MASSnet: Deep-Learning-Based Multiple-Antenna Spectrum Sensing for Cognitive-Radio-Enabled Internet of Things

Luxin Zhang, Shilian Zheng, Kunfeng Qiu, Caiyi Lou, and Xiaoniu Yang

***Abstract*—Cognitive radio-based Internet of Things (CR-IoTs) provide an efficient spectrum management for IoT networks with massive wireless access and data transmission needs. As one of the key technologies of CR-IoT, spectrum sensing is of great research significance. Motivated by the recent boom on applications of deep learning (DL) in wireless communications networks and IoT, several spectrum sensing methods based on DL have emerged. However, these algorithms train the sensing models with the extracted features of received signals and require a retraining of sensing models when the number of sensing antennas changes. Thus, we develop multiple-antenna spectrum sensing methods based on convolutional neural networks (MASSnets) using the in-phase (I) and quadrature (Q) components of the signals as the input. The three schemes of MASSnet also provide the flexibility to choose between retraining the sensing models or using the obtained models for different sensing antenna configurations. Experiment results demonstrate the superior performance of the proposed methods over existing DL-based spectrum sensing methods in terms of probability of detection especially in very low-signal-to-noise ratio (SNR) condition. Furthermore, the proposed methods have good generalization ability to new noise distribution, new fading channel, different frequency offsets, and detecting signals with a new modulation even without retraining.**

***Index Terms*—Cognitive radio (CR), deep learning (DL), Internet of Things (IoT), multiple antenna, spectrum sensing.**

I. Introduction

THE PAST decades have witnessed an exponential growth of Internet of Things (IoT) devices. The massive connections and data transmission for IoT have taken the burden to the wireless communication services. Hence, the demand for spectrum resource management and transmission bandwidth utilization is greater than ever before [1]. The cognitive radio-based IoT (CR-IoT) is expected to become an effective way to address the lack of spectrum problem [2]. The CR-IoT utilizes a two-step method to deal with the data traffic for IoT [3]. Specifically, the spectrum sensing technique is first used to judge whether there is available spectrum for secondary users (SUs) via sensing the presence of the primary user (PU). Then CR-IoT optimizes the data transmission after obtaining the licensed spectrum [4]. By exploiting cognitive radio (CR) to IoT, cognitive IoT devices can access the unoccupied frequency band so that the problem of spectrum shortage can be effectively alleviated. The multiaccess and high-interaction characteristics of the IoT also bring great challenges to spectrum sensing.

Spectrum sensing for CR systems has been widely studied in [5], [6], [7], [8], and [9]. Most studies obtain spectrum sensing results via comparing detection statistics with detection thresholds. The detection statistics are usually designed based on the distinguishing characteristics between signals and noise [10], [11], [12], [13]. For instance, matched filter detection (MFD) [14], cyclostationary feature detection (CFD) [15], energy detection (ED) [16] and covariance-based detection (CBD) [17] use various test statistics and they are typical solutions which are widely used in traditional spectrum sensing methods. These methods have their own applicable conditions and coexist with advantages and disadvantages. MFD can obtain a satisfactory detection precision in low-signal-to-noise ratio (SNR) conditions by utilizing the correlation between the information of the ideal primary signal and the received signals. However, this requires having to know the perfect signal information of the PU in advance. CFD also works well in low-SNR situations but it needs to calculate the cyclostationary features of the signals which takes a lot of computing resources. This makes CFD insufficient in CR systems with high-real-time requirements especially in IoT networks. As a popular spectrum sensing method due to simple realization without any prior knowledge of PU's signals, ED calculates the energy of the received signals of SU to sense the presence of the PU via comparing the energy with the detection threshold. But the detection performance is susceptible to noise uncertainty. Like ED, CBD also qualifies as a blind detection technique. In most CBD methods, there is no requirement for a prior knowledge regarding noise power. This characteristic of CBD significantly alleviates the impact of noise power uncertainty, making it a highly attractive advantage when compared to ED. Nevertheless, it should be noted that CBD hinges on the fundamental assumption of signal sample correlation, and its applicability may be limited in scenarios where signal samples are statistically independent.

To further improve the detection performance, some researchers have also focused on applying deep learning (DL) in spectrum sensing. Because DL has been successfully applied to wireless communications system and IoT networks

Manuscript received 7 April 2023; revised 13 August 2023 and 7 November 2023; accepted 8 December 2023. Date of publication 18 December 2023; date of current version 9 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U19B2016 and Grant U20B2038. *(Corresponding authors: Shilian Zheng; Xiaoniu Yang.)*

The authors are with the Innovation Studio of Academician Yang, National Key Laboratory of Electromagentic Space Security, Jiaxing 314033, China (e-mail: lxzhangMr@126.com; lianshizheng@126.com; yexijoe@163.com; loucycetc@163.com; yxn2117@126.com).

Digital Object Identifier 10.1109/JIOT.2023.3343699

for dealing with image classification [18], speech recognition [19], signal detection [20], and signal classification [21] etc. DL-based spectrum sensing methods utilize deep neural networks (DNNs) to intelligently extract features of input data for distinguishing the presence or absence of the PU. However, in the realm of existing DL-based sensing methods, a majority of them utilize extracted features, such as covariance or correction matrices, derived from the received signals, as the input for DNNs. For data-driven schemes, it is indeed possible to utilize raw signal data as input, which contains more comprehensive information than certain extracted features. To the best of our knowledge, in multiantenna spectrum sensing, there are no existing studies that use raw in-phase (I) and quadrature (Q) data of the signals as the input of neural networks. To fill the gap, in this article, we explore multiple-antenna spectrum sensing approaches based on convolutional neural network (CNN), namely, MASSnet, using raw IQ data as the input. We believe that the DNNs can learn more hidden features from raw IQ data than from the preprocessed features. Experimental results validate that the proposed methods have better detection performance than the DL methods based on covariance (CM-CNN) [22] or correlation matrices (DS2MA) [23].

Given the diverse array of devices connecting to CR-IoT networks, it is likely that different types of devices will be equipped with varying numbers of antennas. Hence, it is necessary to investigate spectrum sensing algorithms for different receiving antenna configurations. In the existing literatures, spectrum sensing schemes are typically addressed for a specific number of receiving antennas. When the number of antennas changes, they require the retraining of sensing models. Nevertheless, developing distinct spectrum sensing models for each device might prove impractical in practical implementations, considering the laborious task of obtaining labeled training data and the resource-intensive training process. A more sensible approach is to develop efficient algorithms that facilitate the reuse of sensing models across different devices. Thus, we devise three schemes of MASSnet, namely, MASSnet-B, MASSnet-F, and MASSnet-A, to provide SUs of CR-IoT with the flexibility to choose between retraining the sensing model or using the obtained model based on their specific circumstances and requirements for CR-IoT devices equipped with different sensing antennas. Especially, MASSnet-B trains sensing models to with the certain number of sensing antennas. MASSnet-F and MASSnet-A have the capability to train spectrum sensing models that can be easily extended for performing spectrum sensing on various CR-IoT devices. Extensive experiments are carried out to evaluate the performance of the proposed methods.

In summary, the main contributions of this article are as follows.

1) We propose multiple-antenna spectrum sensing methods that utilize the raw IQ data of the signals as the input of CNN. CNN can learn more hidden features from raw IQ data compared to preprocessed features like covariance or correlation matrices. For online detection, the detection threshold can be set according to the desired false alarm probability. Experimental results confirm that the proposed methods outperform the feature-extraction-aware methods in terms of detection performance.
2) We propose three schemes of MASSnet, namely, MASSnet-B, MASSnet-F, and MASSnet-A, to cater to different devices with various numbers of receiving antennas in CR-IoT. MASSnet-B aims to achieve high-detection performance by training the specific CNN detector for each devices. MASSnet-F and MASSnet-A provide solutions that enable the reusability of obtained sensing models across different devices. MASSnet-F is simple to implement, as it is trained using IQ data from a single receiving antenna and then extended to others through feature fusion. MASSnet-A also requires training only once and can effortlessly adapt to any number of antennas. It achieves improved detection performance by using an adaptive feature extraction module that explores the correlation among different sensing antennas.
3) We conduct extensive experiments with the proposed three schemes of MASSnet in various detection conditions. For the situation of variable number of sensing antennas, MASSnet-B can obtain an SNR gain of about 3 dB when the number of antennas is doubled and consistently have a better detection performance than CM-CNN and DS2MA. MASSnet-A and MASSnet-F also get the incremental performance from the increasing number of antennas without retraining and MASSnet-A outperforms CM-CNN method even though CM-CNN has to be retrained when the number of sensing antennas is changed. Furthermore, experimental results show that the proposed schemes of MASSnet have good generalization ability to new noise distribution, new fading channel, different frequency offsets, and detecting signals with a new modulation even without retraining.

The remainder of this article is organized as follows. We discuss the related work in Section II and describe the basic problem of multiple-antenna spectrum sensing in Section III. In Sections IV, V, and VI, three schemes of MASSnet are discussed in detail. Section VII provides simulation results of the proposed methods and Section VIII concludes this article.

II. Related Work

A. Traditional Spectrum Sensing Methods

In the context of traditional spectrum sensing methods, the construction of the test statistic emerges as the most influential factor on detection performance and represents a key distinguishing factor among various methods. For instance, if we have perfect prior knowledge regarding the PU's signals, the utilization of MFD can be employed to improve the detection performance. Zhang et al. [24] studied a MFD-based spectrum sensing and considered the scenario where the transmit power level of the PU is not constant. Brito et al. [25] developed general MFD to a hybrid spectrum sensing method. It adopted double MFD and executed different schemes with a false alarm probability of 0.5 as the boundary. If we only have access to partial information about the PU's signals but are aware of the statistical properties exhibiting temporal periodicity, the

application of CFD becomes a viable approach. Shen and Alsusa [26] designed a CFD method which can circumvent the computational burden associated with the 4th-order cumulant. To solve the multiple lags for a fixed cycle frequency of traditional CFD, Shen and Alsusa [27] proposed a new test statistics jointly utilizing cycle frequencies and lags. Different from MFD and CFD, ED obtains spectrum sensing results by calculating the energy of the received signals without the need for prior knowledge about the signals. Digham et al. [28] used the energy detector based on sampling theory for the unknown transmit signals. Further considering the probability of signal occurrence at different sampling points, Ma and Li [29] proposed a probability-based ED method. Yang et al. [30] considered a heterogeneous CR network and explored a normalized ED method based on the principle of log-likelihood ratio test according to the different sensing reliability of SUs. In addition, the covariance matrix (CM) is regarded as a versatile detection statistic that encompasses some discriminative characteristics for spectrum sensing. Especially, Zeng and Liang [17] designed the test statistic based on the absolute value of the covariance, while Liu et al. [31] employed the maximum eigenvalue of CM. On the other hand, Ali et al. [32] utilized the maximum-minimum eigenvalue approach. In order to improve the reliability of detection in complex channel environment, multiantenna reception is used for CR systems which can increase receiving gain and overcome channel fading. Wang et al. [33] proposed a sensing algorithm based on multiple high-order cumulants for the CR-IoT with multiple sensing antennas. It can allay the adverse effects of noise uncertainty and balance computation and detection performance effectively. Zhang et al. [34] studied the multiple antennas-based CR-IoT with additive Gaussian mixture noise model. They designed a kernelized test statistic by mapping the received signal matrix to high-dimensional feature space using the nonlinear Gaussian kernel function. Singh et al. [35] ameliorated the traditional energy detector for cooperative spectrum sensing of multiple CR networks with multiple receiving antennas which can obtain stable detection results at the low-SNR conditions of less than 0 dB. In general, these traditional methods exhibit certain constraints and restrictions in their application.

B. DL-Based Spectrum Sensing Methods

The principal distinctions among diverse DL-based spectrum sensing methods primarily manifest in variations within neural network architectures and the modalities of input data. Liu et al. [22] proposed a spectrum sensing framework based on CNN. It explored a data-driven test statistic via using the CM of samples as the input of CNN. Zheng et al. [5] developed the CNN-based detector based the power spectrum of sensing data. Short-time Fourier transform was applied to the signal samples and a spectrogram-aware CNN algorithm was proposed to settle the spectrum sensing problem in [36]. Soni et al. [37] considered the spectrum data as time-series data and exploited a detection architecture based on long short-term memory (LSTM) network which can improve the sensing performance at low-SNR regions. Moreover, considering both spatial and temporal features of received signals, Xie et al. [38] proposed a CNN-LSTM detector. It obtained the series of energy-correlation features from the CMs via CNN which were sent to LSTM to learn the PU activity pattern. In [39], a graph learning-based method to capture the received signal strengths (RSSs) of different SUs. Exploiting the property of low rank, the matrix constructed with RSSs from SUs fits nicely into the graph learning paradigm for a good detection performance. For the CR assisted Internet of Vehicles, Ahmed et al. [40] developed a learning-based spectrum sensing model with residual learning equipped with atrous spatial pyramid pooling (ASPP) module. It learned different situations of PU activity in the network from spectrograms. Besides supervised learning, Xie et al. [41] developed an auto-encoder for spectrum sensing approach. In contrast to the aforementioned approaches, it does not require a large amount of labeled data for training while demonstrating detection performance comparable to supervised methods under Gaussian and Laplace noise. In multiantenna receiving scenarios, DL-based methods have also gained popularity in the spectrum sensing domain. Liu et al. [22] proposed a learning-based multiantenna spectrum sensing architecture using the CM of received signals as the input of CNN (CM-CNN). Similarly, Wang et al. [42] considered a CR system where a PU and SU are equipped with multiple receiving antennas and proposed a spectrum sensing method based on adversarial learning via extracting the high-dimensional features of CM. They designed three coupled neural networks to improve the adaptability of different conditions of different SNR. Keunhong and Yusung [23] devised a deep spectrum sensing method for multiple antennas (DS2MA) receiving, utilizing the auto-correlation and cross-correlation functions of received signals. However, among the existing methods, the use of raw IQ data as input for DL-based multiple-antenna spectrum sensing approaches and the exploration of deployment scenarios with varying numbers of antennas have not been thoroughly investigated. These motivate our study here.

III. Preliminary Basics

A. Signal Model

We consider a CR-IoT network composed of one PU with single transmitting antenna and several SUs with different number of receiving antennas. As shown in Fig. 1, the SU senses the presence of PU with M ($M \geq 1$) received signals. For every SU, the problem of spectrum sensing can be formulated as a binary hypothesis test

$$\begin{cases} \mathcal{H}_0 : \mathbf{r}(n) = \mathbf{w}(n) \\ \mathcal{H}_1 : \mathbf{r}(n) = \mathbf{h}(n)s(n) + \mathbf{w}(n) \end{cases} \tag{1}$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ denote the PU is absent and present, respectively, $\mathbf{r}(n)$ is the received signal at the SU, $s(n)$ is the transmitted signal of the PU, $\mathbf{h}(n)$ is propagation channel between PU and SU, $\mathbf{w}(n)$ is additive noise, $n = 0, 1, \ldots, N-1$, N is the number of received samples. These vectors can be detailed as following equations at time slot n:

$$\mathbf{r}(n) = [r_1(n), r_2(n), \ldots, r_M(n)]^T \in \mathbb{C}^{M\times 1} \tag{2}$$

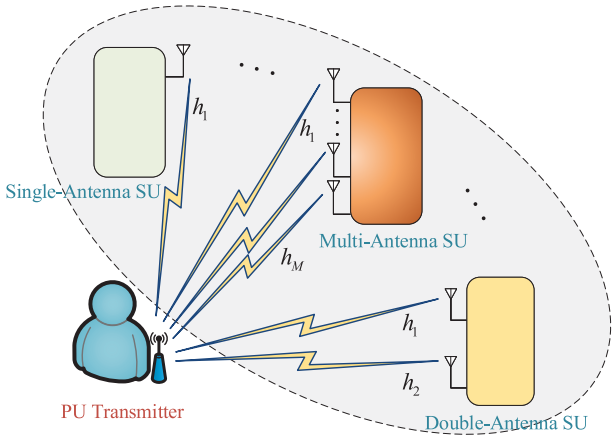


Fig. 1. Signal model.

$$\mathbf{h}(n) = [h_1(n), h_2(n), \ldots, h_M(n)]^T \in \mathbb{C}^{M\times 1} \tag{3}$$

$$\mathbf{w}(n) = [w_1(n), w_2(n), \ldots, w_M(n)]^T \in \mathbb{C}^{M\times 1} \tag{4}$$

where $r_i(n)$ denotes the received signal of the ith antenna, $h_i(n)$ is the channel response between the transmitter and the ith antenna, and $w_i(n)$ represents the noise of the ith antenna.

B. Noise Model

1) AWGN: The additive white Gaussian noise (AWGN) is a kind of ideal noise model which is easy to analyze. Most of existing literatures are researched on this model which is used to approximate the actual noise in a certain frequency band. Under AWGN, the noise is assumed to have a constant power spectral density over the entire channel bandwidth and the amplitude follows a Gaussian probability distribution:

$$p(\omega) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{\frac{(\omega-\mu)^2}{2\sigma^2}\right\} \tag{5}$$

where μ is the mean and σ^2 is the variance.

2) AGGN: However, it has been suggested that there were several communication systems followed noise distributions of generalized Gaussian distribution, such as underwater communications system, sensor network, local spectrum sensing application, and so on. Thus, we also consider additive generalized Gaussian noise (AGGN) model to analyze the performance of proposed algorithms. The probability function of AGGN can be expressed as

$$p(\omega) = \frac{\rho}{2\gamma\Gamma(1/\rho)} \exp\left\{-\left|\frac{\omega-\mu}{\gamma}\right|^{\rho}\right\} \tag{6}$$

where μ is the mean, γ is the inverse scale parameter, ρ is the shape parameter and $\Gamma(\cdot)$ represents the Gamma function.

C. Channel Model

1) Ideal Channel: We first consider the ideal condition where the channel response is a constant. In this case, when the noise model is AWGN, then the channel model could be called a AWGN channel.

2) Rayleigh Channel: In actual CR systems, the received signals of SU usually affected by multipath fading due to the obstacles between the PU and SU. We consider Rayleigh fading in this article. With the Rayleigh fading channel, the amplitude of the received signal is random and the envelope follows a Rayleigh distribution. The probability density function of Rayleigh distribution can be formulated as:

$$p(\omega) = \frac{\omega}{\sigma^2} \exp\left\{\frac{-\omega^2}{2\sigma^2}\right\}, \quad \omega > 0 \tag{7}$$

where σ^2 is the variance.

IV. Proposed MASSnet-B

Motivated by the recent surge in data-driven schemes utilizing DL in wireless communications networks and the IoT, we develop the MASSnet as a solution to address the challenge of multiple-antenna spectrum sensing. We first introduce the basic scheme, named MASSnet-B, which involves training the spectrum sensing models using the raw IQ data of received signals as input. During the training stage, binary labeled data indicating the presence or absence of PU are utilized as supervised information to train the sensing models. Then in the sensing stage, we employ the output features of MASnet-B as the detection statistic to complete the issue of binary hypothesis test. We refer to MASSnet-B as the basic scheme, as it still involves separate detectors for different devices in the context of CR-IoT with varying numbers of receiving antennas. It is suitable for the new members of CR-IoT that has the sufficient training data and computing resources to train the sensing models. It can generally achieve the satisfying detection performance due to the consistent reception configuration between training and inference stage. For the sake of simplicity in implementation, we design MASSnet-B in an ingenious manner that necessitates modification only to the first convolutional layer when accommodating different numbers of receiving antennas.

A. Net Structure of MASSnet-B

We choose CNN as the basic element for the MASSnet-B which has been successfully exploited to solve classification problems, such as image recognition, text classification, radio signal identification, and so on. The PU signal detection can be modeled as binary classification problem and CNN is one of the perfect approaches to settle it. MASSnet-B is designed based on artificial neural network of residual modules [43]. The residual structure can effectively extract richer features and avoid overfitting by using shortcut connection instead of direct connection between different layers of CNN. The fundamental components of MASSnet-B is shown in Table I, where "Conv" represents the convolutional layer, the fore "15 × 2M" is the kernel size and "/2 × 2M" is the stride of kernel movement, "MaxPool" and "GAvgPool" mean the maximum pool and global average pool operations, "Residual Block" denotes the basic residual module as shown in Fig. 2, "Dropout" denotes the dropout layer for avoiding overfitting and "Fc ×2" is the fully connection layer with 2 neurons. A

TABLE I
STRUCTURE OF MASSNET-B

Layer	Output dimensions
Input	$N \times 2M \times 1$
$15 \times 2M$, Conv, $/2 \times 2M$	$\frac{N}{2} \times 1 \times 16$
3×1, MaxPool, $/2 \times 1$	$\frac{N}{4} \times 1 \times 16$
Residual Block1(16)	$\frac{N}{4} \times 1 \times 16$
Residual Block2(32)	$\frac{N}{8} \times 1 \times 32$
Residual Block2(64)	$\frac{N}{16} \times 1 \times 64$
Residual Block2(128)	$\frac{N}{32} \times 1 \times 128$
GAvgPool	$1 \times 1 \times 128$
Dropout(0.5)	$1 \times 1 \times 128$
Fc $\times 2$	$1 \times 1 \times 2$

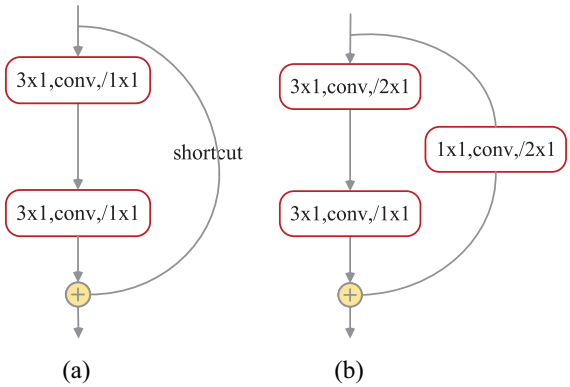


Fig. 2. Structure of residual blocks. (a) Residual block1. (b) Residual block2.

more detailed description of these modules of MASSnet is as follows.

1) Input Layer: To ensure the integrity of the information, we use the original IQ data of received radio signal of every antenna at the receiver as the input of MASSnet-B. Considering a CR system with M receiving antennas, the observation vector can be expressed as a complex matrix with a dimension of $N \times M$. Without loss of generality, we extract the I-channel and Q-channel sequence and form a matrix of real numbers, denoted as

$$\mathbf{x}^M = \begin{bmatrix} I_1(0) & I_1(1) & \cdots & I_1(N-1) \\ Q_1(0) & Q_1(1) & \cdots & Q_1(N-1) \\ I_2(0) & I_2(1) & \cdots & I_2(N-1) \\ Q_2(0) & Q_2(1) & \cdots & Q_2(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ I_M(0) & I_M(1) & \cdots & I_M(N-1) \\ Q_M(0) & Q_M(1) & \cdots & Q_M(N-1) \end{bmatrix} \tag{8}$$

where $\mathbf{x}^M$ is the input matrix, $\mathbf{x}^M \in \mathbb{R}^{N \times 2M \times 1}$. Thus, the input of MASSnet-B depends on the length of the received signals and the number of receiving antennas.

2) Convolutional Layer: The convolutional layer applies sliding convolutional filter equipped with weight $\mathbf{W}$ and bias b to the input $\mathbf{S}$. The convolutional operation computes the dot product of weight and the input by moving the filter vertically and horizontally, and then adds the bias. Assume that the input matrix $\mathbf{S} \in \mathbb{R}^{J \times K \times D}$ is a 3-D (3-D) tensor and we obtain the output feature tensor $\mathbf{Y} \in \mathbb{R}^{J' \times K' \times P}$ after convolved by the moving filter. Specifically, the process of feature mapping is formulated as

$$\mathbf{Y}_p = \sum_{d=1}^{D} \mathbf{W}_{p,d} \otimes \mathbf{S}_d + b_p, p \in P. \tag{9}$$

Note that we add padding operation with convolutional layer to ensure the size of the output is the same as the input if the stride equals to 1. A striding step size greater than 1 can reduce the size of output features. Especially, we design the stride of convolutional layer after the input layer of MASSnet-B to $2 \times 2M$ to ensure its output size be $(N/2) \times 1$. It is through this method of setting the stride of the first convolutional layer according to the number of receiving antennas to adapt to different CR systems with minimal modification.

In order to enhance the representation and computation capability of MASSnet-B, we apply rectified linear unit (ReLU) as nonlinear activation function to the output of convolutional layer, ReLU($\mathbf{Y}$), which is expressed as

$$\text{ReLU}(\mathbf{Y}) = \max\{0, \mathbf{Y}\}. \tag{10}$$

3) Pooling Layer: The pooling layer performs down-sampling by dividing the input into rectangular pooling regions where the maximum or average values are computed when max pooling layer or average pooling layer is adopted. Assume that the pooling region is defined as $R_{j,k}$, the result can be obtained as follows after the max-pooling operation:

$$y_{j,k} = \max_{i \in R_{j,k}} x_i \tag{11}$$

where x_i represents activity values of neurons in the covered area of $R_{j,k}$. Similarly, we have the average-pooling operation

$$y_{j,k} = \frac{1}{|R_{j,k}|} \sum_{i \in R_{j,k}} x_i. \tag{12}$$

The global average pooling performs computing a global mean through a sufficiently large pooling region.

4) Residual Block: As shown in Fig. 2, residual blocks improve the efficiency of information propagation in CNN via adding shortcut connections to nonlinear convolutional layers. The output of the residual block can be obtained by adding the results of identity mapping (or only perform dimension up or down) and residual mapping

$$z(y) = g(y) + \mathcal{F}(y) \tag{13}$$

where $g(\cdot)$ denotes the identity mapping implemented with some layer skips as shown in Fig. 2(a). If the $\mathcal{F}(\cdot)$ contains downsampling or upsampling operation via a stride not equal to one, it usually performs a convolutional operation with 1×1 kernel size to change the dimension of feature map, as illustrated in Fig. 2(b). This transformation of the objective

function learned by neural network from $z(y)$ to $z(y)-g(y)$ has been confirmed that can obtain a better training result [43].

5) Dropout Layer: The dropout layer helps prevent the MASSnet-B from overfitting via setting input elements to zero with a given probability which is designed to 0.5 in this article. Note that this operation only works in the training phase and the output of this layer is equal to its input at inference stage.

6) Fully Connected Layer: Since the problem of spectrum sensing has a binary solution, we design a fully connected layer with 2 neurons as the output of MASSnet and obtain the classification feature corresponding to the class labels (corresponding to hypothesis $\mathcal{H}_1$ and $\mathcal{H}_0$, respectively), i.e.,

$$h(z) = \begin{bmatrix} h_{|\mathcal{H}_1}(z) \\ h_{|\mathcal{H}_0}(z) \end{bmatrix} \tag{14}$$

where h represents the fully connected layer and $h_{|\mathcal{H}_i}(z)$ is the feature value of hypothesis $\mathcal{H}_i$, $i = 0, 1$.

B. Training Method of MASSnet-B

We use batch normalization (BN) technique [44] between convolutional layer and nonlinearity to speed up the training process. The BN normalizes the activations of each channel of mini-batch convolutional layers' output $\mathbf{z}$ by subtracting the mean μ and dividing by the standard deviation σ. A learnable offset β and scale factor γ are used to shift normalized parameters

$$\mathbf{z}_{\text{norm}} = \frac{\mathbf{z} - \mu}{\sqrt{\sigma^2 + \varepsilon}}\gamma + \beta \tag{15}$$

where ε is a constant for ensuring numerical stability and avoiding division by zero.

Once the output vector of the last fully connected layer is obtained, we use Softmax as the activation function for the binary classification problem of spectrum sensing

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{c=1}^{C} e^{z_c}} \tag{16}$$

where z_i denotes the output of the ith neuron of the fully connected layer of MASSnet-B and C is the number of categories which is designed to 2 in this article.

In addition, we train the MASSnet via stochastic gradient with momentum (SGDM) algorithm by minimizing the cross-entropy error loss as

$$\begin{aligned} \mathcal{L}\left(\mathbf{y}, f_B\left(\mathbf{x}_{\text{train}}^M; \theta_B\right)\right) &= -\mathbf{y}^\top \log \text{Softmax}\left(f_B\left(\mathbf{x}_{\text{train}}^M; \theta_B\right)\right) \\ &= -\sum_{c=1}^{C} y_c \log \text{Softmax}\left(f_{B,c}\left(\mathbf{x}_{\text{train}}^M; \theta_B\right)\right) \end{aligned} \tag{17}$$

$$\theta_{l+1} = \theta_l - \alpha \nabla_{\theta_l} \mathcal{L}(\theta_l) + \zeta(\theta_l - \theta_{l-1}) \tag{18}$$

where $f_B(\cdot;\cdot)$ is a parameterized function with respect to MASSnet-B, $\mathbf{x}_{\text{train}}^M \in \mathbb{R}^{N\times 2M\times 1}$ is one of the training samples, θ_B is the learnable parameters of MASSnet-B containing weights and biases, $\mathbf{y}$ is the label of $\mathbf{x}_{\text{train}}^M$ which is usually represented as a one-hot vector, α denotes the learning rate, l means the index of updating iteration, ∇ represents the gradient operation and ζ is a step-size hyperparameter which determines the contribution of previous gradient step to the current iteration. The detailed training algorithm for MASSnet-B is shown in Algorithm 1.

Algorithm 1: Training Process for MASSnet-B

Input : Training dataset $\mathcal{D}_{train} = \{\mathbf{x}_{train}^M, \mathbf{y}_{train}\}$, learning rate α, step-size hyper-parameter ζ, max number of training epochs $\mathcal{I}$;
Output: Trained MASSnet-B model $f_B(\theta_B)$;

```
Create a MASSnet-B model with random initial parameter;
for i = 1 to I do
    for the whole dataset D_train do
        Randomly choose a bath samples from D_train without repetition;
        Obtain the output of MASSnet-B model and calculate classification loss according to (17);
        Update parameters of the MASSnet-B model f_B(θ_B) according to (18).
    end
end
```

C. Spectrum Sensing With MASSnet-B

After the training stage of MASSnet-B, we use the trained model to sense the existence of PU with the unlabeled data. The test samples are sent to the trained model with the same form of training data and we can obtain the feature vector as follows:

$$f_B\left(\mathbf{x}_{\text{test}}^M; \theta_B\right) = \begin{bmatrix} f_{B|\mathcal{H}_1}\left(\mathbf{x}_{\text{test}}^M; \theta_B\right) \\ f_{B|\mathcal{H}_0}\left(\mathbf{x}_{\text{test}}^M; \theta_B\right) \end{bmatrix} \tag{19}$$

where $\mathbf{x}_{\text{test}}^M \in \mathbb{R}^{N\times 2M\times 1}$ is one of the test samples and $f_B(\cdot;\theta_B)$ the parameterized function with respect to the trained MASSnet-B model with parameters θ_B. We further achieve the detection results based on the output features. Generally speaking, when the PU signal is present, $f_{B|\mathcal{H}_1}(\mathbf{x}_{\text{test}}^M; \theta_B)$ tends to be large and $\text{Softmax}(f_{B|\mathcal{H}_1}(\mathbf{x}_{\text{test}}^M; \theta_B))$ will approach 1. When the PU is absent, $f_{B|\mathcal{H}_1}(\mathbf{x}_{\text{test}}^M; \theta_B)$ tends to be small and $\text{Softmax}(f_{B|\mathcal{H}_1}(\mathbf{x}_{\text{test}}^M; \theta_B))$ will approach 0. So we can use $f_{B|\mathcal{H}_1}(\mathbf{x}_{\text{test}}^M; \theta_B)$ as the detection statistic and adopt the following detection rule:

$$\begin{cases} \mathcal{H}_0 : f_{B|\mathcal{H}_1}\left(\mathbf{x}_{\text{test}}^M; \theta_B\right) < \lambda_B \\ \mathcal{H}_1 : f_{B|\mathcal{H}_1}\left(\mathbf{x}_{\text{test}}^M; \theta_B\right) \geq \lambda_B \end{cases} \tag{20}$$

where λ_B is the detection threshold.

As for the binary hypothesis testing problem, the probability of false alarm (P_f) and the probability of detection (P_d) are generally used to evaluate the performance of spectrum sensing methods. P_f represents the probability that SU detects the existence of PU (hypothesis $\mathcal{H}_1$) when the PU is absent (hypothesis $\mathcal{H}_0$), denoted as

$$P_f = P(\mathcal{H}_1|\mathcal{H}_0). \tag{21}$$

P_d represents the probability that SU can correctly judge the existence of PU (hypothesis $\mathcal{H}_1$) when the PU is really present

Algorithm 2: Spectrum Sensing Process for MASSnet-B

Input : Testing dataset $\mathcal{D}_{test} = \{\mathbf{x}_{test}^{M}\}$, pure noise samples, trained MASSnet-B model $f_B(\theta_B)$, the probability of false alarm P_f;
Output: Spectrum sensing results;
Obtain the noise feature vectors of trained MASSnet-B model based on pure noise samples according to (19);
Compute the detection threshold based on noise feature vectors and P_f [22];
for *the whole dataset* $\mathcal{D}_{test}$ **do**
  Randomly choose a bath samples from $\mathcal{D}_{test}$ without repetition;
  Obtain the output features of bath data based on MASSnet-B model according to (19);
  Compare the output features with the detection threshold to obtain the spectrum sensing results according to (20).
end

(hypothesis $\mathcal{H}_1$), denoted as

$$P_d = P(\mathcal{H}_1|\mathcal{H}_1). \tag{22}$$

Once the probability of false alarm P_f is set, the detection threshold λ_B is determined. We can set the threshold according to the desired probability of false alarm P_f [22]. We provide the pseudo-code of spectrum sensing process for MASSnet-B in Algorithm 2.

V. Proposed MASSnet-F

While proposed MASSnet-B can be directly adapted to different devices in CR-IoT with a same number of sensing antennas, it requires retraining of sensing models when the number of sensing antennas changes. Without loss of generality, massive connections lead IoT networks to be composed of different devices with varying numbers of receiving antennas. Since obtaining labeled training data is laborious and the training process demands a substantial amount of computational resources and execution time, it may be impractical to train separate spectrum sensing models for each device in practical implementations. A more sensible approach is to develop efficient algorithm that facilitate the reuse of sensing models across different devices. Taking into account the ease of implementing a single-antenna spectrum sensing model, we devise MASSnet-F which can extend the capability of single-antenna model to accommodate an arbitrary number of antennas through feature fusion from the received signals across diverse antennas. To be more specific, MASSnet-F only needs to be trained once with labeled IQ data of signals acquired by a single antenna. During the sensing stage, we input the IQ data from different receiving antennas separately into the trained model and calculate the average of the obtained feature matrices as the test statistic. Based on this statistic, we compute the detection threshold and ultimately complete spectrum sensing.

A. Net Structure and Training of MASSnet-F

The net structure of MASSnet-F can be considered as a special case of MASSnet-B with a single receiving antenna. Specifically, we build the MASSnet-F model according to Table I via setting M to 1. Hence, the input layer dimension is $N \times 2 \times 1$ and the kernel size of first convolutional layer is 15×2 where its stride equals 2×2. The input matrix $\mathbf{x}^1 \in \mathbb{R}^{N\times 2\times 1}$ can be represented as

$$\mathbf{x}^1 = \begin{bmatrix} I(0) & I(1) & \cdots & I(N-1) \\ Q(0) & Q(1) & \cdots & Q(N-1) \end{bmatrix}. \tag{23}$$

We collect signal samples of the CR system composed of only one antenna as the training data set. As mentioned in Section IV, we minimize the cross entropy between the true labels and predicted results of MASSnet-F model $\mathcal{L}(\mathbf{y}, f_F(\mathbf{x}^1_{\text{train}}; \theta_F))$ and update its parameters θ_F via SGDM technique until the training process is converged.

B. Spectrum Sensing With MASSnet-F

While the MASSnet-F model is trained with a single-antenna scenario, it can be used for multiantenna CR systems to sense the presence of PU signals. Considering a SU equipped with M receiving antennas, the assembled observation vector can be expressed as

$$\mathbf{x}^M_{\text{test}} = \begin{bmatrix} I_1(0) & I_1(1) & \cdots & I_1(N-1) \\ Q_1(0) & Q_1(1) & \cdots & Q_1(N-1) \\ I_2(0) & I_2(1) & \cdots & I_2(N-1) \\ Q_2(0) & Q_2(1) & \cdots & Q_2(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ I_M(0) & I_M(1) & \cdots & I_M(N-1) \\ Q_M(0) & Q_M(1) & \cdots & Q_M(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(1)}_{\text{test}} \\ \mathbf{x}^{(2)}_{\text{test}} \\ \vdots \\ \mathbf{x}^{(M)}_{\text{test}} \end{bmatrix}. \tag{24}$$

In this way, $\mathbf{x}^M_{\text{test}} \in \mathbb{R}^{N\times 2M\times 1}$ can be devided into M groups of input data $\mathbf{x}^{(m)}_{\text{test}} \in \mathbb{R}^{N\times 2\times 1}$, $m = 1, 2, \ldots, M$. These M matrices will be fed into the trained MASSnet-F model separately to obtain the classification feature vectors as

$$f_F\left(\mathbf{x}^{(m)}_{\text{test}}; \theta_F\right) = \begin{bmatrix} f_{F|\mathcal{H}_1}\left(\mathbf{x}^{(m)}_{\text{test}}; \theta_F\right) \\ f_{F|\mathcal{H}_0}\left(\mathbf{x}^{(m)}_{\text{test}}; \theta_F\right) \end{bmatrix}, \quad m = 1, 2, \ldots, M. \tag{25}$$

Then we merge the M feature vectors and calculate their mean as the detection statistics. The detection rule can be represented as

$$\begin{cases} \mathcal{H}_0 : \frac{1}{M}\sum_{m=1}^{M} f_{F|\mathcal{H}_1}\left(\mathbf{x}^{(m)}_{\text{test}}; \theta_F\right) < \lambda_F \\ \mathcal{H}_1 : \frac{1}{M}\sum_{m=1}^{M} f_{F|\mathcal{H}_1}\left(\mathbf{x}^{(m)}_{\text{test}}; \theta_F\right) \geq \lambda_F \end{cases} \tag{26}$$

where λ_F is the detection threshold which can be set according to the desired probability of false alarm P_f. Compared with MASSnet-B, MASSnet-F only needs to obtain the classification feature of each antenna and fuse the features instead of retraining the models to adapt to CR systems with different number of receiving antennas. The spectrum sensing algorithm for MASSnet-F can be seen in Algorithm 3.

Algorithm 3: Spectrum Sensing Process for MASSnet-F

Input : Testing dataset $\mathcal{D}_{test} = \{\mathbf{x}_{test}^{M}\}$, pure noise samples, trained MASSnet-F model $f_F(\theta_F)$, the probability of false alarm P_f;
Output: Spectrum sensing results;
Obtain M feature vectors of trained MASSnet-F model based on pure noise samples according to (25);
Merge M feature vectors and calculate the mean as the ultimate noise feature vector;
Compute the detection threshold based on noise feature vector and P_f [22];
for *the whole dataset* $\mathcal{D}_{test}$ **do**
  Randomly choose a bath samples from $\mathcal{D}_{test}$ without repetition;
  Obtain M output features for every sample of bath data based on MASSnet-F model according to (25);
  Obtain detection statistics by calculating the mean of M feature vectors and compare it with the detection threshold to obtain the spectrum sensing results according to (26).
end

TABLE II
STRUCTURE OF MASSNET-A

Layer	Output dimensions
Input	$N \times 2M \times 1$
31×4, Conv, /1×2	$N \times (M-1) \times 16$
$7 \times (M-1)$, AvgPool, /$1 \times (M-1)$	$N \times 1 \times 16$
15×1, Conv, /2×1	$\frac{N}{2} \times 1 \times 16$
3×1, MaxPool, /2×1	$\frac{N}{4} \times 1 \times 16$
Residual Block1(16)	$\frac{N}{4} \times 1 \times 16$
Residual Block2(32)	$\frac{N}{8} \times 1 \times 32$
Residual Block2(64)	$\frac{N}{16} \times 1 \times 64$
Residual Block2(128)	$\frac{N}{32} \times 1 \times 128$
GAvgPool	$1 \times 1 \times 128$
Dropout(0.5)	$1 \times 1 \times 128$
Fc $\times 2$	$1 \times 1 \times 2$

VI. PROPOSED MASSNET-A

MASSnet-F, with its easily implementable characteristics, allows for effortless reusability of the spectrum sensing model across different devices by the feature fusion technique. However, obtaining separate feature matrices tends to overlook the correlation between different receiving antennas and the detection performance is suboptimal in this sense. In this section, we propose MASSnet-A method which needs to be trained only once and can be adaptive to arbitrary number of antennas by taking into account the correlation among signals of different receiving antennas to improve the detection performance. Without loss of generality, a multiantenna receiving SU with a small number of receiving antennas is easier to implement and, in this scenario, allows for more cost-effective and efficient training of the sensing models. We devise MASSnet-A to train the sensing model using the raw IQ data of more than 2 receiving antennas. During the sensing stage, we can effortlessly adapt MASSnet-A for deployment in other IoT devices with varying numbers of receiving antennas by making minor modifications and recalculating the detection thresholds. Because we design an adaptive feature extraction module to capture the interantenna information, incorporating a variable, parameter-free pooling layer to preserve the fixed dimension of output feature matrices when handling different IQ data from various CR-IoT devices with various numbers of receiving antennas.

A. Net Structure and Training of MASSnet-A

Based on MASSnet-B, we add a front module used to extract the features of different antennas and limit the output feature size for forward propagation at the following layers. The front module includes a convolutional layer and an average pooling layer. As shown in Table II, we set the kernel size of first convolutional layer to 31×4 in order to learn the correlation between adjacent 2 antennas. A stride of 1×2 is used to ensure that the covered region of convolution occupies IQ signals of exactly 2 antennas. To make the MASSnet-A work for CR systems with arbitrary number of antennas, we design an average pooling layer with kernel size of $7 \times (M-1)$ and stride of $1 \times (M-1)$ after the first convolutional layer. Once performing down-sampling of this layer, a feature map of size $N \times 1 \times 16$ will be obtained no matter what the value of M is. Note that the BN and ReLU operations are also placed between the convolutional layer and the pooling layer. Then we adjust the next convolutional layer's kernel size to 15×1 and stride to 2×1. The subsequent layers has the same structure as MASSnet-B.

We note that in training MASSnet-B, the same number of receiving antennas is used to acquire the training samples and the testing samples. In training MASSnet-F, only a single antenna is needed to acquire the signals to construct the training samples. Different from these two methods, in training MASSnet-A, the configuration of the required number of antennas is more flexible. Generally speaking, we can use $M_1 \geq 2$ antennas to construct the training samples of MASSnet-A. As mentioned earlier, we minimize the cross entropy between the true labels and predicted results of MASSnet-A model $\mathcal{L}(\mathbf{y}, f_A(\mathbf{x}_{\text{train}}^{M_1}; \theta_A))$ and update its parameters θ_A via SGDM technique until the training process is converged. The detailed training method can be performed with reference to MASSnet-B, which will not be repeated here for simplicity.

B. Spectrum Sensing With MASSnet-A

With a front module of convertible parameters, MASSnet-A can be adopted to new CR systems in the inference stage which are equipped with different number of sensing antennas

Algorithm 4: Spectrum Sensing Process for MASSnet-A

Input : Testing dataset $\mathcal{D}_{test} = \{\mathbf{x}_{test}^{M}\}$, pure noise samples, trained MASSnet-A model $f_A(\theta_A)$, the number of antennas in training dataset M_1, the probability of false alarm P_f;

Output: Spectrum sensing results;

if $M \neq M_1$ **then**
  Modify the kernel size of the average pooling layer in the front module of MASSnet-A model to $7 \times (M-1)$;
end
Obtain the noise feature vectors of trained MASSnet-A model based on pure noise samples according to (27);
Compute the detection threshold based on noise feature vectors and P_f [22];
for *the whole dataset* $\mathcal{D}_{test}$ **do**
  Randomly choose a bath samples from $\mathcal{D}_{test}$ without repetition;
  Obtain the output features of bath data based on MASSnet-A model according to (27);
  Compare the output features with the detection threshold to obtain the spectrum sensing results according to (28).
end

as offline training stage. The detailed spectrum sensing process is shown in Algorithm 4. In online detection stage, we first verify if the number of receiving antennas for the data to be sensed corresponds to the configuration of training data set. If they do not align, it is necessary to adjust the parameters of MASSnet-A according to the current number of antennas. After that, we obtain the sensing data $\mathbf{x}_{\text{test}}^{M}$ which is represented as the observation matrix in (8). With the trained model, the feature vector can be obtained as

$$f_A\left(\mathbf{x}_{\text{test}}^{M};\theta_A\right) = \begin{bmatrix} f_{A|\mathcal{H}_1}\left(\mathbf{x}_{\text{test}}^{M};\theta_A\right) \\ f_{A|\mathcal{H}_0}\left(\mathbf{x}_{\text{test}}^{M};\theta_A\right) \end{bmatrix}. \tag{27}$$

Similarly, we can use $f_{A|\mathcal{H}_1}(\mathbf{x}_{\text{test}}^{M};\theta_A)$ as the detection statistic and adopt the following decision rule:

$$\begin{cases} \mathcal{H}_0 : f_{A|\mathcal{H}_1}\left(\mathbf{x}_{\text{test}}^{M};\theta_A\right) < \lambda_A \\ \mathcal{H}_1 : f_{A|\mathcal{H}_1}\left(\mathbf{x}_{\text{test}}^{M};\theta_A\right) \geq \lambda_A \end{cases} \tag{28}$$

where λ_A is the detection threshold which can be set according to the desired probability of false alarm P_f. It can be seen that with MASSnet-A, the inference only needs to be carried out once for each testing sample, which can save computational complexity compared with MASSnet-F.

VII. Performance Evaluation

A. Parameter Settings

We generate the simulation data for evaluating the performance of our proposed MASSnets. QPSK with the initial phase of $\pi/4$ is chosen as the modulation type of the PU as the basic setting. 16QAM is also used to test the performance of the methods in the adaptation to new modulations. We apply the raised cosine FIR pulse-shaping filter to the modulated signals with an oversampling factor of 8. Each signal (under $\mathcal{H}_1$) contains 1024 symbols which means that the length of each signal is 8192. The arrival time offset of every receiving antenna is randomly chosen in the range of -5 to $+5$ samples. The SNR ranges from -50 dB to -5 dB with an interval of 1 dB. There are 500 samples at each SNR for training and 250 samples at each SNR for testing. Without loss of generality, the same amount of noise data (under $\mathcal{H}_0$) is generated. All of the following experiments are accomplished on dell laptop with Intel(R) Core(TM) i9-9900K CPU @ 3.6 GHz and NVIDIA GeForce RTX 2080 GPU. The hyperparameter setting in training the MASSnets is shown in Table III. Note that for MASSnet-B, the model needs to be retrained for each number of receiving antennas, while for MASSnet-F and MASSnet-A, the model is trained only once. Specifically, MASSnet-F is trained with a single antenna and MASSnet-A is trained with the configuration of 4 receiving antennas.

TABLE III
Training Options

Parameters	Values
Optimizer	SGDM
Max Number of Epochs	45
Mini-Batch Size	64
Initial Learning Rate	0.01
Learning Rate Drop Period	7
Learning Rate Drop Factor	0.2

B. Comparison Methods

In order to evaluate the effectiveness of our proposed algorithms, we compare the detection performance of three schemes of MASSnet with two existing DL-based multiantenna spectrum sensing methods. Note that both comparison methods have not consider the reuse of sensing models across different CR-IoT devices. When the number of sensing antennas changes, they require executing the entire training operation to obtain the new sensing models. In other words, the training and inference stages remain consistent when using the two methods.

1) CM-CNN: CM-CNN is a popular spectrum sensing framework based on CNN, which has been confirmed to outperform the conventional methods [22]. It also formulates the binary hypothesis test as a binary classification problem and trains the CNN detector using labeled data of the PU is absent or present. Utilizing the powerful ability of CNN in extracting features of matrix-shaped data, CM-CNN calculates the CM of received signal as the input of detector

$$\mathbf{R}_{\mathbf{r}}(N) = \frac{1}{N}\sum_{n=0}^{N-1}\mathbf{r}(n)\mathbf{r}^{T}(n) \tag{29}$$

where $\mathbf{R}_{\mathbf{r}}(N)$ is the CM of N observation vectors with a dimension of $M \times M$. Then the real and imaginary parts of complex

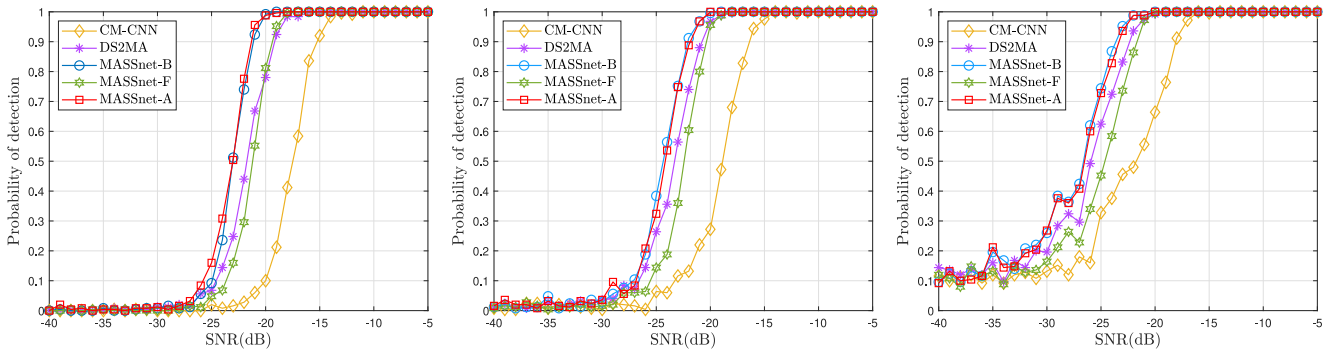


Fig. 3. Detection performance versus SNR of different methods under various probabilities of false alarm: $M = 4$. (a) $P_f = 0.001$. (b) $P_f = 0.01$. (c) $P_f = 0.1$.

matrix $\mathbf{R_r}(N)$ are further expanded to form the input matrices of dimension $M \times M \times 2$ for the CNN.

Once obtained the well-trained detector, CM-CNN can determine the detection threshold according the output of CNN and the probability of false alarm. Finally, spectrum sensing is accomplished based on the detection threshold.

2) DS2MA: DS2MA is a newly proposed method designed to tackle the issue of multiple antennas spectrum sensing with DL [23]. It merges the auto-correlation and cross-correlation of received signals as the input matrices for CNN-based detector. The auto-correlation function matrix can be expressed as

$$\phi_{i,\tau} = \frac{1}{N}\sum_{n=0}^{N-1} r_i(n) r_i^*(n-\tau) \tag{30}$$

where $\tau = 0, 1, 2, \ldots, L-1$, $i = 0, 1, 2, \ldots, M-1$, $r_i^*(n)$ is the conjugate of $r_i(n)$, L is the symbol duration, M is the number of antennas and $\phi_{i,\tau}$ is the element of the auto-correlation matrix. The cross-correlation function matrix can be expressed as

$$\psi_{c,\tau} = \frac{1}{N}\sum_{n=0}^{N-1} r_i(n) r_{i'}^*(n-\tau) \tag{31}$$

where $\psi_{c,\tau}$ is the element of the cross-correlation matrix, $i' = 0, 1, 2, \ldots, M-1$, $i \neq i'$, c is an index representing different cross-correlations and $c = 0, 1, 2, \ldots, C_M^2$.

The input of detector comprises a combination of the auto-correlation matrix and the cross-correlation matrix, possessing a dimension of $(M + C_M^2) \times L \times 2$. The detector is also trained through supervised learning with binary labels and produces the detection results based on a specified false alarm probability.

C. Simulation Results

1) Performance Comparison: We first compare the detection performance of proposed three spectrum sensing algorithms with the CM-CNN and DS2MA methods. For the sake of illustration, we show the probability of detection versus SNR ranged from -40 dB to -5 dB under different false alarm probability P_f in Fig. 3. Without loss of generality, both the training data set and the test data set are generated under the simulation condition of $M = 4$ receiving antennas. Simulation results demonstrate that the detection performance of the proposed MASSnet-B and MASSnet-A is superior to that of CM-CNN and DS2MA detection methods under various P_f, owing to the richer information obtained from IQ data compared to the spatial-temporal correlation of the received signals. Even in cases where single antenna fusion is employed, MASSnet-F shows superior detection performance compared to CM-CNN and achieves similar results to DS2MA when the SNR exceeds -20 dB. In the scenario of same number of antennas at training and inference stage, the performance of MASSnet-A is close to that of MASSnet-B and better than that of MASSnet-F. Among the three proposed methods, in terms of probability of detection, the worst performed MASSnet-F still has performance gain than CM-CNN. For instance, when SNR $= -18$ dB, MASSnet-F can achieve a probability of detection close to 100% with a probability of false alarm of 0.01 while the DS2MA has the comparable results and the CM-CNN has a probability of detection of nearly 68% with the same probability of false alarm.

2) Adaptation to Different Number of Antennas: We now evaluate the performance of CM-CNN, DS2MA and proposed MASSnets with different number of receiving antennas. The false alarm probability is kept 0.01 in these experiments. Results are shown in Fig. 4. Note that MASSnet-B, CM-CNN and DS2MA need to be retrained for different number of antennas while MASSnet-A and MASSnet-F only recalculate the detection thresholds. MASSnet-A was trained under the case of 4-antenna reception and evaluated in other scenarios of different amount of receiving antennas. It is obvious that all the detection algorithms can obtain performance gain with the increase of amount of receiving antennas. As shown in Fig. 4(a), in the case that the receiving antennas are not too many, MASSnet-A can achieve very close performance to MASSnet-B where there is an SNR gain of about 3 dB when the number of antennas is doubled. As the number of antennas increases, the performance of MASSnet-A and MASSnet-B consistently outperform CM-CNN and DS2MA. Specifically, in the case of 8 receiving antennas, they can

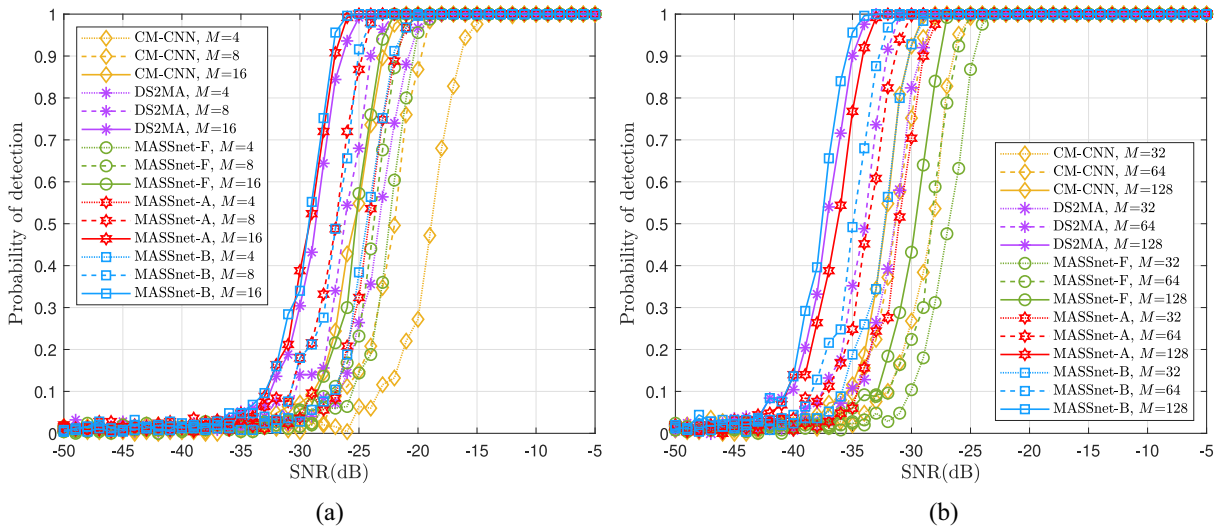


Fig. 4. Probability of detection of different algorithms with various numbers of antennas: $P_f = 0.01$. (a) $M = 4, 8, 16$. (b) $M = 32, 64, 128$.

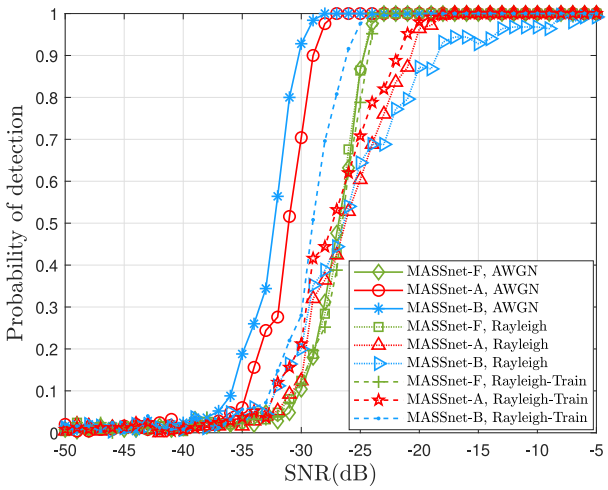


Fig. 5. Detection performance versus SNR of proposed MASSnet under new channel model: $P_f = 0.01$, $M = 32$.

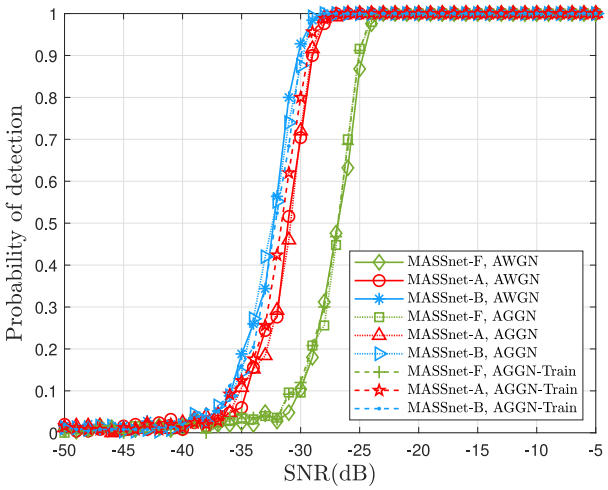


Fig. 6. Detection performance versus SNR of proposed MASSnet under new noise model: $P_f = 0.01$, $M = 32$.

surpass the CM-CNN with 16 receiving antennas. As for MASSnet-F, although it can be extended from only single-antenna scenario to arbitrary number of antennas without retraining, it obtains the least gain from the increasing number of antennas. While the detection performance of MASSnet-F is not as good as DS2MA, it still surpasses CM-CNN. When the number of antennas is 16, the detection results acquired by MASSnet-F are close to those of CM-CNN which needs to be retrained. When the scale of antenna further expanded, MASSnet-B still achieves about 3 dB gain from antenna doubling as shown in Fig. 4(b). MASSnet-A does not perform as well as MASSnet-B, but performs significantly better than CM-CNN. For instance, the performance of MASSnet-A with 32 antennas is similar as that of CM-CNN with 64 antenna and the performance of MASSnet-B with 32 antennas is similar as that of CM-CNN with 128 antennas. We should note that both DS2MA and MASSnet-B involve retraining the sensing model for specific number of receiving antennas, but the proposed MASSnet-B achieves a higher detection probability than DS2MA in low-SNR situations. These results validate the superiority of our proposed methods.

3) Performance Under Rayleigh Channel: The above experiments show the advantages of our proposed three schemes of MASSnet which are based on the assumption that the propagation channel is AWGN channel. To validate the generalizability of MASSnet, we study the detection performance under different wireless communication channels. Considering the mobility of IoT devices, we show the simulation results under Rayleigh fading channel in Fig. 5. We assume the Rayleigh fading is frequency-flat and the maximum Doppler shift is set to 30 Hz [45]. Note that "Rayleigh-Train" means we train the MASSnet models based on the simulation samples under the corresponding Rayleigh channel and "Rayleigh" means we perform spectrum sensing for Rayleigh fading channel using the trained AWGN-based models. Furthermore, the retraining operations of MASSnet-F and MASSnet-A are only performed on the basic models.

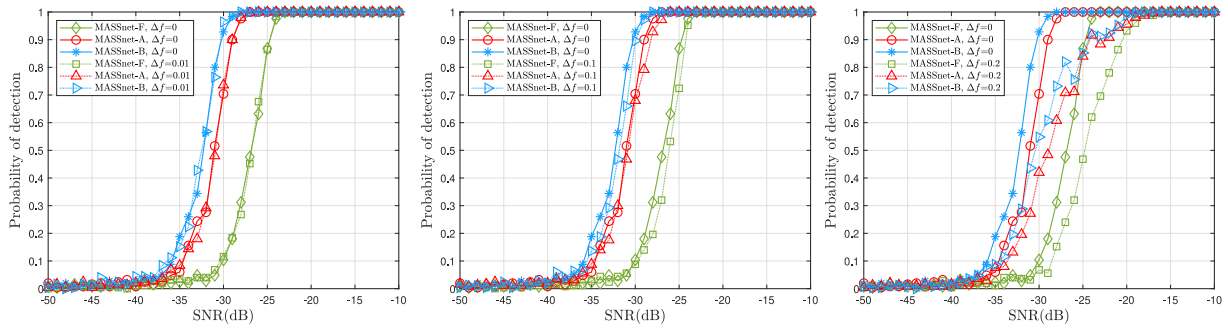


Fig. 7. Detection performance of proposed MASSnets under various frequency offsets Δf: $M = 32$, $P_f = 0.01$. (a) $\Delta f = 0.01$. (b) $\Delta f = 0.1$. (c) $\Delta f = 0.2$.

Specifically, for MASSnet-F we only train the models once with the Rayleigh-based samples of single-antenna condition. For MASSnet-A, we train the model only once using 4-antenna data set in Rayleigh channel. Fig. 5 demonstrates the good adaptability of MASSnet-F and MASSnet-A under Rayleigh fading channel whether executing retraining or not. Though AWGN-trained MASSnet-B model has a greater loss than other schemes of MASSnet under Rayleigh fading channel, MASSnet-B can obtain more noticeable performance increment with retraining of the data under the channel to be sensed. Under low SNR, it achieves a better performance than the other schemes. Overall, the proposed MASSnets have satisfactory performance under different channel models. While the training data of Rayleigh channel is available, they can achieve a detection probability over 90% with a false alarm probability of 0.01 at SNR $= -21.5$ dB.

4) Performance Under AGGN: Most traditional spectrum sensing methods consider the condition of AWGN, but the actual CR systems may not necessarily conform to the ideal characteristics of AWGN. Thus, we evaluate the MASSnets under other noise model. Fig. 6 shows the probability of detection of MASSnets with AGGN. For AGGN, the arbitrary location parameter is 0, the inverse scale parameter is 1 and the shape parameter is 1.5. As shown in Fig. 6, MASSnets adapt to AGGN well with or without retraining. The detection performance is similar under following three cases: 1) detecting under AWGN with trained AWGN-based models (denoted as "AWGN"); 2) detecting under AGGN with trained AWGN-based models (denoted as "AGGN"); and 3) detecting under AGGN with trained AGGN-models (denoted as "AGGN-Train"). It should be noted that the detection threshold should be redefined under AGGN with the trained AWGN-based models.

5) Effect of Frequency Deviation on Detection Performance: In the actual CR system, the signal received by the SU tends to has a frequency deviation due to oscillator mismatch or Doppler shift between the transmitter and receiver. This is an important factor affecting the performance of spectrum sensing algorithms. We study the detection performance of MASSnets under different carrier frequency deviations. As shown in Fig. 7, when the normalized carrier offset Δf (relative to the symbol rate) is set to 0.01, the performance of MASSnets is almost the same as the condition without frequency deviation. If the Δf increases to 0.1, the probability of detection has a slight decrease and the degree of decline is similar for the three schemes of MASSnets. Fig. 7(c) further shows that an increased frequency deviation causes a greater performance degradation. In conclusion, MASSnets have some robustness against frequency deviation although the training data set is composed of samples without frequency deviation.

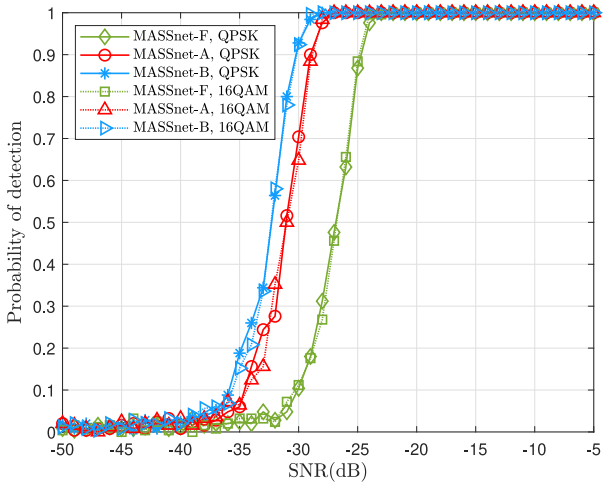


Fig. 8. Performance of detecting new signals: $P_f = 0.01$, $M = 32$.

6) Adaptation to New Signals: Finally, we study the performance of MASSnets with the new signals with untrained modulation type. Note that all the models of proposed schemes of MASSnets are trained with QPSK data. Without loss of generality, the signal samples of 16QAM are used to verify the ability of MASSnets, where the probability of false alarm is kept 0.01 and the number of receiving antennas of the CR system is chosen 32. Simulation results are shown in Fig. 8. It demonstrates that MASSnets can adapt to the new signals of 16QAM modulation with almost the same performance as the case of QPSK. This validates that the trained MASSnets can be used to detect new signals that they have never seen before.

VIII. Conclusion

In this article, we have investigated a multiple-antenna spectrum sensing architecture named MASSnet for CR-IoT, which

relies on raw IQ data of received signals. We formulate the problem of PU signal detection as a binary hypothesis test and trains the MASSnet as a binary classification model. We can delimit the detection threshold flexibly according the different false alarm probability via putting the noise data to the well-trained model. The proposed three schemes can be adapted to different devices with different numbers of receiving antennas in CR-IoT. Two of them even allow for the reusability of sensing models when the number of antennas changes. It is more feasible in practical implementations. Experiment results have shown that the MASSnet based on raw IQ data is superior to the covariance or correlation matrices-aware methods at low-SNR situations. Results validate the robustness and scalability of proposed three schemes under different conditions of different number of sensing antennas, different signal modulations, different wireless channel models and noise models. In our future work, we will further explore the collaborative sensing of different SUs and conduct over-the-air experiments to verify the performance of MASSnet in real-world situations.

REFERENCES

[1] S. Surekha and M. Z. U. Rahman, "Spectrum sensing and allocation strategy for IoT devices using continuous-time Markov chain-based game theory model," *IEEE Sensors Lett.*, vol. 6, no. 4, pp. 1–4, Apr. 2022.

[2] L. Niu and F. Li, "Cooperative spectrum sensing for Internet of Things using modeling of power-spectral-density estimation errors," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7802–7814, May 2022.

[3] Y.-C. Liang, Y. Zeng, E. C. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1326–1337, Apr. 2008.

[4] D. Wang, W. Zhang, B. Song, X. Du, and M. Guizani, "Market-based model in CR-IoT: A Q-probabilistic multi-agent reinforcement learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 179–188, Mar. 2020.

[5] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Commun.*, vol. 17, no. 2, pp. 138–148, Feb. 2020.

[6] T. Xiong, Y.-D. Yao, Y. Ren, and Z. Li, "Multiband spectrum sensing in cognitive radio networks with secondary user hardware limitation: Random and adaptive spectrum sensing strategies," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3018–3029, May 2018.

[7] J. Park, P. Pawelczak, and D. Cabric, "Performance of joint spectrum sensing and MAC algorithms for multichannel opportunistic spectrum access Ad Hoc networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 7, pp. 1011–1027, Jul. 2011.

[8] X. Liu, C. Sun, M. Zhou, C. Wu, B. Peng, and P. Li, "Reinforcement learning-based multislot double-threshold spectrum sensing with Bayesian fusion for industrial big spectrum data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3391–3400, May 2021.

[9] J. Jiang, H. Sun, D. Baglee, and H. V. Poor, "Achieving autonomous compressive spectrum sensing for cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1281–1291, Mar. 2016.

[10] S. D. Arthur Nkalango, H. Zhao, Y. Song, and T. Zhang, "Energy efficiency under double deck relay assistance on cluster cooperative spectrum sensing in hybrid spectrum sharing," *IEEE Access*, vol. 8, pp. 41298–41308, 2020.

[11] A. I. Perez-Neira, M. A. Lagunas, M. A. Rojas, and P. Stoica, "Correlation matching approach for spectrum sensing in open spectrum communications," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4823–4836, Dec. 2009.

[12] Y. Yilmaz, Z. Guo, and X. Wang, "Sequential joint spectrum sensing and channel estimation for dynamic spectrum access," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 2000–2012, Nov. 2014.

[13] B. Li, S. Li, A. Nallanathan, Y. Nan, C. Zhao, and Z. Zhou, "Deep sensing for next-generation dynamic spectrum sharing: More than detecting the occupancy state of primary spectrum," *IEEE Trans. Commun.*, vol. 63, no. 7, pp. 2442–2457, Jul. 2015.

[14] E. Conte, M. Lops, and G. Ricci, "Adaptive matched filter detection in spherically invariant noise," *IEEE Signal Process. Lett.*, vol. 3, no. 8, pp. 248–250, Aug. 1996.

[15] P. Sepidband and K. Entesari, "A CMOS spectrum sensor based on quasi-cyclostationary feature detection for cognitive radios," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 12, pp. 4098–4109, Dec. 2015.

[16] M. S. Oude Alink, A. B. J. Kokkeler, E. A. M. Klumperink, G. J. M. Smit, and B. Nauta, "Spectrum sensing with high sensitivity and interferer robustness using cross-correlation energy detection," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 3, no. 4, pp. 566–575, Dec. 2013.

[17] Y. Zeng and Y.-C. Liang, "Spectrum-sensing algorithms for cognitive radio based on statistical covariances," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1804–1815, May 2009.

[18] I. Mehmood et al., "Efficient image recognition and retrieval on IoT-assisted energy-constrained platforms from big data repositories," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9246–9255, Dec. 2019.

[19] S. Aunkaew and S. Tantiviwat, "The application of IoT-controlled electrical device by using Smartphone speech recognition among people with mobility impairment," in *Proc. Int. Conf. Electr., Comput. Energy Techn. (ICECET)*, 2021, pp. 1–6.

[20] F. Wei et al., "Detection of direct sequence spread spectrum signals based on deep learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 3, pp. 1399–1410, Sep. 2022.

[21] S. Zheng et al., "Toward next-generation signal intelligence: A hybrid knowledge and data-driven deep learning framework for radio signal classification," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 3, pp. 564–579, Jun. 2023.

[22] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, "Deep CM-CNN for spectrum sensing in cognitive radio," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2306–2321, Oct. 2019.

[23] C. Keunhong and K. Yusung, "DS2MA: A deep learning-based spectrum sensing scheme for a multi-antenna receiver," *IEEE Wireless Commun. Let.*, vol. 12, no. 6, pp. 952–956, Jun. 2023.

[24] X. Zhang, F. Gao, R. Chai, and T. Jiang, "Matched filter based spectrum sensing when primary user has multiple power levels," *China Commun.*, vol. 12, no. 2, pp. 21–31, Feb. 2015.

[25] A. Brito, P. Sebastião, and F. J. Velez, "Hybrid matched filter detection spectrum sensing," *IEEE Access*, vol. 9, pp. 165504–165516, 2021.

[26] J.-C. Shen and E. Alsusa, "An efficient multiple lags selection method for Cyclostationary feature based spectrum-sensing," *IEEE Signal Process. Lett.*, vol. 20, no. 2, pp. 133–136, Feb. 2013.

[27] J.-C. Shen and E. Alsusa, "Joint cycle frequencies and lags Utilization in Cyclostationary feature spectrum sensing," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5337–5346, Nov. 2013.

[28] F. F. Digham, M.-S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels," *IEEE Trans. Commun.*, vol. 55, no. 1, pp. 21–24, Jan. 2007.

[29] J. Ma and Y. G. Li, "A probability-based spectrum sensing scheme for cognitive radio," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2008, pp. 3416–3420.

[30] G. Yang et al., "Cooperative spectrum sensing in heterogeneous cognitive radio networks based on normalized energy detection," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1452–1463, Mar. 2016.

[31] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, "Maximum eigenvalue-based goodness-of-fit detection for spectrum sensing in cognitive radio," *IEEE Trans. Veh. Technol*, vol. 68, no. 8, pp. 7747–7760, Aug. 2019.

[32] S. S. Ali, W. Zhao, M. Jin, and S.-J. Yoo, "Enhanced maximum-minimum eigenvalue based spectrum sensing," in *Proc. Informat. Commun. Technol. Conv. (ICTC)*, 2019, pp. 708–713.

[33] D. Wang, P. Qi, Q. Fu, N. Zhang, and Z. Li, "Multiple high-order Cumulants-based spectrum sensing in full-duplex-enabled cognitive IoT networks," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9330–9343, Jun. 2021.

[34] J. Zhang, L. Liu, M. Liu, Y. Yi, Q. Yang, and F. Gong, "MIMO spectrum sensing for cognitive radio-based Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8874–8885, Sep. 2020.

[35] A. Singh, M. R. Bhatnagar, and R. K. Mallik, "Cooperative spectrum sensing in multiple antenna based cognitive radio network using an improved energy detector," *IEEE Commun. Lett.*, vol. 16, no. 1, pp. 64–67, Jan. 2012.

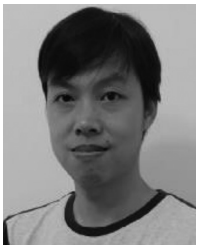
[36] L. Cai, K. Cao, Y. Wu, and Y. Zhou, "Spectrum sensing based on spectrogram-aware CNN for cognitive radio network," *IEEE Wireless Commun. Lett.*, vol. 11, no. 10, pp. 2135–2139, Oct. 2022.

[37] B. Soni, D. K. Patel, and M. López-Benítez, "Long short-term memory based spectrum sensing scheme for cognitive radio using primary activity statistics," *IEEE Access*, vol. 8, pp. 97437–97451, 2020.

[38] J. Xie, J. Fang, C. Liu, and X. Li, "Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2196–2200, Oct. 2020.
[39] T. Jiang, M. Jin, Q. Guo, Y. Tian, and J. Liu, "Graph learning-based cooperative spectrum sensing in cognitive radio networks," *IEEE Wireless Commun. Lett.*, vol. 12, no. 1, pp. 138–142, Jan. 2023.
[40] R. Ahmed, Y. Chen, B. Hassan, L. Du, T. Hassan, and J. Dias, "Hybrid machine learning-based spectrum sensing and allocation with adaptive congestion-aware modeling in CR-assisted IoV networks," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 25100–25116, Dec. 2022.
[41] J. Xie, J. Fang, C. Liu, and L. Yang, "Unsupervised deep spectrum sensing: A variational auto-encoder based approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5307–5319, May 2020.
[42] C. Wang, Y. Xu, Z. Chen, J. Tian, P. Cheng, and M. Li, "Adversarial learning-based spectrum sensing in cognitive radio," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 498–502, Mar. 2022.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.
[45] S. Zheng, S. Chen, and X. Yang, "DeepReceiver: A deep learning-based intelligent receiver for wireless communications in the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 5–20, Mar. 2021.

Luxin Zhang received the M.S. degree in control science and engineering from Zhejiang University of Technology, Hangzhou, China, in 2021.

He is currently an Assistant Engineer with the National Key Laboratory of Electromagentic Space Security, Jiaxing, China. His research interests include cognitive radio–radio signal processing and learning-based radio signal recognition.



Shilian Zheng received the B.S. degree in telecommunication engineering and the M.S. degree in signal and information processing from Hangzhou Dianzi University, Hangzhou, China, in 2005 and 2008, respectively, and the Ph.D. degree in communication and information system from Xidian University, Xi'an, China, in 2014.

He is currently a Researcher with the National Key Laboratory of Electromagentic Space Security, Jiaxing, China. His research interests include cognitive radio, deep learning-based radio signal processing, and electromagnetic space security.

Kunfeng Qiu was born in Shaoxing, Zhejiang, China, in 1997. He received the M.S. degree in control engineering from Zhejiang University of Technology, Hangzhou, China, in 2022.

He is currently an Assistant Engineer with the National Key Laboratory of Electromagentic Space Security, Jiaxing, China. His research interests include radio signal processing and radio signal recognition based on deep learning.

Caiyi Lou received the bachelor's degree in communications engineering and the master's degree in communications and electronic systems from Xidian University, Xi'an, China, in 1991 and 1998, respectively.

He is currently a Researcher with the National Key Laboratory of Electromagentic Space Security, Jiaxing, China. He coauthored the first software radio book in China [along with X. Yang, C. Lou, and J. Xu, *Software Radio Principles and Applications*, (Publishing House of Electronics Industry, 2001) (in Chinese)]. His research interests include software-defined radio and communication signal processing.

Xiaoniu Yang received the B.S. degree in communications engineering and the M.S. degree in communications and electronic systems from Xidian University, Xi'an, China, in 1982 and 1988, respectively.

He is currently a Chief Scientist with the National Key Laboratory of Electromagentic Space Security, Jiaxing, China. He published the first software radio book in China [along with X. Yang, C. Lou, and J. Xu, *Software Radio Principles and Applications* (Publishing House of Electronics Industry, 2001) (in Chinese)]. He holds more than 40 patents. His current research interests are software-defined satellite, big data for radio signals, and deep learning-based signal processing.

Dr. Yang is also an Academician of Chinese Academy of Engineering and a Fellow of the Chinese Institute of Electronics.