Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Spectrum sensing method based on a multi-scale feature fusion network*

Honghui XIANG, Kejun LEI†‡, Kaiqing ZHOU, Wenjing TUO, Hongbin LIU

School of Communication and Electronic Engineering, Jishou University, Jishou 416000, China

†E-mail: leikejun-123@163.com

Received May 8, 2025; Revision accepted Oct. 14, 2025; Crosschecked Nov. 18, 2025; Published online Dec. 13, 2025

Abstract: Signal-to-noise ratio (SNR) fluctuations significantly affect spectrum sensing performance in wireless communications. Traditional convolutional neural network (CNN) exhibits limited feature extraction capabilities and inefficient feature utilization at low SNR levels, leading to suboptimal spectrum sensing performance. This paper proposes a spectrum sensing method based on a multi-scale feature fusion network (MSFFNet) to address this issue. First, the proposed method employs a multi-scale feature extraction block (MSFEB) to capture multi-scale information from the input data comprehensively. Next, an adaptive feature screening strategy (AFSS) highlights key features while suppressing redundant information. Finally, a multi-level feature fusion mechanism (MLFFM) optimizes and integrates features across scales and levels, enhancing spectrum sensing performance. Simulation results demonstrate that compared to other methods, the proposed approach achieves superior performance in low-SNR communication scenarios. At an SNR of −14 dB, the detection probability P_{d} reaches 0.936, while the false alarm probability P_{fa} is only 0.1. Furthermore, this paper constructs a multi-level mixed-SNR dataset to simulate real communication environments and enhance the robustness of spectrum sensing.

Key words: Cognitive radios; Spectrum sensing; Deep learning; Multi-scale feature fusion
https://doi.org/10.1631/FITEE.2500297 **CLC number:** TN929.5

1 Introduction

With the rapid advancement of wireless communication technology, the demand for spectrum resources continues to increase. However, fixed spectrum allocation results in low utilization and exacerbates the spectrum shortage issue. Spectrum sensing, a core cognitive radio (CR) technology, dynamically monitors the wireless environment and detects idle frequency bands, enhancing spectrum utilization and mitigating resource shortages. However, improving the accuracy and robustness of spectrum sensing in low signal-to-noise ratio (SNR) and complex wireless environments remains a critical challenge.

Currently, spectrum sensing research is primarily divided into two categories: model-driven traditional methods and data-driven intelligent methods. Traditional spectrum sensing techniques include energy detection (Urkowitz, 1967), eigenvalue-based detection (Zeng and Liang, 2009), and cyclostationary detection (Sutton et al., 2008). These methods are susceptible to noise, leading to a significant decline in detection performance under low-SNR conditions. Moreover, they depend on prior information and struggle to adapt to dynamic wireless environments.

Compared with traditional methods, deep learning (DL) can autonomously extract time–frequency

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 62066016 and 62161012), the Natural Science Foundation of Hunan Province of China (No. 2024JJ7395), the Scientific Research Project of Education Department of Hunan Province of China (No. 22B0549), and the Scientific Research Project of Jishou University (No. Jdy24049)

ORCID: Honghui XIANG, https://orcid.org/0009-0001-6538-0652; Kejun LEI, https://orcid.org/0000-0001-7881-1768

features from signals through large-scale data training, enhance adaptability to complex wireless environments, and improve detection performance under low-SNR conditions. Liu et al. (2019) proposed a general spectrum sensing framework based on a deep neural network (DNN), which ensures the optimality of the test statistic. They further designed a covariance matrix-aware convolutional neural network (CM-CNN) for spectrum sensing. Gai et al. (2021) addressed the gradient vanishing problem in deep networks using skip connections and developed a deep convolutional neural network (DCNN) to enhance spectrum sensing accuracy. To further improve feature utilization, Gai et al. (2022) proposed a spectrum sensing method based on residual dense networks, which enhances feature fusion through dense connectivity. Geng et al. (2022) proposed a spectrum sensing method based on feature extraction and a convolutional neural network (CNN). The method constructs a feature matrix composed of energy, power spectrum, and cyclostationary features, which is then fed into the CNN for classification. Cai et al. (2022) proposed a spectrum sensing method based on a spectrogram-aware CNN, which effectively extracts time-frequency features by using the spectrum graph obtained from the short-time Fourier transform of the signal as input. Janu et al. (2023) proposed a collaborative spectrum sensing method using a hierarchical long short-term memory network (LSTM)-based architecture, which integrates CNN and LSTM to extract spatial and temporal features from secondary user (SU) sensing data. Li SP et al. (2024) proposed a multi-antenna spectrum sensing method that integrates hierarchical fuzzy dispersion entropy (HFDE) with a CNN-gated recurrent unit (GRU) architecture. HFDE is used to extract frequency-domain features, while the CNN-GRU network effectively captures both spatial and temporal dependencies. Pan et al. (2025) proposed a DL-based spectrum prediction framework, DeepSPred, which employs an encoder to extract spatiotemporal spectrum features and designs a 3D Swin Transformer block-based network (3D-SwinSTB) and a 3D Swin Transformer with linear predictor (3D-SwinLinear) for spectrum monitoring and dynamic access tasks, respectively. Saraswathi and Logashanmugam (2024) proposed a spectrum sensing method that integrates the chicken swarm algorithm with a deep belief network. The deep belief network is employed to learn the dynamic features of the spectrum in the offline phase, while the chicken swarm algorithm is used in the online phase to optimize nonlinear thresholds for efficient sensing. Gao et al. (2024) proposed an intelligent spectrum sensing method based on a GAN-GRU-YOLO framework, which integrates wavelet transform with a dual-branch GRU-YOLO network to extract time-frequency features of signals for efficient spectrum sensing.

Although DL models can autonomously extract features, their ability to perceive critical information remains limited under complex channel conditions. By focusing on discriminative features, the attention mechanism can enhance the model's efficiency in learning meaningful information. Hou et al. (2021) introduced a coordinate attention (CA) mechanism that enhances feature representation by embedding coordinate information, achieving notable results in various computer vision tasks. Compared to classic attention mechanisms, such as squeeze-and-excitation network (SE-Net) (Hu et al., 2018), CA offers lower computational complexity. Li JC et al. (2018) introduced a multi-scale residual network for image super-resolution. The network adaptively extracts multi-scale features via residual blocks and employs skip connections for feature fusion, effectively leveraging local multi-scale information. Zhang et al. (2024) proposed a spectrum Transformer model leveraging a multi-head self-attention mechanism. The model captures both inner-band features and inter-band correlations, effectively addressing the limitations of traditional CNNs in modeling long-range spectral dependencies. Peng et al. (2024) proposed a DL-based wideband spectrum signal detection network. The model integrates a residual dilated convolutional network with a horizontal shift attention mechanism, effectively enhancing signal detection performance in complex electromagnetic environments. Bai et al. (2025) proposed a collaborative spectrum sensing method that incorporates a channel attention mechanism. The method uses CNN to extract spatial features and enhances key information through the channel attention mechanism, while integrating an LSTM to capture the temporal characteristics of SUs, thereby significantly improving the sensing accuracy. Xu et al. (2025) proposed an underwater acoustic spectrum sensing algorithm that integrates federated learning with the

dark-CBAM-YOLOv5 (DCYOLO) model, incorporating the dark convolutional block attention module (dark-CBAM) feature extraction and a multi-feature fusion module, thus enhancing the sensing accuracy of underwater signals. Hong and Xu (2025) proposed a multi-scale time-correlation perception attention model. The model employs a parallel architecture to fully leverage correlations across multiple time scales and identify dependencies between them, thereby enhancing detection performance and noise robustness.

Despite the notable performance of DL-based spectrum sensing methods, several key challenges remain. (1) Most existing models employ single-scale convolutional structures, which are constrained by fixed receptive fields and struggle to capture multi-scale signal features. (2) Feature fusion commonly relies on simple splicing or concatenation operations, which lack guidance and often introduce redundant information, thereby reducing sensing efficiency and accuracy. (3) Existing works introduce attention mechanisms to focus on key features, but these are typically confined to local enhancement within the channel or spatial dimensions and fail to capture global feature interactions across scales and layers.

To address these challenges, this paper proposes a spectrum sensing method based on a multi-scale feature fusion network (MSFFNet). The main contributions are as follows:

1. To introduce a novel MSFFNet for spectrum sensing. Compared to traditional single-scale CNNs, MSFFNet extracts local and global spectrum features in parallel using multi-scale feature extraction blocks (MSFEBs), thereby enhancing the model's ability to perceive multi-scale signal patterns.

2. To propose an adaptive feature screening strategy (AFSS) and a multi-level feature fusion mechanism (MLFFM) for enhanced feature representation. AFSS evaluates the contributions of features at different scales to the sensing task, enabling adaptive adjustment of multi-scale feature weights to emphasize key features and suppress redundant information. MLFFM combines skip connections and multi-dimensional attention to effectively fuse shallow detail features with deep semantic representations, achieving cross-scale information synergy and improving global feature representation.

3. To evaluate the performance of the MSFFNet under low-SNR conditions. This paper constructs a multi-level mixed-SNR dataset and a Rayleigh fading channel model to simulate complex communication scenarios, and employs multiple evaluation strategies to comprehensively analyze the performance of the sensing model. Experimental results demonstrate that MSFFNet achieves strong performance in terms of sensing accuracy, scalability, and robustness.

2 Spectrum sensing model

In a multi-antenna CR scenario, a CR terminal is equipped with M antennas to receive signals in the target frequency band. SUs employ the CR terminal to detect whether the primary users (PUs) occupy the target band and opportunistically use the idle band without interfering with the PUs' communication. Therefore, spectrum sensing can be modeled as a binary hypothesis testing problem:

$$\begin{cases} H_1 : \boldsymbol{r}(n) = \boldsymbol{x}(n) + \boldsymbol{w}(n), \\ H_0 : \boldsymbol{r}(n) = \boldsymbol{w}(n), \end{cases} \tag{1}$$

where H_1 indicates the presence of the PU signal along with channel noise in the band, while H_0 represents the presence of only noise, meaning that the PU is absent. $\boldsymbol{r}(n) = [r_1(n), r_2(n), \ldots, r_M(n)]^{\mathrm{T}}$ denotes the observation vector received by the CR terminal at time n on M antennas, where $n = 0, 1, \ldots, N-1$, N is the number of received samples, and $\boldsymbol{x}(n)$ represents the PU signal, which undergoes multi-path propagation and channel fading, with the power set to σ_x^2. For SUs, the prior information of the PU signal is unavailable. $\boldsymbol{w}(n)$ represents the additive white Gaussian noise (AWGN) in the target frequency band, with a mean of 0 and a variance of σ_w^2. The received signal matrix is given by

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_M \end{bmatrix} = \begin{bmatrix} r_1(0) & r_1(1) & \cdots & r_1(N-1) \\ r_2(0) & r_2(1) & \cdots & r_2(N-1) \\ \vdots & \vdots & & \vdots \\ r_M(0) & r_M(1) & \cdots & r_M(N-1) \end{bmatrix}_{M \times N}. \tag{2}$$

Spectrum sensing is performed by testing a binary hypothesis. The performance of the spectrum sensing method is evaluated using two key metrics: detection probability P_{d} and false alarm probability P_{fa}. P_{d} is the likelihood of correctly identifying the presence of the PU when it exists, whereas P_{fa} is the likelihood of incorrectly detecting the PU when it is

absent. The specific decision criterion is defined as

$$\begin{cases} P_{\mathrm{d}} = P(T \geq \gamma \mid H_1), \\ P_{\mathrm{fa}} = P(T \geq \gamma \mid H_0), \end{cases} \tag{3}$$

where T denotes the test statistic, and γ represents the decision threshold.

3 Spectrum sensing system based on MSFFNet

The MSFFNet-based spectrum sensing system integrates data preprocessing, offline training, and online detection to create an end-to-end intelligent sensing framework to identify the wireless spectrum status in real time and with high accuracy. The MSFFNet enhances the spectrum sensing method's adaptability and decision-making precision. The system's overall framework is illustrated in Fig. 1.

The following analysis will focus on the three core modules, i.e., data preprocessing, MSFFNet model design, and threshold decision mechanism, to detail how the system achieves spectrum sensing.

3.1 Data preprocessing

Based on the received signal matrix in Eq. (2), the sample covariance matrix is calculated as follows:

$$\boldsymbol{C}(N) = \frac{1}{N}\boldsymbol{R}\boldsymbol{R}^{\mathrm{H}} = \frac{1}{N}\sum_{n=0}^{N-1}\boldsymbol{r}(n)\boldsymbol{r}^{\mathrm{H}}(n). \tag{4}$$

Since the H_1 hypothesis includes both the PU signal and the noise signal, while the H_0 hypothesis includes only the noise signal, assuming statistical independence between the PU and the noise signals, Eq. (1) is transformed into Eq. (5):

$$\begin{cases} H_1 : \boldsymbol{C}(N) = \boldsymbol{C}_x(N) + \sigma_w^2\boldsymbol{I}_M, \\ H_0 : \boldsymbol{C}(N) = \sigma_w^2\boldsymbol{I}_M, \end{cases} \tag{5}$$

where $\boldsymbol{I}_M$ represents the M-dimensional identity matrix. For H_0, since the AWGN is independent and identically distributed (i.i.d.), there is no correlation between the noises. Hence, the sample covariance matrix contains only the diagonal elements representing the noise power. For H_1, the PU signal is present in addition to the noise signal. Due to the spatial correlation between antennas in a multi-antenna receiver system and temporal correlation in the PU signal, the diagonal elements of the sample covariance matrix represent the sum of the noise power and PU signal power (i.e., $\sigma_x^2 + \sigma_w^2$). The off-diagonal elements represent the quantification of the correlation between the multi-antenna received signals. The sample covariance matrices under both hypotheses are visualized in grayscale, as illustrated in Fig. 2.

Under high-SNR conditions (SNR = 5 dB), the two hypothesized covariance matrices exhibit noticeable differences in brightness in the off-diagonal region. Under low-SNR conditions (SNR = −19 dB), the grayscale images are nearly identical, suggesting that the signal is obscured by noise. This paper transforms the (32 × 32)-dimensional covariance matrix into a 128 × 128 resolution grayscale image called $\boldsymbol{C}$. The covariance matrix contains various signal characteristics, including signal power, eigenvalues, and correlation. High-resolution grayscale images accentuate subtle structural features in the

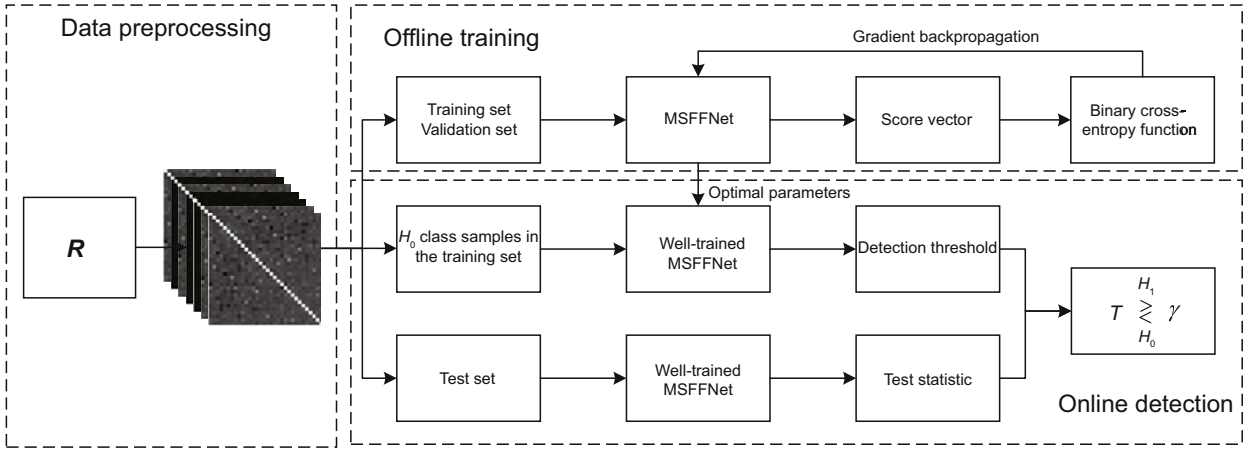


Fig. 1 Spectrum sensing system based on MSFFNet

data matrix while meeting the input size requirements of DNN. This provides precise and reliable features for MSFFNet, aiding subsequent feature extraction and pattern recognition.

3.2 MSFFNet model design

MSFFNet adopts a hierarchical architecture with three stages: feature extraction, feature fusion, and feature classification, as shown in Fig. 3. This hierarchical design aims to provide modularity by separating feature extraction, fusion, and classification, enabling easy adjustment and extension to diverse spectrum sensing scenarios.

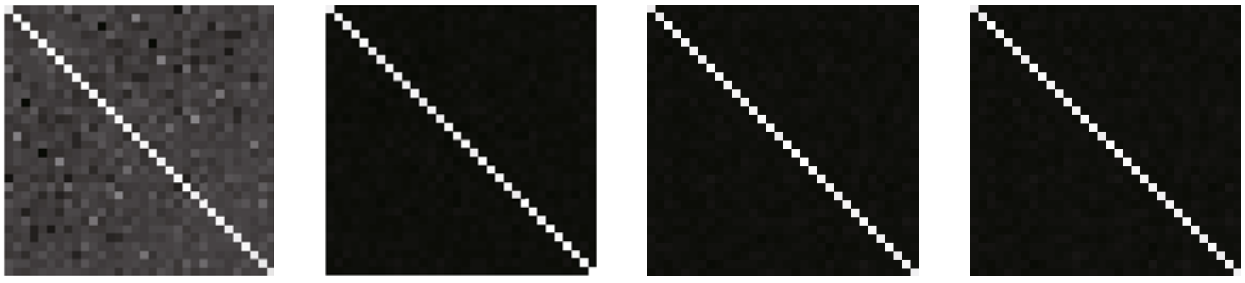
In the feature extraction stage, MSFEBs extract multi-scale features from the covariance matrix grayscale image, while AFSS selectively emphasizes key features and suppresses redundant ones. In the feature fusion stage, MLFFM combines multi-scale features from each layer using attention weights, reinforcing important features and reducing information loss. In the classification stage, the extracted features are integrated by a fully connected (FC) layer, and a softmax activation generates a score vector, whose elements represent the probability distribution of hypotheses H_0 and H_1. Finally, a threshold-based decision mechanism is applied to the score vector to complete the classification task. The detailed parameters of MSFFNet are listed in Table 1.

3.2.1 MSFEB

Convolution kernels with varying receptive fields capture features at different scales, enabling the model to focus on local and global information during feature extraction. This allows the model to

Table 1 Structural parameters of the MSFFNet

Layer	Output feature map size
Input	1×128×128
C0	32×64×64
MSFEB 1	64×32×32
MSFEB 2	128×16×16
MSFEB 3	256×8×8
MLFFM	480×8×8
FC1 + ReLU	30 720×512
FC2 + softmax	512×2
Output	1×2



(a) (b) (c) (d)

Fig. 2 Grayscale image of the sample covariance matrix: (a) H_1 when SNR = 5 dB; (b) H_0 when SNR = 5 dB; (c) H_1 when SNR = −19 dB; (d) H_0 when SNR = −19 dB

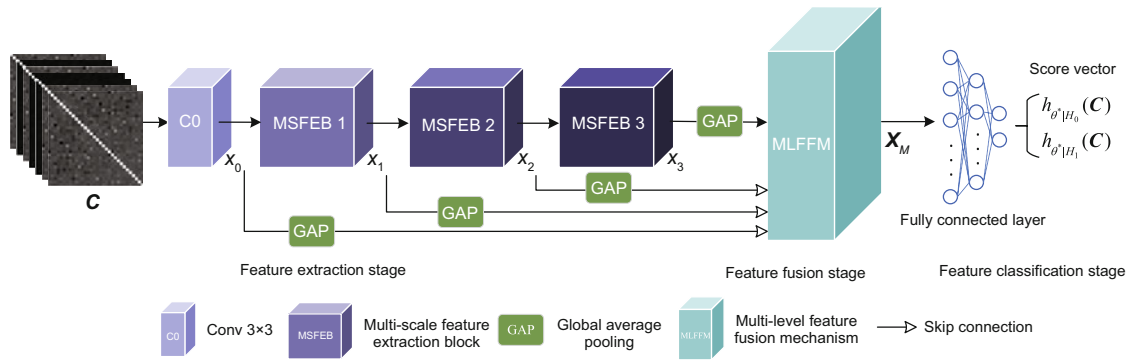


Fig. 3 MSFFNet model structure

learn the relationship between detailed features and macroscopic patterns, improving the recognition accuracy of complex spectral signals. Therefore, this paper designs an MSFEB, which includes multi-scale feature extraction (Fig. 4a) and an AFSS (Fig. 4b).

1. Multi-scale feature extraction

Unlike traditional single-scale CNNs, this paper's MSFEB captures multi-scale information using a dual-branch structure. Specifically, convolution kernels with receptive fields of 1×1, 3×3, and 7×7 are employed to build a multi-scale feature extraction and residual connection branches. In the multi-scale feature extraction branch, 3×3 convolution extracts local features, while 7×7 convolution captures global information. The first convolution layer downsamples the input feature map, reducing its size to half of the original one, while the number of channels remains unchanged. This reduces model parameters through feature compression and enables the extraction of more abstract features. Features of different scales are then concatenated along the channel dimension to facilitate interaction between them. The second convolution layer preserves the feature map size while doubling the number of channels. Doubling the channels enables the model to learn richer feature representations. The multi-scale features are then adaptively screened to remove redundancy and emphasize key features, improving the model's ability to learn important information. The third layer uses 1×1 convolution to fuse local and global features, enabling cross-channel information interaction and creating a more expressive feature representation. In the residual connection branch, a 1×1 convolution adjusts the feature dimension to supplement information and enhance model efficiency. Finally, the two branches are combined to produce the final feature output.

2. AFSS

To account for the varying contributions of different scale features to spectrum sensing tasks, this

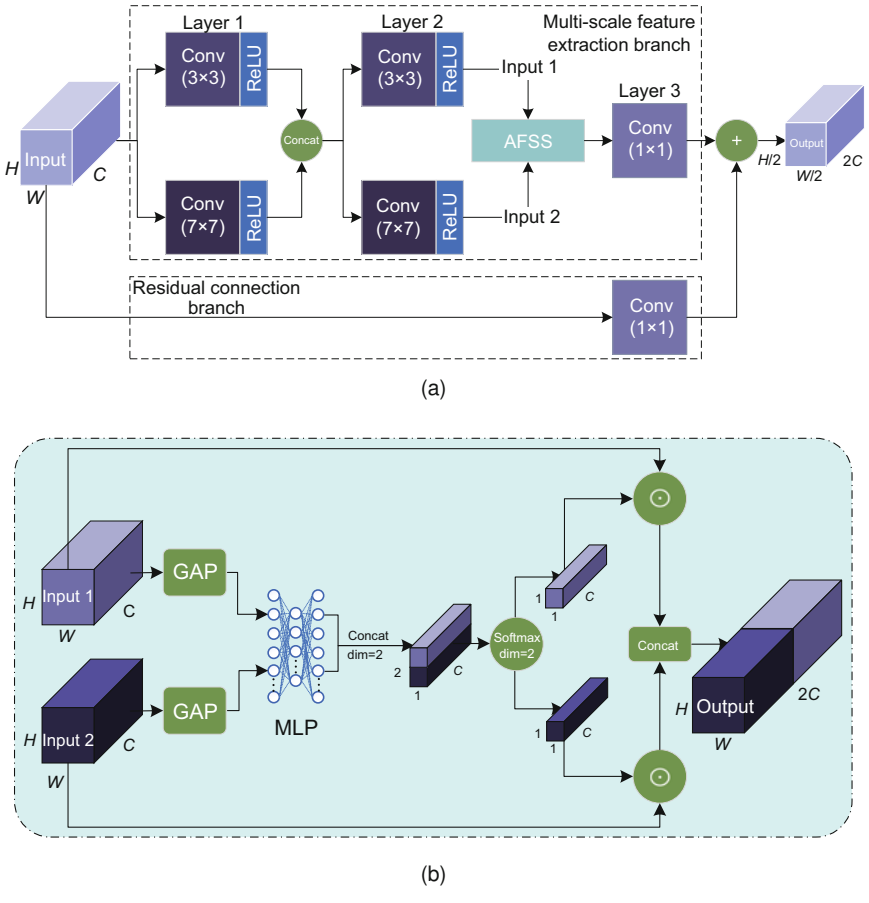


Fig. 4 Illustration of the multi-scale feature extraction (a) and AFSS in the MSFEB (b). ⊙: channel-wise multiplication

paper proposes an AFSS that emphasizes key features while suppressing redundant ones. This strategy screens feature maps extracted by 3×3 and 7×7 convolution kernels. First, the global average pooling (GAP) calculates the average channel weights of feature maps extracted by 3×3 and 7×7 convolution kernels. These weights are then fed into a multi-layer perceptron (MLP) built with 1×1 convolution to model the channel's feature relationships and perform nonlinear transformations. Next, two sets of weights are concatenated, and a softmax function is applied across the corresponding channels to obtain probabilistic feature interaction weights. Finally, the initial feature map is multiplied by its corresponding interaction weight for each channel, and the resulting weighted maps are concatenated along the channel dimension to screen and enhance multi-scale features.

3.2.2 MLFFM

As each layer of the MSFEB downsamples the feature map, the spatial representation capability of the model decreases with increasing network depth while its semantic representation ability improves. This may result in the loss of detailed features during transmission. MSFEBs at various levels capture features at different scales. Shallow modules focus on details and local structures, while deep modules capture complex semantic features. Given the strong performance of the attention mechanism in feature fusion, this paper proposes an MLFFM based on the CA. It enhances the network's feature representation by modeling multi-dimensional global information and effectively integrating shallow details with deep semantic features.

As shown in Fig. 5, the core mechanism of this module extracts global information in the horizontal, vertical, and spatial directions by performing adaptive average pooling along the width, height, and channel dimension of the feature map, generating three attention weights: width-wise, height-wise, and spatial attention. These weights then adjust the importance of the input feature map probabilistically, enabling the model to focus on salient areas. The module first resizes the features (x_0, x_1, x_2, x_3) extracted in the feature extraction stage (Fig. 3) using skip connections and GAP, and then concatenates them along the channel dimension. Next, GAP is applied across the width, height, and channel dimensions to capture global feature information in the horizontal, vertical, and spatial directions. A lightweight MLP compresses and reconstructs the features along the width and height dimensions, generating position weights through sigmoid activation and softmax normalization. Finally, the width-wise, height-wise, and spatial attention weights are applied to the input features element-wise, enabling the fusion and enhancement of multi-dimensional information.

This module effectively combines global spatial and channel information, enhancing the sensitivity of the model to feature distribution. Fusing local features at all levels prevents the loss of detail, fully uses detailed and semantic features, and learns more general representations to enhance the model generalization. Skip connections improve gradient flow, mitigate gradient vanishing and explosion issues in deep networks, and stabilize model training. The attention mechanism adaptively fuses features,

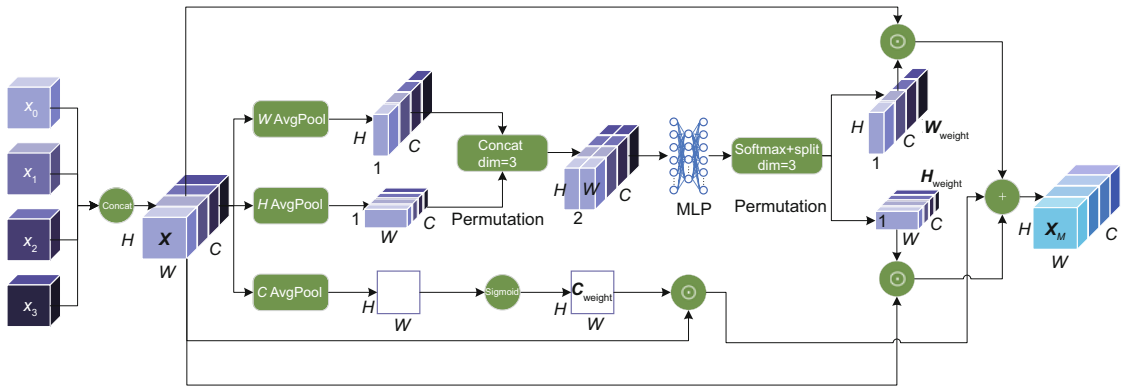


Fig. 5 Illustration of the MLFFM

emphasizes key features, and reduces reliance on irrelevant ones, improving training efficiency and convergence speed.

The theoretical analysis of the MLFFM is provided in Eqs. (6)–(14). Let $\boldsymbol{X} \in \mathbb{R}^{B\times C\times H\times W}$ be the input feature map, where B denotes the batch size, C denotes the number of channels, and H and W denote the height and width of the feature map, respectively. GAP is applied to dimensions W, H, and C, respectively.

$$\boldsymbol{X}_w(c,h)=\frac{1}{W}\sum_{w=1}^{W}\boldsymbol{X}(b,c,h,w),\ \boldsymbol{X}_w\in\mathbb{R}^{B\times C\times H\times 1}, \tag{6}$$

$$\boldsymbol{X}_h(c,w)=\frac{1}{H}\sum_{h=1}^{H}\boldsymbol{X}(b,c,h,w),\ \boldsymbol{X}_h\in\mathbb{R}^{B\times C\times 1\times W}, \tag{7}$$

$$\boldsymbol{X}_c(h,w)=\frac{1}{C}\sum_{c=1}^{C}\boldsymbol{X}(b,c,h,w),\ \boldsymbol{X}_c\in\mathbb{R}^{B\times 1\times H\times W}. \tag{8}$$

Apply the permutation operation to transpose $\boldsymbol{X}_h$, aligning its shape with that of $\boldsymbol{X}_w$. Concatenate $\boldsymbol{X}_w$ and $\boldsymbol{X}_h$ along the w dimension:

$$\boldsymbol{Y}=\mathrm{Concat}(\boldsymbol{X}_w,\boldsymbol{X}_h),\quad \boldsymbol{Y}\in\mathbb{R}^{B\times C\times H\times 2}. \tag{9}$$

An MLP with two 1×1 convolutional layers processes the concatenated feature $\boldsymbol{Y}$, where σ is the sigmoid activation function, $\boldsymbol{W}_1 \in \mathbb{R}^{C\times C/16}$ is the weight of the reduction of dimensions, and $\boldsymbol{W}_2 \in \mathbb{R}^{C/16\times C}$ is the weight of the increase of dimensions:

$$\boldsymbol{Z}=\sigma\left(\boldsymbol{W}_2\cdot\mathrm{ReLU}\left(\boldsymbol{W}_1\boldsymbol{Y}\right)\right),\quad \boldsymbol{Z}\in\mathbb{R}^{B\times C\times H\times 2}. \tag{10}$$

Apply the softmax operation to the last dimension of $\boldsymbol{Z}$, decompose the resulting weights into W and H direction components, and restore the $\boldsymbol{H}_{\mathrm{weight}}$ to the original dimension ($B\times C\times 1\times W$):

$$\mathrm{Softmax}(\boldsymbol{Z},\mathrm{dim}=3), \tag{11}$$

$$\boldsymbol{W}_{\mathrm{weight}},\boldsymbol{H}_{\mathrm{weight}}=\mathrm{Split}(\boldsymbol{Z},[1,1],\mathrm{dim}=3). \tag{12}$$

Generate spatial attention weights by applying the sigmoid activation function to $\boldsymbol{X}_c$:

$$\boldsymbol{C}_{\mathrm{weight}}=\sigma(\boldsymbol{X}_c). \tag{13}$$

The input feature map $\boldsymbol{X}$ is element-wise multiplied by $\boldsymbol{W}_{\mathrm{weight}}$, $\boldsymbol{H}_{\mathrm{weight}}$, and $\boldsymbol{C}_{\mathrm{weight}}$, and the sum is used to generate the output feature map $\boldsymbol{X}_M \in \mathbb{R}^{B\times C\times H\times W}$:

$$\boldsymbol{X}_M=\boldsymbol{X}\odot\boldsymbol{W}_{\mathrm{weight}}+\boldsymbol{X}\odot\boldsymbol{H}_{\mathrm{weight}}+\boldsymbol{X}\odot\boldsymbol{C}_{\mathrm{weight}}, \tag{14}$$

where $\odot$ denotes the element-wise multiplication.

3.3 Threshold decision mechanism

Setting an appropriate decision threshold effectively mitigates the impact of noise and uncertainty on decision outcomes. To simplify the spectrum sensing problem into a binary decision task while ensuring real-time processing, this paper presents a threshold decision mechanism based on CNN (Liu et al., 2019). The mechanism consists of three main steps:

Step 1: constructing test statistics. Input the test set into the well-trained MSFFNet and output the score vector:

$$h_{\theta^*}(\tilde{\boldsymbol{C}})=\left[\ h_{\theta^*|H_0}(\tilde{\boldsymbol{C}}),\quad h_{\theta^*|H_1}(\tilde{\boldsymbol{C}})\ \right], \tag{15}$$

where $h_{\theta^*|H_0}(\tilde{\boldsymbol{C}})+h_{\theta^*|H_1}(\tilde{\boldsymbol{C}})=1$, $h_{\theta^*}(\cdot)$ represents the trained model expression, $h_{\theta^*|H_i}(\cdot)$ represents the category probability of H_i, and $\tilde{\boldsymbol{C}}$ represents the test set sample. According to the Neyman–Pearson (NP) criterion, the test statistic T is defined as

$$T=\left[\frac{h_{\theta^*|H_1}(\tilde{\boldsymbol{C}})}{h_{\theta^*|H_0}(\tilde{\boldsymbol{C}})}\right]. \tag{16}$$

Step 2: setting the decision threshold. Input the H_0 class samples (q in total) into the well-trained MSFFNet, compute the test statistic using Eq. (16), and arrange the results in descending order to form a list $T_{H_0}=\{T_1,T_2,...,T_q\}$. Multiply the given false alarm probability P_{fa} by q, round down using the function $\lfloor\cdot\rfloor$, and use $\lfloor P_{\mathrm{fa}}\cdot q\rfloor$ as the index to retrieve the corresponding test statistic from T_{H_0}, which serves as the threshold, as shown in Eq. (17):

$$\gamma=T_{H_0}\left[\lfloor P_{\mathrm{fa}}\cdot q\rfloor\right]. \tag{17}$$

Step 3: online detection. In the online detection stage, real-time input samples are used to compute the test statistic using Eq. (16), which is then compared with the decision threshold to complete spectrum sensing:

$$\begin{cases} T\geq\gamma, & H_1,\\ T<\gamma, & H_0. \end{cases} \tag{18}$$

3.4 Spectrum sensing workflow based on MSFFNet

The workflow of the spectrum sensing algorithm based on MSFFNet is illustrated in Algorithm 1 and Fig. 6, comprising three stages: data preprocessing, feature extraction, and threshold decision. The dataset consists of m pairs of training data $D_m = \{(\boldsymbol{C}_1, y_1), \ldots, (\boldsymbol{C}_m, y_m)\}$, n pairs of validation data $D_n = \{(\boldsymbol{C}_{m+1}, y_{m+1}), \ldots, (\boldsymbol{C}_{m+n}, y_{m+n})\}$, and k pairs of test data $D_k = \left\{(\widetilde{\boldsymbol{C}}_1, \widetilde{y}_1), \ldots, (\widetilde{\boldsymbol{C}}_k, \widetilde{y}_k)\right\}$. Here, $\boldsymbol{C}$ denotes the grayscale image of the covariance matrix for the training and validation samples, $\widetilde{\boldsymbol{C}}$ represents the grayscale image of the covariance matrix for the test samples, and $\boldsymbol{y}$ is the one-hot label.

$$\boldsymbol{y} = \begin{cases} [1,0], & H_0, \\ [0,1], & H_1. \end{cases} \tag{19}$$

The loss function is the binary cross-entropy (BCE) loss:

$$\begin{aligned} \mathrm{Loss}(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\Big[& y_i \log h_{\theta|H_1}(\boldsymbol{C}_i) \\ & + (1-y_i)\log h_{\theta|H_0}(\boldsymbol{C}_i)\Big]. \end{aligned} \tag{20}$$

The model is trained on the training data, and the gradient of the loss function concerning the model parameter θ is computed and updated via backpropagation. The optimal model weight θ^* is determined using validation data. Finally, the test data are input into the trained model to perform online spectrum sensing based on the threshold decision mechanism.

Algorithm 1 Spectrum sensing algorithm based on the MSFFNet

Initialize best_acc $\leftarrow 0$ and randomly initialize the model parameters θ
Input the training dataset D_m and validation dataset D_n into $h_\theta(\cdot)$
repeat
  Compute the loss according to Eq. (20) and update θ
  Evaluate the accuracy on D_n using $h_\theta(\cdot)$, denoted as val_acc
  if val_acc > best_acc **then**
    best_acc $\leftarrow$ val_acc
    Save the current model parameters as optimal
  end if
until loss function converges
Save the optimal model parameters as θ^*
Compute the threshold γ using Eq. (17)
Input the test data D_k into $h_{\theta^*}(\cdot)$
Construct the test statistic T according to Eq. (16)
Determine the spectrum state based on Eq. (18)

4 Simulations

4.1 Simulation configuration

The dataset used in this paper is generated through simulations in MATLAB and includes the generation of PU signals, Rayleigh fading channels, and AWGN. The modulation scheme of the PU signal is orthogonal frequency-division multiplexing (OFDM). The number of receiving antennas in the CR terminal, M, is set to 32, with the number of sampling points N set to 5000. The simulated signal is sampled, and a sample covariance matrix grayscale map of dimension 128×128 is constructed. The simulation SNR range is -19 dB to 0 dB. The dataset consists of training, validation, and test sets, with 250 H_1 and 250 H_0 class samples generated for each

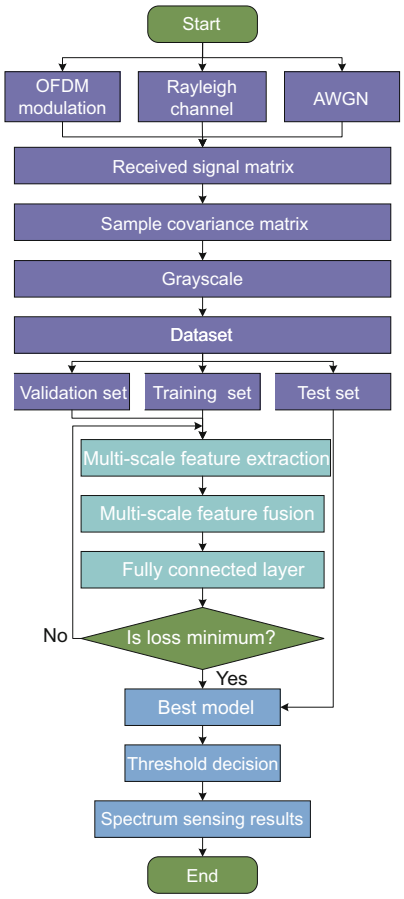


Fig. 6 Spectrum sensing workflow based on the MSFFNet

SNR level, totaling 10 000 samples. The training, validation, and test sets are split in an 8:1:1 ratio, with H_1 class samples labeled 1 and H_0 class samples labeled 0. The dataset construction details are shown in Table 2.

The simulations use an Intel® Core™ i5-12400F central processing unit (CPU) and an NVIDIA GeForce GTX 4060 (8 GB) graphics processing unit (GPU). The proposed algorithm is implemented in the PyTorch framework for the construction, training, and testing of the MSFFNet model. The BCE loss function is used for training, and the Adam optimizer is employed. The learning rate scheduler is StepLR. Hyperparameter settings are detailed in Table 3.

Table 2 Dataset construction

Dataset type	Label	Number	Total
Training set	0	4000	8000
	1	4000	
Validation set	0	500	1000
	1	500	
Test set	0	500	1000
	1	500	

Table 3 Training hyperparameters

Parameter	Value
Number of epochs	50
Batch size	32
Learning rate	0.001
Weight decay	0.001
Step size	20
Gamma	0.1

4.2 Simulation results and analysis

4.2.1 Detection probability of different spectrum sensing methods under low SNRs

This simulation compares the detection probabilities of seven spectrum sensing methods: MSFFNet, DCNN (Gai et al., 2021), ViT-B/16 (Dosovitskiy et al., 2021), CM-CNN (Liu et al., 2019), SenseNet (Geng et al., 2022), maximum–minimum eigenvalue detection (MME) (Zeng and Liang, 2009), and energy detection (ED) (Urkowitz, 1967). With the P_{fa} of 0.1, P_{d} is plotted across the −19 dB to 0 dB SNR range. As shown in Fig. 7, the MSFFNet consistently outperforms other spectrum sensing methods. At an SNR of −14 dB, the P_{d} of MSFFNet is 0.936. In contrast, DCNN, ViT-B/16, CM-CNN, SenseNet, MME, and ED achieve P_{d} values of 0.782, 0.818, 0.618, 0.464, 0.456, and 0.326, respectively. For SNR values below −11 dB, MSFFNet consistently achieves superior performance compared to others. This demonstrates the strong anti-noise robustness of MSFFNet under low-SNR conditions. The improved performance is attributed to its MSFEB, which allows the network to capture features at various scales, helping it resist noise interference and enhance detection accuracy.

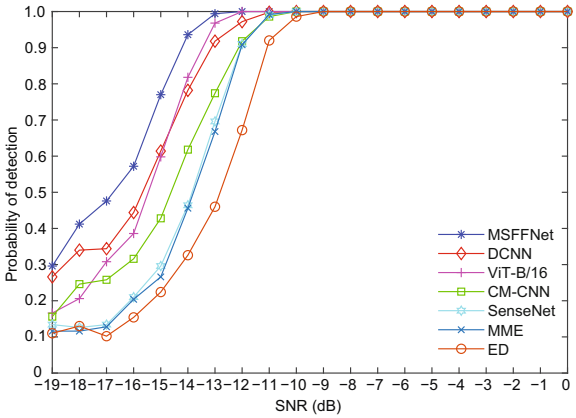


Fig. 7 P_{d}–SNR curves of the MSFFNet and six comparison methods

4.2.2 ROC curves of different spectrum sensing methods

Besides P_{d}, P_{fa} is another key metric for evaluating spectrum sensing methods. In this simulation, an SNR of −15 dB is used, and receiver operating characteristic (ROC) curves are compared for seven spectrum sensing methods: MSFFNet, DCNN, ViT-B/16, CM-CNN, SenseNet, MME, and ED. By adjusting P_{fa} and determining the corresponding threshold values, P_{d} of each model is obtained. The ROC curve visually represents the model's performance under varying P_{fa}. As shown in Fig. 8, under the same P_{fa}, MSFFNet significantly outperforms the others. In the low P_{fa} range, MSFFNet maintains a high P_{d}. At $P_{\mathrm{fa}}=0.1$, the P_{d} value of MSFFNet is 0.77. In contrast, the P_{d} values of DCNN, ViT-B/16, CM-CNN, SenseNet, MME, and ED are 0.614, 0.598, 0.428, 0.296, 0.266, and 0.224, respectively. This demonstrates that MSFFNet can effectively detect target signals while minimizing false alarms. The

model's ability to maintain high P_{d} across varying P_{fa} is attributed to its deeper and broader network architecture, along with MLFFM and skip connections, which enhances the feature representation at each layer and improves the gradient flow.

4.2.3 Ablation simulation

To assess the effectiveness of the AFSS and MLFFM in the MSFFNet, an ablation simulation was designed to evaluate four configurations: MSFFNet, MSFFNet without AFSS, MSFFNet without MLFFM, and MSFFNet without AFSS or MLFFM. Fig. 9 presents the P_{d} curves for the four models under the P_{fa} of 0.1 and an SNR range of [−19 dB, −10 dB]. The results indicate that the joint application of MSFEB, AFSS, and MLFFM achieves markedly better performance than any individual module.

This improvement originates from their complementary roles: (1) MSFEB extracts local and global features in parallel, thereby strengthening feature representation; (2) AFSS adaptively evaluates feature contributions across different scales, emphasizing critical features while suppressing redundant ones; (3) MLFFM combines shallow details with deep semantic features to mitigate information loss. Together, these modules establish a coherent process of "multi-scale feature extraction—adaptive feature screening—cross-level feature fusion," which markedly enhances the spectrum sensing under low-SNR conditions.

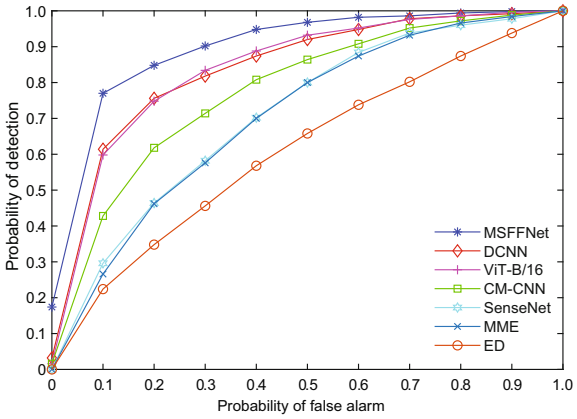


Fig. 8 ROC curves of the MSFFNet and six comparison methods

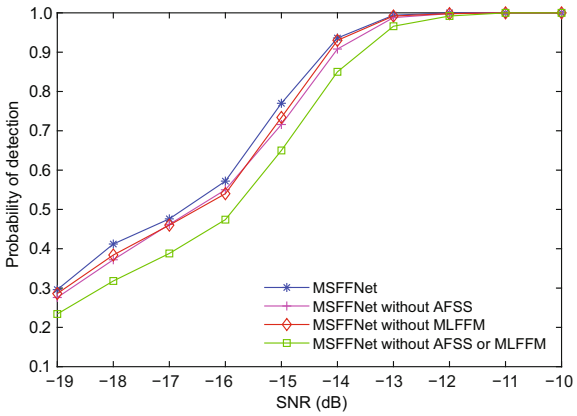


Fig. 9 Ablation simulation of the MSFFNet

4.2.4 Influence of the number of antennas and sampling points on the spectrum sensing performance

This simulation evaluates the scalability of the MSFFNet by analyzing its performance with different antenna counts (M = 16, 32, 64) and varying sampling point numbers (N = 2500, 5000, 10 000). With an SNR of −15 dB and 5000 sampling points, the ROC curve is employed to assess the detection performance of the proposed algorithm with varying antenna counts. Fig. 10 demonstrates that as the number of antennas increases, the MSFFNet exhibits improved detection performance. At the P_{fa} of 0.1, the P_{d} is 0.966 for M = 64, 0.770 for M = 32, and 0.396 for M = 16.

With an SNR of −15 dB and 32 antennas, the ROC curve of the MSFFNet is compared and analyzed under varying sampling points. Fig. 11 illustrates that as the number of sampling points increases, the detection performance of the MSFFNet improves. At a P_{fa} of 0.1, the P_{d} is 0.932 for N = 10 000, 0.77 for N = 5000, and 0.424 for N = 2500. Increasing the number of antennas and sampling points enhances the detection performance of

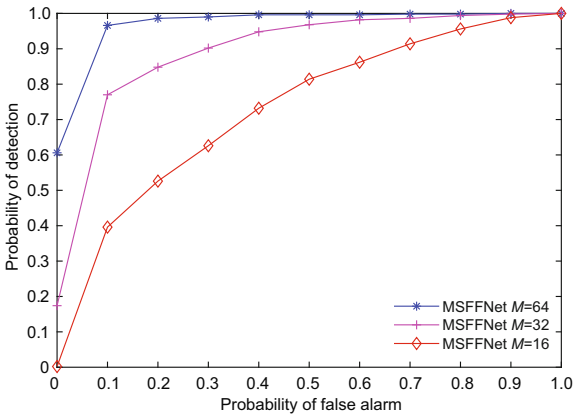


Fig. 10 ROC curves of the MSFFNet with different numbers of antennas

the MSFFNet in low-SNR environments. The multi-antenna system reduces the misjudgment caused by signal fading or multi-path interference on a single receiving path by leveraging spatial diversity. Additionally, more sampling points provide richer training data for the sensing model, particularly in complex channel environments, thus improving the model's overall performance.

4.2.5 Analysis of robustness of spectrum sensing methods under colored noise

To evaluate the generalization ability and robustness of the proposed model under non-ideal channel conditions, this study further analyzes its performance in a colored noise (Zhivomirov, 2018) environment.

In this simulation, colored noise is used in place of AWGN, and a dataset is constructed using the same configuration settings as in Section 4.1 to evaluate various methods. Fig. 12a illustrates the P_{d}–SNR curves of different methods under the condition of $P_{\mathrm{fa}}=0.1$. The results indicate that the MSFFNet maintains high detection accuracy in the presence of colored noise. When the SNR exceeds −13 dB, P_{d} surpasses 0.9. At SNR = −13 dB, the P_{d} value of MSFFNet is 0.938, significantly outperforming DCNN (0.838), ViT-B/16 (0.820), CM-CNN (0.802), SenseNet (0.676), MME (0.552), and ED (0.422). Fig. 12b illustrates the ROC curves of each method under SNR = −15 dB. These results are consistent with the P_{d}–SNR analysis. The MSFFNet maintains a high P_{d} across the P_{fa} range, and its area under the curve (AUC) is significantly higher than those of the comparison methods.

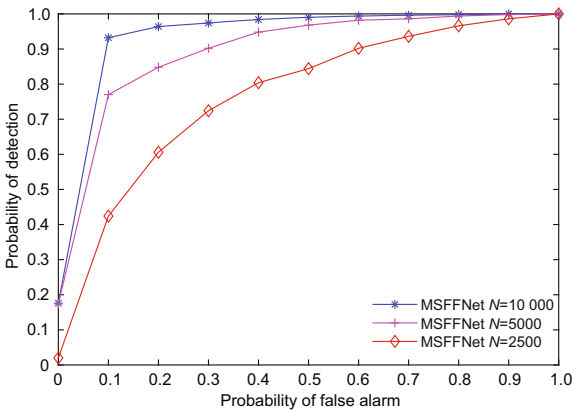


Fig. 11 ROC curves of the MSFFNet with different numbers of sampling points

Furthermore, Fig. 13 compares the P_{d}–SNR curves of MSFFNet under AWGN and colored noise conditions to evaluate its robustness against different noise statistical characteristics. Simulation results show that although the performance slightly decreases under colored noise interference, the overall detection accuracy remains stable, demonstrating strong noise resilience. As the SNR increases,

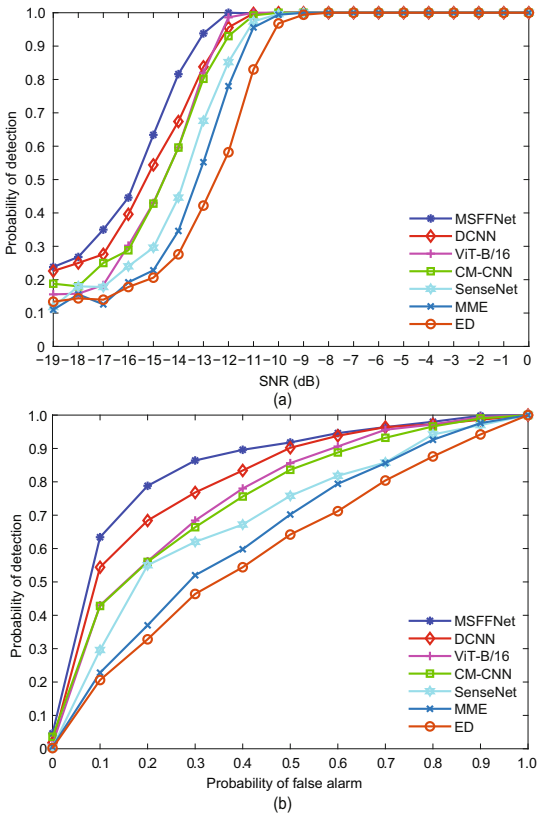


Fig. 12 Detection performance of spectrum sensing methods under colored noise: (a) P_{d}–SNR curves; (b) ROC curves

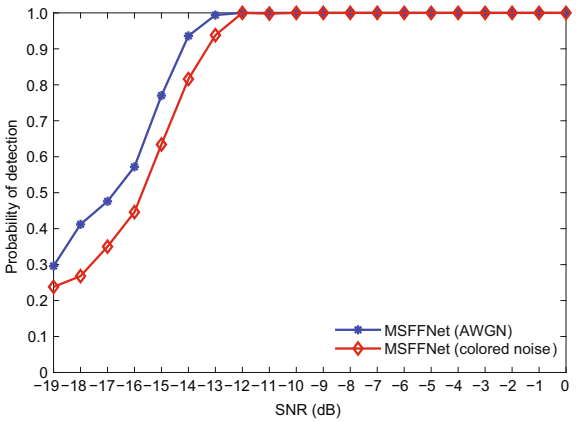


Fig. 13 P_{d}–SNR curves of the MSFFNet under AWGN and colored noise

the model's detection performance under the two noise conditions gradually converges. These findings indicate that MSFFNet adapts well to diverse noise distributions and sustains effective detection in complex channel conditions, demonstrating robust performance.

4.2.6 Analysis of model complexity and real-time performance

To assess both resource consumption and detection capability, we conducted a complexity comparison between MSFFNet and four baseline models. Two toolkits were used: torchsummary for memory footprint and ptflops for parameters and multiply–accumulate operations (MACs). Table 4 reports spatial complexity, temporal complexity, and detection performance P_{d} with $P_{\mathrm{fa}} = 0.1$ and SNR= -14 dB.

For spatial complexity, ViT-B/16 requires 735.83 MB of memory and 8.58×10^7 parameters, far exceeding those of the other models. By contrast, MSFFNet reduces the resource usage while maintaining high detection accuracy, requiring only 104.49 MB of memory and 2.222×10^7 parameters; MSFFNet achieves superior overall detection performance.

For temporal complexity, MSFFNet requires 9.6962×10^8 MACs, higher than those of single-scale models, such as SenseNet, DCNN, and CM-CNN, mainly due to its multi-scale convolutional structure. Under low-SNR conditions, however, MSFFNet achieves the highest detection performance, with $P_{\mathrm{d}} = 0.936$. In comparison, SenseNet (0.464), DCNN (0.782), and CM-CNN (0.618) fail to provide reliable sensing. Although ViT-B/16 reaches a P_{d} of 0.818, its computational overhead is far greater.

Accordingly, Table 5 presents a comparison of the training time and single-sample inference time for each model. Although MSFFNet has a longer training time (851 s) compared to SenseNet (494 s), DCNN (577 s), and CM-CNN (596 s), this does not affect real-time decision-making in spectrum sensing, as training is performed offline. In terms of inference efficiency, SenseNet (1 ms) achieves the fastest inference. MSFFNet (4 ms) exhibits comparable inference to CM-CNN (2 ms) and DCNN (2 ms) and significantly outperforms ViT-B/16 (9 ms), meeting real-time processing requirements.

Although MSFFNet exhibits slightly higher time complexity, it delivers markedly better detection performance under low-SNR conditions while maintaining manageable resource overhead. This demonstrates a favorable balance between performance and efficiency. Moreover, its modular design facilitates lightweight adaptations for spectrum sensing under diverse resource constraints.

4.2.7 Observability analysis of simulation results

The feature extraction capability of the MSFEB in MSFFNet is evaluated through visualization. As shown in Fig. 14, an H_1 sample is fed into the trained MSFFNet, and multi-scale features are sequentially extracted by MSFEB 1–3. The results indicate that deeper layers extract features with progressively higher levels of semantic abstraction. Shallow modules capture basic input patterns, such as main diagonal elements and edge contours, producing high-resolution feature maps. Middle-level modules produce more abstract representations by enhancing and combining local features, thereby laying the foundation for high-level semantics. Deep modules emphasize high-level semantic features, reducing spatial resolution while expanding the number of channels. This reflects the network's compact representation of semantic information. These visualizations confirm the effectiveness of MSFEB in layer-wise abstraction and multi-scale fusion, demonstrating its progressive expression from low-level to high-level semantic features.

Table 4 Comparison of the complexity of spectrum sensing models

Model	Memory (MB)	N_{par} ($\times 10^6$)	N_{MACs} ($\times 10^6$)	P_{d}
SenseNet	17.89	4.21	5.32	0.464
DCNN	22.18	0.10	76.41	0.782
MSFFNet	104.49	22.22	969.62	0.936
CM-CNN	107.36	25.63	138.86	0.618
ViT-B/16	735.83	85.80	16 880	0.818

N_{par}: number of parameters; N_{MACs}: number of MACs

Table 5 Comparison of training and inference time for spectrum sensing models

Model	Training time (s)	Inference time (ms)
SenseNet	494	1
CM-CNN	596	2
DCNN	577	2
MSFFNet	851	4
ViT-B/16	4532	9

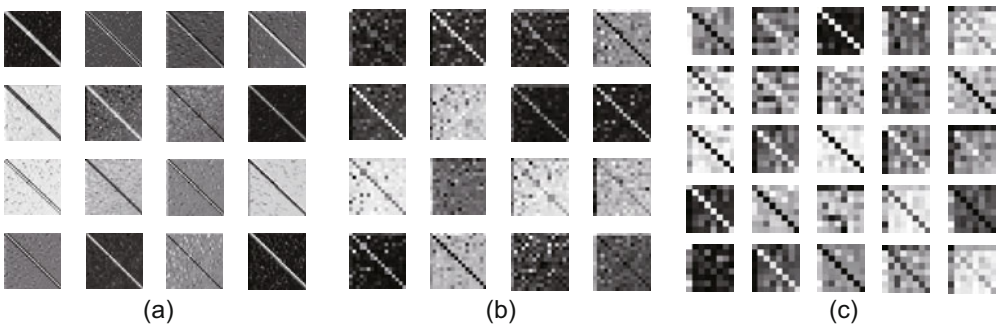


Fig. 14 Visualization of feature maps at different levels: (a) MSFEB 1; (b) MSFEB 2; (c) MSFEB 3

5 Conclusions and future work

5.1 Conclusions

To address the limitations of traditional CNNs in low-SNR spectrum sensing tasks, this paper proposes a spectrum sensing method based on MSFFNet. The MSFFNet employs MSFEB to capture key information from OFDM signals at different scales, allowing the model to learn local and global signal features. Furthermore, an AFSS is designed to optimize feature representation. Feature information at different levels is integrated through MLFFM, which preserves the deep network's high-dimensional representation capability while mitigating feature loss caused by excessively deep layers. This network design effectively addresses the feature extraction limitations of CNNs and significantly enhances the detection accuracy and robustness for spectrum sensing tasks. Simulation results demonstrate that in low-SNR scenarios, the MSFFNet-based spectrum sensing method outperforms other sensing models, achieving significantly higher detection performance at the same false alarm probability.

5.2 Future work

Although the current DL techniques have achieved significant success in the Internet of Things (IoT) domain, such as in automatic modulation classification (Ma et al., 2025), channel state information feedback (Li JX et al., 2025), and spectrum sensing (Kim et al., 2023), their high computational, data, time, and energy costs continue to hinder further development. To address this, swarm intelligence (SI) optimization methods, including particle swarm optimization (Yan et al., 2023), harmony search algorithm (Ye et al., 2023; Qin et al., 2025), and grey wolf optimizer (He et al., 2022), as well as inference models (Jiang et al., 2022; Xiang et al., 2025), are being integrated to develop lightweight sensing models. Moreover, model compression techniques, including pruning (Pham et al., 2024), quantization, and knowledge distillation (Gou et al., 2024), have been extensively investigated for accelerating DNNs. Inspired by these findings, the future work could be categorized into two directions:

1. Lightweight DL formalism. SI algorithms will be used to overcome the limited processing capabilities of edge and IoT devices, leveraging CR principles. These efforts can be further combined with pruning, quantization, and knowledge distillation methods to achieve efficient and accurate lightweight sensing models.

2. Adaptive thresholding. SI algorithms will help develop a learnable thresholding mechanism to address the limitations of empirical threshold settings in dynamic communication environments.

Contributors

Kejun LEI designed the research. Honghui XIANG and Wenjing TUO processed the data. Honghui XIANG and Hongbin LIU drafted the paper. Kaiqing ZHOU helped organize the paper. Kejun LEI and Kaiqing ZHOU revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Bai WW, Zheng GQ, Mu Y, et al., 2025. Cooperative spectrum sensing method based on channel attention and parallel CNN-LSTM. *Dig Signal Process*, 158:104963. https://doi.org/10.1016/j.dsp.2024.104963

Cai LN, Cao KT, Wu YP, et al., 2022. Spectrum sensing based on spectrogram-aware CNN for cognitive radio network. *IEEE Wirel Commun Lett*, 11(10):2135-2139. https://doi.org/10.1109/lwc.2022.3194735

Dosovitskiy A, Beyer L, Kolesnikov A, et al., 2021. An image is worth 16×16 words: transformers for image recognition at scale. Proc 9th Int Conf on Learning Representations.

Gai JX, Xue XF, Wu JY, et al., 2021. Cooperative spectrum sensing method based on deep convolutional neural network. *J Electron Inform Technol*, 43(10):2911-2919 (in Chinese). https://doi.org/10.11999/JEIT201005

Gai JX, Xue XF, Nan RX, et al., 2022. Spectrum sensing method based on residual dense network. *J Commun*, 42(12):182-191 (in Chinese). https://doi.org/10.11959/j.issn.1000-436x.2021220

Gao ZH, Li YF, Chen Z, et al., 2024. Intelligent spectrum sensing of consumer IoT based on GAN-GRU-YOLO. *IEEE Trans Consumer Electron*, 70(3):6140-6148. https://doi.org/10.1109/TCE.2024.3418103

Geng Y, Huang JY, Yang JX, et al., 2022. Spectrum sensing for cognitive radio based on feature extraction and deep learning. *J Phys Conf Ser*, 2261(1):012016. https://doi.org/10.1088/1742-6596/2261/1/012016

Gou JP, Chen Y, Yu BS, et al., 2024. Reciprocal teacher-student learning via forward and feedback knowledge distillation. *IEEE Trans Multimed*, 26:7901-7916. https://doi.org/10.1109/tmm.2024.3372833

He ZH, Jin G, Wang YJ, 2022. A novel grey wolf optimizer and its applications in 5G frequency selection surface design. *Front Inform Technol Electron Eng*, 23(9):1338-1353. https://doi.org/10.1631/FITEE.2100580

Hong S, Xu WQ, 2025. Spectrum sensing in very low SNR environment using multi-scale temporal correlation perception with residual attention. *Sensors*, 25(2):528. https://doi.org/10.3390/s25020528

Hou QB, Zhou DQ, Feng JS, 2021. Coordinate attention for efficient mobile network design. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.13713-13722. https://doi.org/10.1109/cvpr46437.2021.01350

Hu J, Shen L, Sun G, 2018. Squeeze-and-excitation networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.7132-7141. https://doi.org/10.1109/cvpr.2018.00745

Janu D, Singh K, Kumar S, et al., 2023. Hierarchical cooperative LSTM-based spectrum sensing. *IEEE Commun Lett*, 27(3):866-870. https://doi.org/10.1109/lcomm.2023.3241664

Jiang W, Zhou KQ, Sarkheyli-Hägele A, et al., 2022. Modeling, reasoning, and application of fuzzy Petri net model: a survey. *Artif Intell Rev*, 55(8):6567-6605. https://doi.org/10.1007/s10462-022-10161-0

Kim H, Kim YJ, Kim WT, 2023. Multitask learning-based deep signal identification for advanced spectrum sensing. *Sensors*, 23(24):9806. https://doi.org/10.3390/s23249806

Li JC, Fang FM, Mei KF, et al., 2018. Multi-scale residual network for image super-resolution. Proc 15th European Conf on Computer Vision, p.527-542. https://doi.org/10.1007/978-3-030-01237-3_32

Li JX, Wang YJ, Liu Z, et al., 2025. Metaformer-based lightweight neural network for massive MIMO CSI feedback. *IEEE Wirel Commun Lett*, 14(2):275-279. https://doi.org/10.1109/LWC.2024.3493236

Li SP, Han YZ, Gaber J, et al., 2024. A multiantenna spectrum sensing method based on HFDE-CNN-GRU under non-Gaussian noise. *Int J Intell Syst*, 2024:1085161. https://doi.org/10.1155/2024/1085161

Liu C, Wang J, Liu XM, et al., 2019. Deep CM-CNN for spectrum sensing in cognitive radio. *IEEE J Sel Areas Commun*, 37(10):2306-2321. https://doi.org/10.1109/jsac.2019.2933892

Ma QY, Xiao J, Shao ZQ, et al., 2025. Frequency-learning adversarial networks based on transfer learning for cross-scenario signal modulation classification. *Front Inform Technol Electron Eng*, 26(5):816-832. https://doi.org/10.1631/FITEE.2400080

Pan GL, Wu QH, Zhou B, et al., 2025. Spectrum prediction with deep 3D pyramid vision transformer learning. *IEEE Trans Wirel Commun*, 24(1):509-525. https://doi.org/10.1109/TWC.2024.3495812

Peng TQ, Yang SY, Feng ZX, et al., 2024. Spectrum sensing via residual dilated network and horizontal shift attention for cognitive IoT. *IEEE Internet Things J*, 11(22):36817-36828. https://doi.org/10.1109/JIOT.2024.3429200

Pham VT, Zniyed Y, Nguyen TP, 2024. Efficient tensor decomposition-based filter pruning. *Neur Netw*, 178:106393. https://doi.org/10.1016/j.neunet.2024.106393

Qin F, Zain AM, Zhou KQ, et al., 2025. Hybrid weighted fuzzy production rule extraction utilizing modified harmony search and BPNN. *Sci Rep*, 15(1):11012. https://doi.org/10.1038/s41598-025-95406-y

Saraswathi M, Logashanmugam E, 2024. Chicken swarm optimization modelling for cognitive radio networks using deep belief network-enabled spectrum sensing technique. *PLoS ONE*, 19(8):e0305987. https://doi.org/10.1371/journal.pone.0305987

Sutton PD, Nolan KE, Doyle LE, 2008. Cyclostationary signatures in practical cognitive radio applications. *IEEE J Sel Areas Commun*, 26(1):13-24. https://doi.org/10.1109/jsac.2008.080103

Urkowitz H, 1967. Energy detection of unknown deterministic signals. *Proc IEEE*, 55(4):523-531. https://doi.org/10.1109/proc.1967.5573

Xiang YH, Zhou KQ, Sarkheyli-Hägele A, et al., 2025. Parallel fault diagnosis using hierarchical fuzzy Petri net by reversible and dynamic decomposition mechanism. *Front Inform Technol Electron Eng*, 26(1):93-108. https://doi.org/10.1631/FITEE.2400184

Xu LW, Li YF, Gulliver TA, 2025. Federal underwater acoustic spectrum sensing algorithm based on DCYOLO. *Expert Syst Appl*, 275:126928. https://doi.org/10.1016/j.eswa.2025.126928

Yan H, Zhong CQ, Wu YH, et al., 2023. A hybrid-model optimization algorithm based on the Gaussian process and particle swarm optimization for mixed-variable CNN hyperparameter automatic search. *Front Inform Technol Electron Eng*, 24(11):1557-1573. https://doi.org/10.1631/FITEE.2200515

Ye SQ, Zhou KQ, Zain AM, et al., 2023. A modified harmony search algorithm and its applications in weighted fuzzy production rule extraction. *Front Inform Technol Electron Eng*, 24(11):1574-1590. https://doi.org/10.1631/FITEE.2200334

Zeng YH, Liang YC, 2009. Eigenvalue-based spectrum sensing algorithms for cognitive radio. *IEEE Trans Commun*, 57(6):1784-1793. https://doi.org/10.1109/tcomm.2009.06.070402

Zhang WS, Wang Y, Chen X, et al., 2024. Spectrum transformer: an attention-based wideband spectrum detector. *IEEE Trans Wirel Commun*, 23(9):12343-12353. https://doi.org/10.1109/twc.2024.3391515

Zhivomirov H, 2018. A method for colored noise generation. *Roman J Acoust Vibrat*, 15(1):14-19. https://rjav.sra.ro/index.php/rjav/article/view/40