discrete form of the signal can be expressed as

$$y_i'(n) = f_i'(x(n)) + u'(n), n = 1, 2, \cdots, N, \tag{2}$$

where N is the number of samples. $f_i'(\cdot)$ represents the RFF without CFO, $u'(n)$ denotes the noise after preprocessing.

B. FPS Algorithm

In order to suppress noises, we propose the FPS algorithm to perform signal denoising before the RFF feature extraction. The detailed process is illustrated in the Algorithm 1.

As shown in line 1, the standard signal $x(n)$ can be transformed to the frequency domain by the DFT as

$$X(k) = \mathcal{DFT}(x(n)) = \sum_{n=1}^{N} x(n)\, e^{-j\left(\frac{2\pi kn}{N}\right)}. \tag{3}$$

Then, the denoising vector is obtained by extracting the frequency components in the $X(k)$ whose amplitudes are larger than the threshold γ:

$$\Theta(k) = \begin{cases} 1 & if\ |X(k)| \geq \gamma \\ 0 & if\ |X(k)| < \gamma, \end{cases} \tag{4}$$

where $|\cdot|$ donates absolute operations, the threshold γ is an empirical value that can be determined using the RFF identification performance of the validation set in the training.

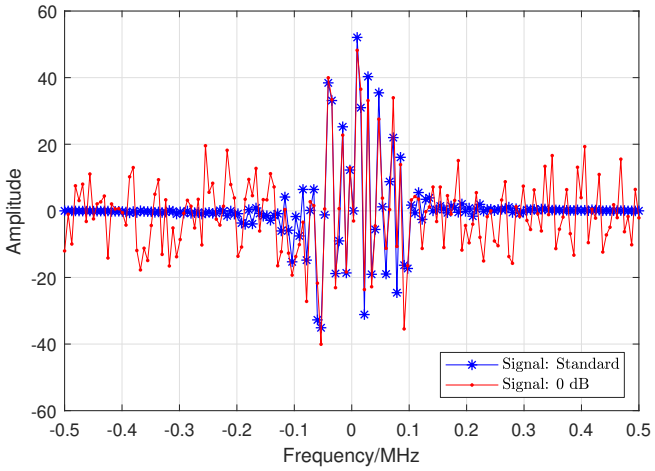

Fig. 3. Spectrum of the part signal at standard and 0 dB.

The frequency domain of signal $y_i'(n)$ is expressed as

$$\begin{aligned} Y_i(k) =& \mathcal{DFT}(y_i'(n)) \\ =& \sum_{n=1}^{N} \left(f_i'(x(n)) + u'(n)\right) e^{-j\left(\frac{2\pi kn}{N}\right)} \\ =& \sum_{n=1}^{N} f_i'(x(n))\, e^{-j\left(\frac{2\pi kn}{N}\right)} + \sum_{n=1}^{N} u'(n)\, e^{-j\left(\frac{2\pi kn}{N}\right)} \\ =& F_i(k) + U(k) \\ k =& 1, 2, \cdots, N. \end{aligned} \tag{5}$$

where $F_i(k)$ and $U(k)$ represent the spectrum of signal $f_i'(x(n))$ and noise, respectively.

In theory, the RFF function $f_i'(\cdot)$ hardly ever introduces new frequency components into the signal $x(n)$. Only the non-linear hardware may introduce a few new frequency components, which are almost negligible. Thus, the frequency components in $F_i(k)$ are considered to be a subset of the frequency components of the standard signal $x(n)$. However, the noise will introduce a large number of new frequency components that are independent of the transmitted signal. With part of the standard preamble signal as an example, Fig. 3 illustrates the spectrum of the standard signal and the signal with 0 dB noise. As it shows, the frequency components of the standard signal are concentrated in the low-frequency region. In contrast, the 0 dB signal has a lot of spurious frequency components in the high-frequency region. Therefore, noise can be suppressed by filtering these noise frequency components with the denoising vector Θ,

$$\mathbb{Y}_i(k) = Y_i(k)\,\Theta(k). \tag{6}$$

Finally, the time domain signal after noise suppression can be calculated by IDFT as

$$\widehat{y}_i(n) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{Y}_i(k)\, e^{j\left(\frac{2\pi kn}{N}\right)} \tag{7}$$

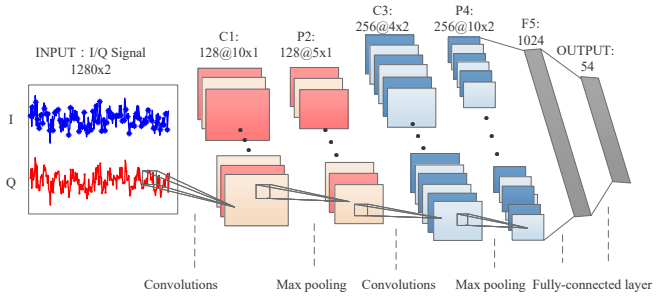

Fig. 4. The architecture of the proposed CNN.

C. CNN

CNN has attracted many research interests with the advantages of eliminating manual feature selection and excellent performance. CNN usually consists of convolutional layers, pooling layers, as well as fully-connected layers. Convolutional layers are equivalent to a feature extractor, using convolutional kernels to extract features from the input signal. The pooling layer implements data dimensionality reduction and outputs higher-level features. Finally, the fully-connected layer performs feature classification.

Referring to the classic neural network which works well on the MNIST dataset in the computer vision area and the work in [14], [15], we design a CNN model as shown in Fig. 4. The CNN contains two convolutional layers with rectified linear unit (ReLU) activation functions, containing 128 filters and 256 filters, respectively. Each convolutional layer is followed by a max pooling layer (size 5×1 and 10×2). Finally, two fully-connected layers are designed with activation functions of ReLU and softmax individually. The softmax function maps the output of the fully-connected layer, $z = (z_1, z_2, \cdots, z_M)$, to a probability list of predicted categories, $\mathbf{P} = (P_1, P_2, \cdots, P_M)$, where

$$P_i = \frac{e^{z_i}}{\sum_j^M e^{z_j}}, i = 1, 2, \cdots, M. \tag{8}$$

The category with the highest probability is selected as the predicted label

$$PredictLabel = \arg\max_i \mathbf{P}(i). \tag{9}$$

In addition, the dropout function with parameter 0.5 between the two fully-connected layers and the $L2$ regularization in the second fully-connected layer are set to improve network generalization performance. Other parameters such as optimizer, initial learning rate, the number of epochs, and batch size were set to Adam, 0.0001, 200, and 64, respectively, and the validation loss was monitored. Furthermore, we design a dynamic decreasing learning rate. Once the validation loss stops decreasing for 10 patients, the learning rate is divided by 2, which can reduce the fluctuations of the loss function in the later stages of training and makes it easier to approach the local or global optimal solution.

In our experiments, the preamble data with the length N of 1280 samples were selected as the signal to be identified.

TABLE I
RFF IDENTIFICATION PERFORMANCE OF THE PROPOSED SCHEME AT DIFFERENT THRESHOLD VALUES (TRAIN SNR = TEST SNR = 10 DB)

Threshold	0	1	2	3	4	5
Accuracy (%)	84.80	86.19	87.00	86.88	86.68	85.90

The specific parameters of the CNN were designed for this particular input signal.

IV. EXPERIMENTAL EVALUATION

The receiver used in our experiments was an Ettus Research N210 USRP software-defined radio (SDR) platform and the devices under test were 54 TI CC2530 ZigBee devices. The data collection was performed several times between 2016 and 2018 for a total of 12,238 frames of the signal were collected. These signals were collected with an SNR of about 30 dB. To verify the performance of the proposed scheme at different SNR, AWGN with different power levels, {0, 2, 4, 6, 8, 10, 12, 15, 20, 25, 30} dB, was applied to each captured signal with MATLAB. For each SNR, the ratio of the training, validation, and test sets was 6:2:2. In addition, all the training and test of our network models were running on TensorFlow 2.1.0 with an NVIDIA GeForce GTX 1660 Ti GPU.

To fully evaluate the performance of our proposed scheme, we involved three scenarios.

- Scenario I: The noise levels of the training and test data were the same, i.e., with the same SNR.
- Scenario II: we evaluate the RFF identification performance with the assumption that the training set SNR is fixed while the test set SNR varies.
- Scenario III: the training set contains signals with hybrid SNRs.

A. Scenario I

Since some frequency components in a noisy signal contain both RFF and noise, there is a trade-off between the RFF loss and the SNR gain caused by filtering out this frequency component. An amplitude threshold of the RFF frequency component is set to do the optimization. Table 1 illustrates the identification accuracy of the proposed scheme with different thresholds γ when the SNR is 10 dB. As can be seen from Table I, the variation in RFF identification accuracy at different thresholds is not significant. This indicates that the amount of RFF information in the small-amplitude RFF frequency component of the ZigBee signal is a small proportion of the total RFF. In other words, the range of reasonable thresholds is relatively large. However, in order to illustrate the best RFF recognition performance, here we select the threshold ($\gamma = 2$) corresponding to the highest accuracy rate as the parameter for all experiments in this paper.

Fig. 5 gives the identification accuracy of the CNN without denoising (Benchmark), the CNN with FPS algorithm, and the CNN with the smoothing-based denoising method in [16]. In the low SNR range (0 dB to 8 dB), the accuracy improvement brought by the FPS algorithm is significant, especially at 0 dB,

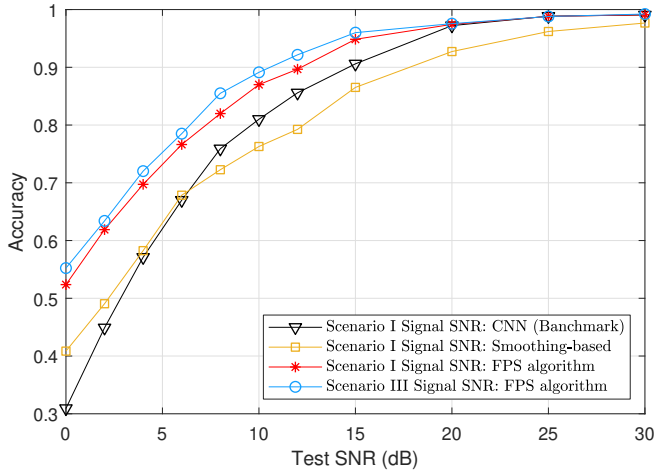

Fig. 5. Performance comparsion.

with a maximum of over 20%. In the SNR range of 10 dB to 15dB, The FPS algorithm also shows satisfactory denoising capabilities, introducing an average accuracy improvement of more than 4%. When the SNR is high (20 dB and 30 dB), the results of the CNN with the FPS algorithm are basically the same as the benchmark. The main reason is that the FPS algorithm uses a non-zero amplitude threshold γ to filter the non-RFF frequency components, which can introduce RFF information loss. Furthermore, the performance degradation due to RFF losses and the performance gains that comes from denoising in high SNR is offset.

In addition, we compared the FPS algorithm with the smoothing-based denoising method in [16]. The results show that the smoothing-based method can bring some performance improvement when the SNR is low. However, its effectiveness lags far behind that of the FPS algorithm. As the SNR rises, the smoothing-based method even leads to performance degradation. It is due to the summation operation in the smoothing process, which blurs the RFF features.

B. Scenario II

In the practical application of RFF, the environment in which the device to be tested may be variable. Therefore, the SNR of the received signal will fall in a wide range. When each signal to be identified needs to be matched with a training data set with the same noise level, the collection of training data and the training of the network will require a significant amount of work. Therefore, we consider the second scenario of RFF identification, i.e., utilizing a training set with fixed SNR to identify devices in different SNRs.

Here, we select 80% of the collected data and add different levels of AWGN to it to develop 11 training sets (including 20% for validation). The remaining 20% data are used for testing. 11 test sets with different SNRs are constructed by adding noise. This approach avoids data leakage and ensures the reasonableness of the test results. For each experiment, a training set is selected to train an optimal model, and then it is used to evaluate each test set. Table II gives the performance comparison between the benchmark CNN (ξ_c) and the CNN with the data after FPS algorithm (ξ_f) under scenario II.

Overall, ξ_f outperforms ξ_c in all training/test SNR. This is mainly attributed to the noise suppression ability of the FPS algorithm. The FPS algorithm significantly narrows the noise gap between training and testing, which means an obvious improvement in RFF identification accuracy.

When the noise of the training set is between 0 dB and 6 dB, the highest accuracy of ξ_c is about 75%. In comparison, the proposed scheme achieves an exciting performance of 92.52%. In addition, the maximum improvement is over 26% (test SNR = 30 dB, train SNR = 4 dB). when the SNR was above 8 dB for both the training and test sets, all the identification results of ξ_f exceed 85%, which shows a satisfactory performance improvement compared to the baseline.

C. Scenario III

In RFF identification, in order to train a network model with good generalization performance, the training set it is preferable to include a variety of SNR signals, which is called Scenario III in this paper.

Fig. 5 compares the FPS algorithm performance in Scenario III with the benchmark and that in Scenario I. It can be seen that when the training set is hybrid SNR, the accuracy of the FPS algorithm is greatly improved compared with the benchmark, reaching an average of 13% (from 0 dB to 10 dB), especially at 0 dB, the improvement exceeds 24%. In addition, the FPS algorithm in Scenario III shows satisfactory performance improvements in all SNR ranges compared to Scenario I. Those phenomena suggest that the training set with hybrid SNR does have better RFF identification performance than the training set with single SNR. However, only a single CNN model is required in this scenario.

V. Conclusion

In this paper, a robust RFF identification scheme is investigated. First, we propose a denoising algorithm, named FPS, to perform noise suppression while preserving RFF features by noise frequency component filtering. Then, we design a CNN for the particular ZigBee signal. Experiments show that the proposed scheme has excellent RFF recognition performance. In Scenario I, the maximum RFF identification accuracy improvement comes from the FPS algorithm is 21.45% in 0 dB. Then, the FPS algorithm shows exciting noise robustness in Scenario II. When the training set has a hybrid SNR (Scenario III), the FPS algorithm introduces an average accuracy improvement of 13% in the 0 dB to 10 dB interval.

Acknowledgement

This work was supported in part by the Research Foundation for Advanced Talents, Nanjing University of Posts and Telecommunications under Grant XK0160921022, in part by the Research Fund of Jiangsu Provincial Double-Innovation Doctor Program, and in part by National Nature Science Foundation of China under Grant 52105553. The work of J.

TABLE II
RFF IDENTIFICATION PERFORMANCE OF CNN (ξ_c) AND CNN WITH FPS ALGORITHM (ξ_f) WHEN THE SNR OF THE TRAINING AND TEST ARE DIFFERENT

Accuracy (%)		SNR of Train Set (dB)											
		0		2		4		6		8		10	
		ξ_c	ξ_f	ξ_c	ξ_f	ξ_c	ξ_f	ξ_c	ξ_f	ξ_c	ξ_f	ξ_c	ξ_f
SNR of Test Set (dB)	0	30.92	52.37	41.50	52.57	40.16	52.78	39.99	52.21	39.01	51.43	22.55	46.73
	2	32.64	52.49	44.89	61.89	53.31	62.09	51.14	61.36	52.94	59.97	38.11	56.70
	4	33.70	56.25	50.00	66.63	57.11	69.73	60.74	68.26	62.21	68.34	56.78	64.83
	6	34.07	55.76	52.17	68.59	65.36	75.69	66.99	76.63	72.63	77.04	66.91	72.75
	8	35.38	55.60	52.86	70.92	65.60	80.39	72.88	80.64	75.90	81.99	75.45	79.41
	10	34.93	56.25	54.41	71.28	65.28	82.76	73.69	85.54	79.90	85.46	81.00	87.00
	12	35.42	56.58	53.76	72.10	65.11	84.27	73.45	88.11	82.56	88.44	83.66	88.03
	15	35.50	56.17	55.47	72.79	64.83	86.32	74.63	90.11	84.76	91.54	85.87	90.93
	20	36.36	56.21	55.72	73.04	63.11	87.42	74.80	92.20	87.38	94.16	87.05	94.08
	25	36.97	56.50	55.31	72.39	62.42	88.03	75.45	92.52	88.11	94.85	86.93	94.98
	30	37.14	58.13	57.15	72.30	62.78	88.85	75.75	92.52	89.13	94.65	88.43	96.24
		12		15		20		25		30			
	0	21.12	45.92	20.79	43.75	17.48	38.77	14.75	36.23	12.09	35.95		
	2	34.40	53.88	36.19	53.23	29.82	49.31	25.41	46.00	24.96	45.55		
	4	50.37	64.42	50.61	63.52	42.28	59.44	43.46	56.66	41.87	57.76		
	6	63.36	73.45	63.73	72.47	54.45	69.53	56.90	66.42	55.72	65.89		
	8	74.18	79.09	72.79	79.53	65.32	76.96	68.75	75.74	65.03	74.35		
	10	80.84	87.05	81.17	85.46	74.84	84.48	75.82	82.39	74.80	82.03		
	12	85.58	89.67	85.70	90.77	82.43	89.01	83.82	88.56	83.09	88.52		
	15	88.89	91.87	90.60	94.85	90.77	94.44	92.24	94.24	91.42	93.50		
	20	92.24	94.81	94.12	97.30	97.22	97.43	97.55	98.24	97.55	97.39		
	25	92.93	95.51	95.18	98.16	97.79	98.37	98.85	98.82	98.39	98.33		
	30	94.85	96.57	96.53	98.94	97.84	98.41	98.87	98.90	99.07	99.10		

Zhang was in part supported by the UK EPSRC under grant ID EP/V027697/1.

REFERENCES

[1] M. Ye, N. Jiang, H. Yang, and Q. Yan, "Security analysis of Internet-of-Things: A case study of august smart lock," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Atlanta, GA, USA, Nov. 2017, pp. 499–504.

[2] A. A. Mohammadi, R. Hussain, A. Oracevic, S. M. A. R. Kazmi, F. Hussain, M. Aloqaily, and J. Son, "A novel TCP/IP header hijacking attack on SDN," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, New York, USA, Jun. 2022, pp. 1–2.

[3] B. Danev, D. Zanetti, and S. Capkun, "On physical-layer identification of wireless devices," *ACM Comput. Surv.*, vol. 45, pp. 6:1–6:29, 2012.

[4] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, "No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Trans. Cogn. Commun. Netw.*, pp. 165–178, Oct. 2020.

[5] Y. Huang, P. Liu, and J. Yang, "Radio frequency fingerprint identification method based on ensemble learning," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, New York, USA, Jun. 2022, pp. 1–6.

[6] F. Xie, H. Wen, Y. Li, S. Chen, L. Hu, Y. Chen, and H. Song, "Optimized coherent integration-based radio frequency fingerprinting in Internet of Things," *IEEE Internet Things J.*, vol. 5, pp. 3967–3977, Sep. 2018.

[7] Y. Xing, A. Hu, J. Zhang, L. Peng, and G. Li, "On radio frequency fingerprint identification for DSSS systems in low SNR scenarios," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2326–2329, Nov. 2018.

[8] G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. Cavallaro, "Towards Length-Versatile and Noise-Robust Radio Frequency Fingerprint Identification," Jul. 2022. [Online]. Available: http://arxiv.org/abs/2207.03001

[9] W. Wu, S. Hu, D. Lin, and Z. Liu, "DSLN: Securing Internet of Things through RF fingerprint recognition in Low-SNR settings," *IEEE Internet Things J.*, vol. 9, pp. 3838–3849, Jul. 2022.

[10] J. Yu, A. Hu, F. Zhou, Y. Xing, Y. Yu, G. Li, and L. Peng, "Radio Frequency Fingerprint Identification Based on Denoising Autoencoders," in *2019 Int. Conf. Wireless Mobile Comput. Networking Commun. (WiMob)*, Barcelona, Spain, 2019, pp. 1–6.

[11] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid RF fingerprint extraction and device classification scheme," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 349–360, Feb. 2019.

[12] G. Li, J. Yu, Y. Xing, and A. Hu, "Location-invariant physical layer identification approach for WiFi devices," *IEEE Access*, vol. 7, pp. 106 974–106 986, Aug. 2019.

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

[14] J. Yu, A. Hu, G. Li, and L. Peng, "A robust RF fingerprinting approach using multi-sampling convolutional neural network," *IEEE Internet Things J.*, vol. 6, pp. 6786–6799, Apr. 2019.

[15] Y. Xing, A. Hu, J. Zhang, L. Peng, and X. Wang, "Design of a channel robust radio frequency fingerprint identification scheme," *IEEE Internet Things J*, pp. 1–1, 2022.

[16] W. Wang and L. Gan, "Radio Frequency Fingerprinting Improved by Statistical Noise Reduction," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 3, pp. 1444–1452, Sep. 2022.