2023 28th Asia Pacific Conference on Communications (APCC) | 979-8-3503-8261-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/APCC60132.2023.10460605

An Efficient RFF Extraction Method Using Asymmetric Masked Auto-Encoder

1st Zhisheng Yao
College of Telecommunications and Information Engineering
NJUPT, Nanjing, China
1023010412@njupt.edu.cn

2nd Xue Fu
College of Telecommunications and Information Engineering
NJUPT, Nanjing, China
1020010415@njupt.edu.cn

3rd Shufei Wang
College of Telecommunications and Information Engineering
NJUPT, Nanjing, China
1021010525@njupt.edu.cn

4th Yu Wang
College of Telecommunications and Information Engineering
NJUPT, Nanjing, China
1018010407@njupt.edu.cn

5th Guan Gui
College of Telecommunications and Information Engineering
NJUPT, Nanjing, China
guiguan@njupt.edu.cn

6th Shiwen Mao
Department of Electrical and Computer Engineering
Auburn University, Auburn, USA
smao@ieee.org

***Abstract*—Radio frequency fingerprint (RFF) has been widely used in wireless transceivers as an additional physical security layer. Most of the existing RFF extraction methods rely on a large number of labeled signal samples for model training. However, in real communication environments, it is usually necessary to process timely received signal samples, which are limited in quantity and are difficult to obtain labels, the performance of most RFF methods is generally poor. To effectively extract features from the limited and unlabeled signal samples, we propose an efficient RFF extraction method using an asymmetric masked auto-encoder (AMAE). Specifically, we design an asymmetric extractor-decoder, where the extractor is used to learn the latent representation of the masked signals and the decoder as light as a convolution layer reconstructs the unmasked signal from the latent representation. Using commercial off-the-shelf LoRa datasets and WiFi datasets, we show that the proposed AMAE-based RFF extraction method achieves the best performance compared with four advanced unsupervised methods whether in the case of large data size or small data size, or under line of sight (LOS) and non line of sight (NLOS) channel scenarios. The codes of this paper can be downloaded from Github: https://github.com/YZS666/An-Efficient-RFF-Extraction-Method.**

***Index Terms*—Radio frequency fingerprint (RFF), unsupervised learning, asymmetric masked auto-encoder (AMAE).**

I. Introduction

In wireless communications, due to the openness of wireless channels, the large amount of mobile devices, and the emergence of the Internet of Things (IoT), various physical layer security problems arise, such as medium access control (MAC) address attack, impersonation attack, and so on [1]. The attacker may tamper with the identification information such as password or MAC address, to spoof wireless devices, which deceives the receiver and threatens the security of the device identity. In addition, the complex algorithm of traditional defensive mechanisms, such as protocol analysis, brings an additional costs and resource consumption [2]. Hence, to enhance communication security with a lightweight computing method, the radio frequency fingerprint (RFF) technology based on the unique characteristics of the electromagnetic wave emitted by the transmitter is proposed [3]. Unique characteristics such as in-phase and quadrature (I/Q) origin offset, frequency error, synchronization correlation, etc. are difficult to tamper with and can be used for transmitter identification [4].

However, traditional RFF extraction methods rely on prior expert knowledge, which are usually only applicable for situations with limited number of parameters [5]. In recent years, the emergence of deep learning (DL) overcomes the limitations. DL models can automatically extract high-dimensional signal features in an end-to-end manner, which is conducive to further classify signals [6], [7]. Wang *et al.* [8] used a complex-valued neural network (CVNN) and network compression to identify seven power amplifiers. Shen *et al.* [9] used deep metric learning and the k-NN algorithm to classify and detect 60 malicious commercial LoRa devices. However, these DL methods require a large number of labeled signal samples to train a robust extractor. When the labeled signal samples are insufficient, the performance will sharply decrease. Fu *et al.* [10] proposed a semi-supervised method, which used metric-adversarial training to identify the automatic-dependent surveillance-broadcast devices and WiFi devices.

In addition, unsupervised learning [11] such as auto-encoder (AE) [12], masked auto-encoder (MAE) [13]–[15], deep embedding clustering (DEC) [16], variational auto-encoder (VAE) [17] and deep convolution generative adversarial network (DCGAN) [18] is introduced into solving the problem of missing labels. Xie *et al.* [12] preprocessed the original signals using Hilbert Huang transform (HHT), and use AE for RFF feature extraction. Huang *et al.* [14] used the symmetric mask auto-encoder (SMAE) to learn RFF features, using the residual network as the backbone network. However, the symmetric decoder and residual network which have high floating-point operations (FLOPs) and large model sizes, requires excessive computing resources and training time.

Considering that the actual communication system needs to be able to quickly respond to the captured signal while also hoping to reduce computational costs of the device, a lightweight RFF extraction method based on lightweight model driven by unlabeled signal samples is crucial for practical applications. In this paper, we propose an effective RFF extraction method based on an asymmetric masked auto-encoder (AMAE), which is a simple and scalable self-supervised method. The main contributions of this paper are summarized as follows:

- We propose an RFF extraction method that effectively extract the RFF of emitters in an unsupervised way, which does not rely on sufficient signal samples and corresponding labels to driven the training process of deep model.
- We innovatively design a decoder as light as just a convolutional layer, that reconstructs the unmasked signals from the latent representation of masked signals with a better performance than symmetric models and brings a remarkable speedup in the training process.
- The proposed AMAE-based RFF extraction method is evaluated on commercial LoRa datasets and WiFi datasets. The simulation results show that our AMAE-based RFF extraction method achieves the best robustness and generalization while minimizing computational costs.

The remaining components of the paper are as follows. Section II describes the signal model and problem formulation. In Section III, we show the details of proposed AMAE-based RFF extraction method. In Section IV, we present the simulation results including comparision with the other unsupervised method. Finally, we concludes the paper and point out the future work in Section V.

II. Signal Model and Problem Formulation

A. Signal Model

M emitters are activated for RFF extraction, and their radio frequency signals are captured by a receiver respectively. The received signals from the m-th emitter can be expressed as

$$x_m(t) = h(t) * f_m(s(t)) + n(t), \tag{1}$$

where $x_m(t),\ m \in \{0, 1, \cdots, M-1\}$, is the received signals, $s(t)$ is the transmitted signals, $f_m(\cdot)$ is the effect of specific damage caused by the hardware components for signal modulation and up-conversion, $h(t)$ stands for the time-varying wireless channel impulse response, $*$ denotes the convolution operation, and $n_m(t)$ is the additive white Gaussian noise. We obtain discrete-time samples which is defined as

$$\mathbf{x}_k[n] = \mathbf{x}_k(nT_s),\ n \in \{0, 1, \cdots, N-1\}, \tag{2}$$

where N is the number of sampling points and T_s represents the sampling interval.

B. Problem Formulation

Let $\mathcal{X}$ and $\mathcal{Z}$ be the sample space and feature space, respectively. RFF extractor can be regarded as a mapping function $f_e \in \mathcal{F} : \mathcal{X} \to \mathcal{Z}$. The decoder can be regarded as a mapping function $f_d \in \mathcal{F} : \mathcal{Z} \to \bar{\mathcal{X}}$ where $\bar{\mathcal{X}}$ denotes the reconstructed sample space. When a signal sample $\mathbf{x}_k$ ($\mathbf{x}_k \in \mathcal{X}$) is the input, a feature vector $\mathbf{z}_k$ ($\mathbf{z}_k \in \mathcal{Z}$) of the signal sample is output through f_e, and the reconstructed signal sample $\bar{\mathbf{x}}_k$ ($\bar{\mathbf{x}}_k \in \bar{\mathcal{X}}$) is output through f_d, where f_d and f_e is optimized by minimizing the mean square error (MSE) between the $\bar{\mathbf{x}}_k$ and $\mathbf{x}_k$, which can be written as

$$\min_{f_r(f_e)\in\mathcal{F}} \mathbb{E}_{(\mathbf{x}_k,\bar{\mathbf{x}}_k)\sim \boldsymbol{D}}\{\mathcal{L}_{mse}[\mathbf{x}_k, f_r(f_e(\mathbf{x}_k))]\}, \tag{3}$$

where D is the training dataset and $\mathcal{L}_{mse}$ is MSE loss.

III. The Proposed RFF extraction Method

A. Masked Representation Learning

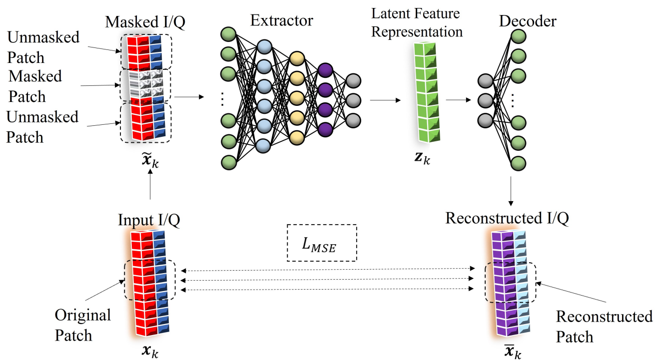


Fig. 1. The framework of the proposed AMAE-based RFFE method.

According to the convolution property, masked representation learning applies a mask on the input data to restrict the model from seeing only partial information of the input data, and reconstruct a masked block by sensing the higher-order feature representation of neighboring blocks. The principle of AMAE is to learn the compressed representation of data, which uses a masking mechanism to force the model to only focus on certain subsets of input data, having specific applications. In this paper, we propose a RFF extraction method based on AMAE, which is shown in Fig. 1. Firstly, the signal samples are randomly masked using a patch, which can be formulated as

$$\mathbf{m}_k[n] = 0 \text{ or } 1,\ n \in \{0, 1, \cdots, N-1\}, \tag{4}$$

$$\gamma = 1 - \frac{\sum_{n=0}^{N-1} \mathbf{m}_k[n]}{N}, \tag{5}$$

$$\tilde{\mathbf{x}}_k[n] = G(\mathbf{x}_k[n]) = \mathbf{m}_k[n] \odot \mathbf{x}_k[n], \tag{6}$$

where $\mathbf{m}_k$ represents the mask block for k-th signal sample, and the length of mask block is the same as that of signal sample, γ is the defined hyper-parameter of mask ratio, $\odot$ indicates element-wise product, and $\tilde{\mathbf{x}}_k$ is the masked signals, where G refers to mask operation. Secondly, the masked signals are compressed into a latent feature representation $\mathbf{z}_k$ using an extractor and then the missing patch is reconstructed using a decoder. The MSE loss are used to evaluate the effect of masked representation learning and can be expressed as

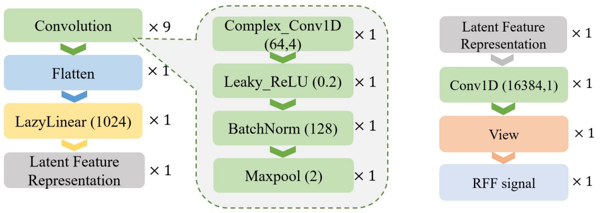
$$\mathcal{L}_{mse}(\mathbf{x}_k, \tilde{\mathbf{x}}_k) = \frac{1}{N_t}\sum_{k=0}^{N_t-1} \left\|(\mathbf{x}_k - \bar{\mathbf{x}}_k) \odot (1 - \mathbf{m}_k)\right\|^2, \tag{7}$$

where $\|\cdot\|$ represents the mold operation, N_t represents the number of training samples. Finally, the Adam optimizer updates the extractor and decoder to minimize the above MSE loss, which means that the parameters of the extractor and decoder are updated at the same time.

978-1-7281-7254-5/21/$31.00 ©2021 IEEE

B. Asymmetric Autoencoder

1) RFF Extractor: The signal sample, composed of in-phase (I) component and quadrature (Q) component, can be regarded as a complex number. However, standard convolutional neural networks cannot be directly operation for complex number. To fully exploit the potential information of the signal sample, we present an effective RFF extractor using complex valued convolutional layer, which is shown in Fig. 2(a). Specifically, we use nine complex valued convolutional blocks for extracting signal information between I and Q component. The Leaky ReLU [20] is used as activation function which makes the negative end incompletely suppressed, so as to solve the problem of gradient disappearance or gradient explosion. LazyLinear [21] further compresses the dimension of latent feature representation to 1,024.



(a) RFF extractor (b) Asymmetric decoder

Fig. 2. The structure of the proposed RFF extractor and asymmetric decoder.

2) Decoder: The design of the decoder can be very flexible, as it can be symmetric or asymmetric with the encoder [22]. To reduce the complexity caused by complex-valued RFF extractor, we design an asymmetric decoder as shown in Fig. 2(b). The decoder first converts the size and shape of the latent feature representation into the size of the input signal through a standard convolutional layer, and the view function. This lightweight design makes the training process more focused on the RFF extraction ability of the extractor, while reducing the computational complexity and having shorter training time.

C. Training Procedure

The full training procedure of the proposed AMAE-based RFF extraction method is described in **Algorithm 1**.

IV. Simulation Results And Analysis

A. Simulation Parameters

We use the LoRa dataset [9] and WiFi dataset [23] for simulation, the detailed simulation parameters are shown in Table. I. Silhouette coefficient (SC) is utilized to indicate the RFF extraction performance. The formula of the SC is given as

$$SC = \frac{1}{N_{te}} \sum_{k=0}^{N_{te}-1} \frac{b_k - a_k}{\max\{a_k, b_k\}}, SC \in [-1, 1] \quad (8)$$

where N_{te} represents the number of samples in the test dataset, a_k represents the average distance between the k-th sample and other samples within the cluster; b_k represents the minimum average distance between the k-th sample and samples from other clusters. The larger the value, the better feature extraction performance. The proposed AMAE-based RFF extraction method is compared with four currently existing unsupervised methods [11], including AE [12], DEC [16], VAE [17], DCGAN [18]. We discuss the RFF extraction performance of different methods in different data sizes and channel scenarios.

Algorithm 1: Training procedure of proposed AMAE-based RFF extraction method.

Input:
- D: Training dataset;
- $\tilde{\mathbf{x}}_k$: Sample of masked signal;
- T: Number of training iterations;
- B: Number of batches in a training iteration;
- $\boldsymbol{\theta}_e$, $\boldsymbol{\theta}_d$: Parameters of the extractor and decoder, respectively;
- lr_e, lr_d: Learning rate of the extractor and decoder, respectively;

Training procedure:
for $t = 1$ *to* T **do**
  for $b = 1$ *to* B **do**
    Sample a batch training dataset from D;
    Forward propagation:
    Get the masked signal samples of $\mathbf{x_k}$:
    $\tilde{\mathbf{x}}_k = G(\mathbf{x_k})$;
    Get the latent feature representation:
    $\mathbf{z_k} = f_e(\boldsymbol{\theta}_e^{t,b}; \tilde{\mathbf{x}}_k)$;
    Get the reconstructed signals:
    $\bar{\mathbf{x}}_k = f_d(\boldsymbol{\theta}_d^{t,b}; \mathbf{z_k})$;
    Calculate the loss: $\mathcal{L}_{mse}(\bar{\mathbf{x}}_k, \mathbf{x}_k)$;
    Backward propagation:
    Updating the parameters of the extractor:
    $\boldsymbol{\theta}_e^{t,b+1} \leftarrow \text{Adam}(\nabla_{\boldsymbol{\theta}_e}, \mathcal{L}, lr_e, \boldsymbol{\theta}_e^{t,b})$;
    Updating the parameters of the decoder:
    $\boldsymbol{\theta}_d^{t,b+1} \leftarrow \text{Adam}(\nabla_{\boldsymbol{\theta}_d}, \mathcal{L}, lr_d, \boldsymbol{\theta}_d^{t,b})$;
  end
end

TABLE I
SIMULATION PARAMETERS

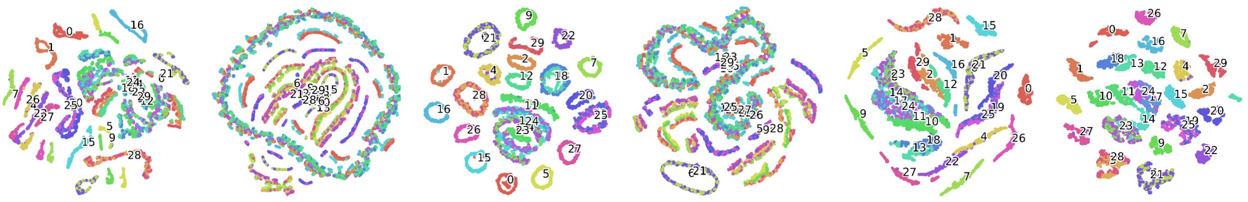
Dataset	LoRa (LOS)	WiFi (LOS)	LoRa (NLOS)
Categories	30	16	10
Training Samples	13,500	43,208	1,800
Validation Samples	1,500	4,801	200
Length of Each Sample	8,192	6,000	8,192
Signal Bandwidth	125 KHz	56 KHz	125 KHz
Oversampling Ratio	8 Ms/s	5 Ms/s	8 Ms/s
Pytorch	1.10.2		
Python	3.6.13		
Signal Format	IQ		
Training vs. Validation	9:1		
Optimizer	Adam		
Epochs	300		
Batch Size	128		
Learning Rate	0.001		
Platform	NVIDIA GeForce GTX 3090Ti GPU		

Note: LOS represents line of sight, NLOS represents non line of sight.

TABLE II
SC ON THE LORA AND WIFI DATASET UNDER **Asymmetric / Symmetric** DECODER

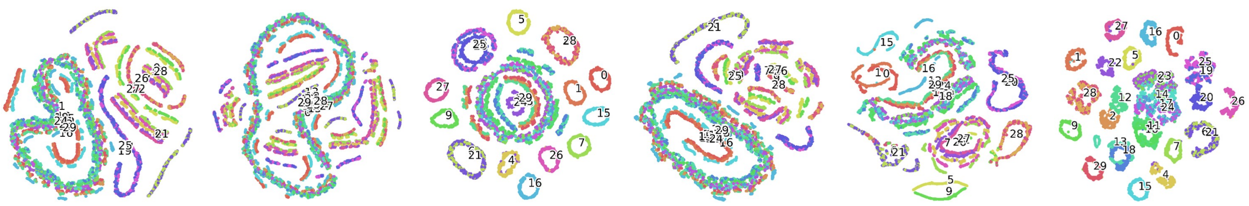
Methods	LoRa (LOS)		WiFi (LOS)		LoRa (NLOS)	
	30-way	**30-way (FS)**	**16-way**	**16-way (FS)**	**10-way**	**10-way (FS)**
AE [12]	0.0035 / -0.01824	-0.0365 / -0.0499	0.1390 / -0.0206	0.0126 / -0.0065	0.0336 / 0.0189	-0.0201 / 0.0757
DEC [16]	-0.0931 / -0.1100	-0.1159 / -0.0171	0.2790 / 0.0131	0.0332 / -0.0326	-0.0046 / 0.5348	-0.0897 / 0.5007
VAE [17]	0.2225 / **0.4676**	0.1398 / 0.2674	0.2230 / 0.2653	0.2211 / 0.2458	-0.3438 / 0.3332	0.3941 / 0.2748
DCGAN [18]	-0.0280 / -0.1496	-0.0365 / 0.0100	-0.0042 / -0.0154	-0.0042 / -0.0076	0.0196 / 0.0368	-0.0537 / -0.0275
AMAE (proposed)	0.4053 / 0.1772	**0.3401** / -0.0533	**0.3688** / 0.0323	**0.2460** / 0.0469	**0.5543** / 0.2647	**0.5428** / 0.2163

Note: FS refers to 20 samples per class.



(a) AE [12] (b) DEC [16] (c) VAE [17] (d) DCGAN [18] (e) SMAE [14] (f) **AMAE (proposed)**

Fig. 3. Visualization of latent feature representation of different unsupervised methods on the LoRa dataset with **sufficient data size**, where AE, DEC, DCGAN, VAE and AMAE are trained with the **asymmetric** decoder, while SMAE is trained with the symmetric decoder.



(a) AE [12] (b) DEC [16] (c) VAE [17] (d) DCGAN [18] (e) SMAE [14] (f) **AMAE (proposed)**

Fig. 4. Visualization of latent feature representation of different unsupervised methods on the LoRa dataset with **few-shot data size**, where AE, DEC, DCGAN, VAE and AMAE are trained with the **asymmetric** decoder, while SMAE is trained with the symmetric decoder.

B. Robustness of Proposed Method

Robustness is the ability of the model to stably extract RFF features even when there is disturbance or distribution bias in the input signal. Table II shows that the proposed AMAE-based RFF extraction method has better RFF extraction performance than other methods in both LOS and NLOS scenarios when using extractors trained with asymmetric decoders. Meanwhile, as the channel scenario becomes more complex which means having a lower signal-to-noise ratio, the RFF extraction ability of other methods such as VAE decreases significantly, while the proposed AMAE-based RFF extraction method still has good RFF extraction ability.

C. Generality of Proposed Method

Analyzing the impact of data size on the performance of the proposed method, the proposed method still has better RFF extraction performance than the other four comparison methods, especially in scenario with limited data size. Specifically, in the LOS scenario of the LoRa dataset, when the data size decreases, the performance of the VAE method sharply decreases because VAE uses latent layer distribution to reconstruct the original signal, and its modeling error will increase as the data size decreases, resulting in poorer RFF extraction performance compared to the proposed AMAE based RFF extraction method when the data size is limited.

D. Effectiveness of Asymmetric Decoder

It can be seen in Table II that in most cases in this paper, the performance of extractors trained with asymmetric decoder is better than that trained with symmetric decoder, which indicates that the asymmetric decoder is more capable of training high-performance RFF extractors. Compared with the VAE method trained with the symmetric decoder in the LOS scenario of the LoRa dataset, the proposed AMAE-based RFFE method has lower SC. However, asymmetric VAE has poorer robustness and generalization. At the same time, it requires to calculate an additional latent layer distribution, and using a heavyweight decoder, which requires more computational resources and longer training time, which will be analyzed in the next sub-section.

E. RFF Feature Visualization

The dimensionality of the extracted features is reduced to two dimensions by t-distributed stochastic neighbor embedding (t-SNE) [24] for visualization, which is shown in Fig. 3, Fig. 4. We show the visualization in the LOS scenario of LoRa dataset, and the number of categories is 30. Obviously, the proposed method has better semantic features in inter-class dispersion and intra-class compactness, while the comparative methods roughly separates the extracted features of different categories.

F. Computational Cost Analysis

To compare the computational costs of different algorithms, as shown in Table. III, we chose FLOPs, training

978-1-7281-7254-5/21/$31.00 ©2021 IEEE

TABLE III
THE COMPUTATIONAL COST ANALYSIS OF PROPOSED METHOD AND COMPARATIVE METHODS USING **Asymmetric / Symmetric** DECODER

Methods	FLOPs	Size (MB)	Training time (s/epoch)	Computational Memory (MB)
AE [12]			44.3 / 71.2	8,513 / 11,333
DEC [16]			71.6 / 110.5	11,327 / 12,529
VAE [17]	**5,669,068,800** / 12,646,645,760	**71,793** / 104,755	49.7 / 64.1	8,545 / 11,335
DCGAN [18]			127.4 / 142.0	13,587 / 13,690
AMAE (Proposed)			**41.1** / 63.8	**8,501** / 11,327

Note: s/epoch refers to the time required to train an epoch.

time and computational memory as indicators. DEC requires two-stage training, including AE and deep embedded clustering, which requires multiple use of kmeans function to calculate the sample center to update model. VAE needs to calculate potential distribution, which needs to calculate Kullback-Leibler (KL) divergence and MSE loss. DCGAN needs to distinguish between true samples and false samples, and needs to calculate lots of loss function. The training process of proposed AMAE is same as that of AE, without the need for additional calculations on latent feature representations, and only uses MSE loss to update model parameters. Besides, AE, DEC, VAE, and DCGAN all needs to calculate the MSE loss for all parts of the sample, but the proposed AMAE only needs to minimize the unmasked part of the MSE loss used for training, which have lower computational memory and can train a RFF extractor in a shorter training time.

To further demonstrate the lightness of asymmetric decoder, we choose model size as indicator for our analysis. All the comparison methods ensure that the backbone network of the extractor is the same. Therefore, these methods have the same model size in the extractor part, and the decoder is divided into symmetric decoder and asymmetric decoder, which is the key to different methods having different model size. The asymmetric decoder composed of only one convolutional layer, which has lower smaller model size.

V. CONCLUSION

In this paper, we proposed an unsupervised RFF extraction method based on AMAE. Specifically, the RFF extractor is composed of nine complex-value convolutional blocks and a full connected layer, and the decoder only uses a real-value convolutional layer, which achieves a better performance in a lightweight way. Moreover, the proposed RFF extraction method was evaluated on LoRa dataset [9] and WiFi dataset [23] and compared with four latest unsupervised methods. We showed that the proposed AMAE-based RFF extraction method has the best robustness, generalization and lightweight. In addition, feature visualization demonstrates the feature separation of RFF extraction. In future work, we will use the RFF extractor to solve downstream tasks, such as few-shot RFF identification, and pay more attention to explore the bounds on number of emitters.

REFERENCES

[1] X. Lu, *el al.*, "Improved physical layer authentication scheme Based on wireless channel phase," *IEEE Wireless Commun. Lett.*, vol. 11, no. 1, pp. 198–202, Jan. 2022.

[2] P. Angueira, *et al.*, "A survey of physical layer techniques for secure wireless communications in industry," *IEEE Commun. Surv. Tutorials*, vol. 24, no. 2, pp. 810–838, Secondquarter 2022.

[3] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," *IEEE Can. J. Electr. Comput. Eng.*, vol. 32, no. 1, pp. 27–33, 2007.

[4] V. Brik, *et al.*, "Wireless device identification with radiometric signatures," in *14th ACM MobiCom*, 2008, pp. 116–127.

[5] J. M. McGinthy, *el al.*, "Groundwork for neural network-based specific emitter identification authentication for IoT," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6429–6440, 2019.

[6] A. Krizhevsky, *el al.*, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 25, no. 1, pp. 9, 2012.

[7] Y. Tu, Y. Lin *el al.*, "Large-scale real-world radio signal recognition with deep learning," *Chinese J. Aeronaut.*, vol. 35, no. 9, pp. 35–48, Sep. 2022.

[8] Y. Wang, *et al.*, "An efficient specific emitter identification method based on complex-valued neural networks and network compression," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2305–2317, Aug. 2021.

[9] G. Shen, *el al.*, "Towards scalable and channel-robust radio frequency fingerprint identification for LoRa," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, no. 14, pp.774–787, Jun. 2022.

[10] X. Fu, *et al.*, "Semi-supervised specific emitter identification method using metric-adversarial training," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10778–10789, 2023.

[11] S. Bond-Taylor, *el al.*, "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7327–7347, Nov. 2022.

[12] C. Xie *et al.*, "Few-shot unsupervised specific emitter identification based on density peak clustering algorithm and meta-learning," *IEEE Sens. J.*, vol. 22, no. 18, pp. 18008–18020, 2022.

[13] K. He *et al.*, "Masked autoencoders are scalable vision learners," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 16000–16009.

[14] K. Huang, *et al.*, "Deep learning of radio frequency fingerprints from limited samples by masked autoencoding," *IEEE Wireless Commun.*, vol. 14, no. 8, pp. 40–47, Sep. 2021.

[15] Z. Yao *et al.*, "Few-shot specific emitter identification using asymmetric masked auto-encoder," *IEEE Commun. Lett.*, early access, doi: 10.1109/LCOMM.2023.3312669

[16] X. Guo, *et al.*, "Deep clustering with convolutional autoencoders," in *Int. Conf. Neural. Inf. Proc.*, Nov. 2017, pp. 373–382.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Representations*, Dec. 2022, pp. 1–13.

[18] J. Li, *et al.*, "Unsupervised representation learning of image-based plant disease with deep convolutional generative adversarial networks," in *Conf. Comput. Commun.*, Oct. 2018, pp. 9159–9163.

[19] T. J. O'Shea, *et al.*, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[20] A. L. Maas, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," *Int. Conf. Mach. Learn.*, vol. 30, no. 1, p. 3, 2013.

[21] S. K. Jha, "Reachability Analysis of Lazy Linear Hybrid Automata," *Turk. J. Electr. Eng. Comput. Sci.*, no. UCB/EECS-2011-112, pp. 1–34, Oct. 2011.

[22] Z. Xie, *et al.*, "Simmim: A simple framework for masked image modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 9653–9663.

[23] K. Sankhe, *et al.*, "ORACLE: Optimized radio classification through convolutional neural networks," in *Proc. Int. Conf. Learn. Representations*, Apr. 2019, pp. 370–378.

[24] L. Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

978-1-7281-7254-5/21/$31.00 ©2021 IEEE