TFF_aDCNN: A Pre-trained Base Model for Intelligent Wideband Spectrum Sensing

Xianghui Li, Zhuhua Hu, *Senior Member, IEEE,* Chong Shen, Huaming Wu, *Senior Member, IEEE,* and Yaochi Zhao

***Abstract*—In the beyond 5G (B5G)/6G era, to achieve ultra-dense and ultra-large-capacity intelligent connection of all things, an intelligent wideband spectrum sensing technology is particularly important. However, in an extremely wide frequency range, it is still a challenge to achieve high-precision and high-reconstruction-capability wideband spectrum sensing (WSS) under a very low SNR. We propose a Time-Frequency-Fused adjustable Deep Convolutional Neural Network (TFF_aDCNN). Meanwhile, a novel TFF_aDCNN-based sensing framework is also proposed. In this framework, we can obtain a pre-trained base model with a single distribution by training TFF_aDCNN. Then, for the sensing task in the actual environment, we use the base model for transfer learning, so that a newly trained sensing model can be obtained very quickly (i.e. fine-tuned model). In the TFF_aDCNN, we design a main network and an adjustable auxiliary network, where the former learns complex and abstract signal features, while the latter assists the main network in learning different data distribution patterns during the training process and regulates the focus direction of the main network during the perception process. Simulation results show that TFF_aDCNN can significantly reduce hardware cost and improve reconstruction accuracy and reconstruction capability, when compared with SOMP and SwSOMP-based WSS algorithms, single-dimensional deep learning spectrum sensing method, and deep learning-based WSS (DLWSS), especially at very low SNRs.**

***Index Terms*—Deep Convolutional Neural Network, Modulated Wideband Converter, Spectrum Sensing, Time-Frequency correlation, PCA**

I. INTRODUCTION

SPECTRUM is an important and scarce strategic resource. With the rapid development of next-generation wireless communication technologies and the Internet of Things (IoT), the types and numbers of devices accessing wireless networks are exploding [1]. Therefore, to meet the spectrum requirements of various wideband devices, the future 6G communication technology will inevitably move into the ultra-wide spectrum range of millimeter wave and terahertz [2], [3]. However, the current static spectrum allocation strategy has made the contradiction increasingly acute between the low utilization of spectrum resources and the shortage of spectrum resources [4]. In the complex electromagnetic environment, the traditional Wideband Spectrum Sensing(WSS) method based on cognitive radio (CR) [5] is difficult to meet future performance requirements. Wideband spectrum sensing faces several challenges that narrowband spectrum sensing doesn't have. First of all, a wide range of spectrum requires super high-speed ADC to sample. Secondly, wideband spectrum sensing needs a large storage room for data. Although some methods can be used to achieve sub-Nyquist sampling, the signal will be contaminated with critical aliasing. Using sub-Nyquist sampling can reduce the requirement of a high-speed ADC, but the sampled signal is difficult to process. However, with the rapid development of artificial intelligence (AI) technology, it is a feasible solution to realize smart spectrum sensing in a very wide spectral range with the help of AI [6].

Spectrum sensing allows secondary users (SUs) to sense the spectrum occupancy state of authorized primary users (PUs) in the surrounding complex radio environment. SUs use spectrum sensing technology to discover spectrum holes [7], and then access them to realize spectrum sharing, which can greatly improve spectrum utilization and alleviate the scarcity problem of high-quality spectrum resources. Most of the existing traditional spectrum sensing methods based on sub-Nyquist sampling do not determine whether there is a PU in the spectrum, and directly reconstruct the support set, resulting in a higher false alarm rate and computational cost.

With the computational power substantially increased, deep learning has exerted a powerful capability. Deep learning methods are able to find out the mapping model of signal to support set in a complex radio environment and do not need to extract features manually. Convolutional neural networks (CNN) have become the fundamental feature extraction network in image processing due to their excellent feature extraction ability. There are a number of studies that have attempted to take advantage of deep learning for spectrum sensing.

Unfortunately, previous studies (regardless of using traditional or deep learning-based spectrum sensing approaches) still suffer from the following three problems:

- The relationship between the PUs' activity habits and the time period is often ignored. So that an effective and reasonable way is needed to fuse the time and the frequency, which can achieve the purpose of improving

Manuscript received May 8, 2022, revised September 14 2022 and February 5 2023; accepted April 21 2023. Date of publication xxx May 2023; date of current version xxx May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant no. 61963012 and in part by the Key Research and Development Project of Hainan Province under Grant no. ZDYF2022SHFZ039 and ZDYF2022GXJS348. The associate editor coordinating the review of this paper and approving it for publication was xxx xxx. (Corresponding authors: Zhuhua Hu and Huaming Wu.)

X. Li, Z. Hu and C. Shen are with the School of Information and Communication Engineering, Hainan University, Haikou 570228, China (e-mail: lxh1318072838@163.com; {eagler_hu, chongshen}@hainanu.edu.cn).

Y. Zhao is with the School of Cyberspace Security (School of Cryptology), Hainan University, Haikou 570228, China (e-mail: zhyc@hainanu.edu.cn).

H. Wu is with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: whming@tju.edu.cn).

perception accuracy.

- Under a low signal-to-noise ratio (SNR) environment, the existing modulated wideband converter (MWC)-based wideband spectrum sensing (WSS) algorithms need more parallel channels to ensure the support set reconstruction probability. Thus, a WSS method that can reduce hardware costs and improve reconstruction capability while keeping the same accuracy is needed.
- The reconstruction ability of existing schemes under a low SNR environment is also greatly limited. Therefore, an effective method is needed to denoise the sampling results and reduce the influence of noise on the model reconstruction ability.

To solve the above-mentioned challenges, a complete smart WSS framework is proposed in this paper. The framework consists of one preparation stage and three implementation stages. In the preparation stage, a time-frequency relationship model is constructed using PUs' spectrum usage habits in different time periods. Then, a sparse wideband signal is obtained based on this model and the signal generation model. The implementation stages include: (i) MWC wideband compressed sampling stage; (ii) Data pre-processing stage; (iii) End-to-end smart WSS stage based on time-frequency fusion. In the second stage, the estimated original signal is denoised by PCA after compressed sampling, and one-hot encoding is used to construct time dimension information. In the third stage, the denoised signal and time dimension information are used as training samples to train a new smart WSS model. Our specific contributions are as follows:

- A time-frequency-fused adjustable deep convolutional neural network (TFF_aDCNN) is innovatively proposed. TFF_aDCNN consists of a main network and an adjustment network. The former is mainly used for learning complex and abstract signal features, and the latter is mainly used to assist the main network in learning the current data distribution patterns.
- A pre-trained base model is obtained by training TFF_aDCNN. Then, we can perform transfer learning based on the base model to obtain a new wideband spectrum sensing model in different electromagnetic environments (also called a fine-tuned model).
- We reasonably assume that the usage habits of the authorized PUs obey a regular pattern according to the time period. By mathematically modeling these assumptions, a time-frequency correlation model is constructed to facilitate the simulation experiments.
- A novel noise reduction operation is carried out in pre-processing. That is, we first obtain the estimated original signal sampled at two consecutive times. Then, the two estimated signals are expanded into two one-dimensional column vectors and stacked along the column direction to form a two-dimensional matrix. Finally, the formed matrix signal is denoised by principal component analysis (PCA).

The rest of the paper is organized as follows: Section II investigates the related work from the aspects of traditional spectrum sensing approaches and deep learning-based spectrum sensing approaches. Section III models the smart WSS system, where the characteristics of the spectrum being occupied are related to the time, and also introduces the MWC model, the PCA denoising model, and the data pre-processing model. Section IV carries out the problem formulation and further describes TFF_aDCNN network in detail. Section V evaluates the performance of the whole framework through simulation results. Finally, Section VI concludes the whole paper and further points out future directions.

II. Related Work

A. Traditional Spectrum Sensing Approaches

In traditional approaches, WSS methods can sense spectrum occupancy state over an extremely wide frequency range at a time, giving SUs more access opportunities compared to narrowband spectrum sensing methods [8]–[12]. The spectrum occupancy is usually sparse in time, frequency and space due to the underutilization of the allocated spectrum [13]. Therefore, multiband sparse signals are usually a common form of signal in practical communications. For such signals, a low-speed analog-to-digital converter can be used for high-speed sampling [14]–[16], and the signals can be recovered from a small number of linear random measurements. Modulated wideband converter (MWC) [14] is a typical method to achieve sub-Nyquist sampling using multiple parallel channels and is frequently used in conventional wideband compressed spectrum sensing algorithms.

Hu *et al.* [17] proposed a SwSOMP algorithm based on MWC, which uses a stage-wise weak selection strategy in simultaneous orthogonal matching pursuit (SOMP) [18]. The algorithm can effectively improve the reconstruction accuracy of the wideband spectrum support set and reduce the computational cost under the Gaussian noise interference. The noise intensity and signal sparsity can be estimated using singular value decomposition. Based on that, an adaptive and blind reduced Multiple Measurement Vectors (MMV) boost (ABRMB) framework [19] was proposed. The framework can adaptively process multiband signals using the estimated noise intensity and signal sparsity, and can improve the support set reconstruction probability. Using the approximate linear characteristics of the sparse multi-band signal tail singular value and the progressive support selection strategy, an progressive support selection-based self-adaptive distributed MWC sensing scheme (PSS-SaDMWC) [20] was proposed. When there are fewer cooperative SUs, the reconstruction probability of the support set can be significantly improved. The machine learning-based WSS scheme [21] used MWC to obtain sampling results and then used sparse Bayesian learning to extract information directly from the sampling results to estimate the support set. The scheme reduces computational complexity by removing the continuous-to-finite (CTF) block and the pseudo-inversion operation. A reconstruction algorithm called nearest orthogonal matching pursuit (N-OMP) based on MWC was proposed [22]. After the occupied sub-bands are detected, the correlation coefficients between the residual vectors and the corresponding column vectors of the two adjacent sub-bands are calculated. The occupancy state of the adjacent sub-bands can be directly determined according to correlation coefficients.

In order to reduce the higher false alarm rate and computational cost in existing spectrum sensing methods, a pairwise channel energy ratio (PCER) detector algorithm was proposed [23]. Before the signal support set reconstruction is performed, the MWC sampling result is used to determine whether PUs exist in the wideband spectrum. The signal support set reconstruction algorithm is performed only after the PUs exist in the wideband spectrum. The algorithm is robust to different SNRs and does not require prior knowledge of PU signals. However, to achieve the support set reconstruction probability greater than 90% in the traditional single-node compressed sensing method, it needs to be in a high SNR environment or increase hardware overhead. This requirement is difficult to meet in actual deployment applications.

B. Deep Learning-based Narrowband Spectrum Sensing Methods

Narrowband spectrum sensing can also take advantage of deep learning for classification accuracy improvement. Firstly, the convolutional, long short-term memory, fully connected deep neural networks (CLDNN) in deep learning [24] were applied to solve the narrowband spectrum sensing problem [25], which can classify the state of narrowband signals effectively. Secondly, Gao *et al.* [26] used an improved CLDNN, which fuses the input with the first three convolution layers' output as the input of long short-term memory (LSTM) for spectrum sensing. Because some information is lost after multi-layer convolution operation, the structure [26] ensures that the data input to the LSTM network retains all features of the original data. The activity pattern aware spectrum sensing (APASS) algorithm [27] was proposed considering the PUs' activity patterns. The algorithm simultaneously takes in the present sensing data and historical sensing data, with which the inherent PU activity pattern can be learned to benefit the detection of PU activity. However, the algorithm needs to retain and feed a large amount of historical information into the trained network each time, which increases the computational complexity. All of the above methods are based on a combination of narrowband spectrum sensing and deep learning.

C. Deep Learning-based Wideband Spectrum Sensing Methods

Wideband spectrum sensing faces several challenges that narrowband spectrum sensing doesn't have. First of all, a wide range of spectrum requires super high-speed ADC to sample. Secondly, wideband spectrum sensing needs a large storage room for data. Although some methods can be used to achieve sub-Nyquist sampling, the signal will be contaminated with critical aliasing. Using sub-Nyquist sampling can reduce the requirement of a high-speed ADC, but the sampled signal is difficult to process. Research combining wideband compressed sensing and deep learning has also been carried out. Firstly, sampling results are obtained by sub-Nyquist sample methods, and the obtained estimated original signal is input to a deep compressed spectrum sensing generative adversarial network (DCSS-GAN) [28] to reconstruct the original signal spectrum. Then, DCSS-GAN uses a reconstructed spectrum to classify the PUs' band occupancy state as a multi-label classification problem. In addition, chandhok *et al.* [29] proposed a deep learning-based wideband spectrum sensing (DLWSS) method by utilizing MWC for WSS. The estimated original signal is fed into a three-layer CNN, and the band occupancy state is determined by outputting the predicted value on each band through the fully connected layer. Subsequently, considering the relationship between spectrum and space, a CNN-based cooperative spectrum sensing model [30] was proposed. The sensing results are arranged into a two-dimensional matrix in spatial order in the fusion center. The sensing results are obtained by each SU through energy detection and the CNN improves reconstruction probability by learning the correlation between space and frequency band.

D. Related Work on Spectrum Distribution in Time Domain

In the research of spectrum sensing, studies considering both spatial-temporal correlations [31]–[33] and space-frequency correlation [34] have been reported, and these works have obtained an improvement in sensing performance. However, it is still an innovative work to use the information of time-frequency correlation modeling as the prior input for deep learning-based wideband spectrum sensing models. In fact, for a specific important application scenario in the same region, the occupancy of the spectrum by the primary user has a regular distribution in the time domain [35]–[37]. The related research on spectrum utilization shows that although the average utilization rate of high-quality frequency bands is higher than that of the non-high-quality spectrum, the absolute utilization rate is still very low. Therefore, we have sufficient reasons to assume that the idleness of the spectrum has a certain distribution relationship with time [38]–[40]. The research and analysis results of relevant literature can also prove that our hypothesis is reasonable and reliable.

III. System Modeling and Compressed Sampling

A. System Modeling

As shown in Fig. 1, we consider a wideband spectrum over 1 GHz and a WSS part in a region where there are several PUs and a SU. According to reality, PUs use different devices at different time periods, then the spectrum occupancy under different time periods in the region has regularity.

Assuming that a very wide band of width B_w is divided into L consecutive non-overlapping narrow bands of width B with band indexes $1, 2, \cdots, L$. The PUs in the system are allowed to use at most a total of N $(N < L)$ consecutive non-overlapping narrow bands of width B at the same time. We divide the day into z time periods, and the state of the k-th band in time period $T_j \in \{T_1, T_2, \cdots, T_z\}$ is denoted as $S_k \in \{0, 1\}$. $S_k = 1$ means that the k-th band is being used, and $S_k = 0$ means that the k-th band is idle. PUs do not occupy all available bands, at which point the SU is able to sense the state $B_{state} = \{S_1, S_2, \cdots, S_L\}$ of each band in the range B_w. We define the set consisting of the occupied

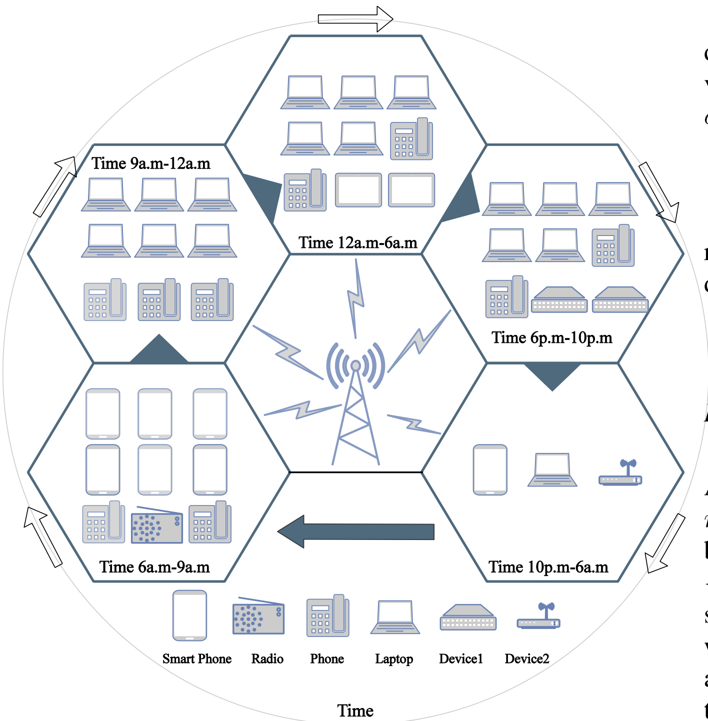


Fig. 1: A spectrum sensing system with the usage habits of the authorized PUs in different time periods obeys a regular pattern.

bands' indexes M belonging to the time period T_j as the signal support set, defined as:

$$\Lambda = \{M_1, M_2, \cdots, M_N\}. \tag{1}$$

Sparse multi-band signal is a type of signal often encountered in cognitive radio communication. We assume that the received signal $x(t)$ is a sparse band-pass analog signal with the frequency spectrum distributed in $B_w = [-f_{nyq}/2\ , f_{nyq}/2\ ]$, and f_{nyq} is the Nyquist sampling rate of the signal. The received signal is as follows:

$$x(t) = p(t) + w(t). \tag{2}$$

where $w(t)$ is the additive white noise that obeys the Gaussian distribution, $p(t)$ is the superposition of the signals from the active PUs at time t in the area, the maximum bandwidth of the signal from PUs is $B_{\max}$, and the signal energy is $E = [E_1, \cdots, E_N]$.

Considering the fact that the activeness of PUs in an area is closely related to the time period, the occupied frequency band has different distributions in different time periods. For example, at around 8: 00 in the morning, the frequency band used by navigation begins to be active, and at around 9: 00, the frequency band used by operators is very active. After 24 o'clock in the evening, there is no particularly active frequency band, but there are still PUs occupying the spectrum randomly. In each time period, except for some specific active frequency bands, other frequency bands will also be randomly occupied by PUs, which is very similar to the characteristics of normal distribution.

Therefore, we assume that the carrier frequency $f_c(T_j)$ in daytime signal $p(t)$ obeys the normal distribution with mean value $\mu(T_j) = \{\mu(T_1), \cdots, \mu(T_z)\}$ and standard deviation $\sigma(T_j) = \{\sigma(T_1), \cdots, \sigma(T_z)\}$ as follows:

$$f_c(T_j) \sim N\left(\mu(T_j), \sigma^2(T_j)\right). \tag{3}$$

After 24 o'clock at night, the spectrum occupation is random, and the carrier frequency of $p(t)$ obeys uniform distribution:

$$f_c(T_j) \sim U(-f_{nyq}/2, f_{nyq}/2). \tag{4}$$

B. Compressed Sampling Process Using MWC

The MWC algorithm is a method of sub-Nyquist sampling. Assuming that the number of parallel sampling channels is m, and on a certain channel g, the signal $x(t)$ is multiplied by a set of ± 1 randomly alternating waveforms $C_g(t), g \in \{1, 2, \cdots, m\}$ with a period of $T_p = 1/f_p$ to realize the shift of the signal spectrum $X(f)$. Then pass a low-pass filter with a frequency of $1/2T_s$ to obtain a baseband signal with a frequency range of $F_s = [-f_s/2, f_s/2]$. Let $f_p = f_s$, so that the signal obtained by the sampling after the low-pass filter contains all the characteristics of the received signal $x(t)$. According to Fourier Transform (FT) and related knowledge of signal, the spectrum of the signal obtained on channel g can be easily obtained:

$$\hat{X}_g(f) = \sum_{l=-\infty}^{\infty} c_{gl} X(f - l f_p) \quad s.t. \quad f \in F_s, \tag{5}$$

where c_{gl} is the coefficient of the Fourier series expansion of the signal $C_g(t)$, which is also the spectrum of $C_g(t)$. l is the coefficient of the original spectrum $X(f)$ shift. The spectrum of a wideband sparse signal is finite, so the number of shifts l is not infinite. Assuming $f_p = f_s$, the number of shifts L_0 can therefore be determined:

$$L_0 = \left\lceil \frac{(f_s + f_{nyq})}{2 f_p} \right\rceil - 1. \tag{6}$$

The relationship between the number of frequency band slices L and L_0 in the system model is $L = 2L_0 + 1$, which can be rewritten as:

$$\hat{X}_g(f) = \sum_{l=-L_0}^{L_0} c_{gl} X(f - l f_p), \quad f \in F_s. \tag{7}$$

The frequency spectrum of the compressed sampling result $\mathbf{Y}(n)$ is expressed as $\mathbf{Y}(f)$, which can be written in the form of a matrix:

$$\mathbf{Y}(f) = \Phi \mathbf{Z}(f). \tag{8}$$

$$\begin{pmatrix} y_1(f) \\ y_2(f) \\ \vdots \\ y_m(f) \end{pmatrix} = \begin{bmatrix} c_{1,-L_0} \cdots c_{1,L_0} \\ c_{2,-L_0} \cdots c_{2,L_0} \\ \vdots \\ c_{m,-L_0} \cdots c_{m,L_0} \end{bmatrix} \begin{pmatrix} X(f + L_0 f_p) \\ \vdots \\ X(f) \\ \vdots \\ X(f - L_0 f_p) \end{pmatrix} \tag{9}$$

where the observation matrix Φ is composed of a row vector c of weighting coefficients of m sampling channels, and the size is (m, L). $\mathbf{Z}(f)$ is a matrix composed of signals on L frequency bands, the size is (L, d), where d is the number of samples per channel.

C. Data Preprocessing

It can be seen from Eq. (8) that the estimated spectrum of the original signal $\tilde{Z}(n)$ can be obtained from the compressed sampling result $Y(n)$ as:

$$\tilde{\mathbf{Z}}(n) = \Phi^{\dagger}\mathbf{Y}(n), \tag{10}$$

where$\Phi^{\dagger}$ is the pseudo inverse of the observation matrix, from which the estimated original signal with noise and aliasing signal superimposed can be obtained.

PCA algorithm is a common method of data dimensionality reduction. It finds the direction with a large variance according to the data, projects the high-dimensional features to the direction with a large variance, and retains most of the information to complete the dimensionality reduction. When only the features of the main components are retained, the noise with a smaller variance will be filtered.

In other words, in noise reduction processing, the PCA algorithm is usually used to find the major component of the two-dimensional matrix, i.e., the part dominated by the signal energy, while discarding the minor component, i.e., the part dominated by the noise energy, which can then achieve the goal of denoising.

Perform MWC twice in succession, and the estimated signal obtained according to Eq. (10) is transformed into the size of $(L \times d, 1)$ and expanded into a matrix $\mathbf{D}$ of $(L \times d, 2)$ according to the column direction:

$$\mathbf{D} = \left[\tilde{\mathbf{Z}}_1(n), \tilde{\mathbf{Z}}_2(n)\right]. \tag{11}$$

Find the covariance matrix $\mathbf{C}$ of $\mathbf{D}$:

$$\mathbf{C} = \frac{1}{2-1}\mathbf{D}\mathbf{D}^{\mathbf{T}}. \tag{12}$$

Then the eigenvalues and eigenvectors of the covariance matrix $\mathbf{C}$ are obtained. Arrange the eigenvalues from large to small, and take the dimension that represents the feature of the data best, that is, the direction of the eigenvectors corresponding to the maximum eigenvalues. The original signal is projected to the new dimension to obtain the data. The horizontal axis of $\mathbf{D}$ represents features, and the vertical axis represents samples. The results of the two samplings are regarded as two dimensions of one signal. Both dimensions represent the same estimated signals, and both dimensions are affected by Gaussian white noise. Using PCA to reduce the two dimensions to one dimension can get the most representative feature of the original signal. Since the direction of the new coordinate axis found by PCA has the largest variance and is most representative of the original signal, the other dimension is discarded containing most of the white Gaussian noise, PCA removes part of the white Gaussian noise by reducing it to one dimension.

Data $\tilde{\mathbf{Z}}[n]$ after PCA denoising needs to use Discrete-time Fourier Transform (DTFT) to obtain spectrum estimation signal $\tilde{\mathbf{Z}}(f)$:

$$\tilde{\mathbf{Z}}(f) = DTFT\left(\tilde{\mathbf{Z}}[n]\right). \tag{13}$$

And then, we transform the DTFT result into the original size of $(L \times d, 1)$. After separating the real and imaginary parts of $\tilde{\mathbf{Z}}(f)$, we stack them as two signal features in the third dimension to become an input signal $\mathbf{I}$ of size $(L, d, 2)$.

D. Network Design

The proposed model consists of a main network and an adjustable network to achieve spectrum sensing. For the main network, we just hope to extract features of the input signal and the output of the adjustable network, so we stack several CNNs to achieve extracting features. However, for the adjustable network, inspired by Conditional Generative Adversarial Network (CGAN), which uses extra conditional input to adjust the network to generate specific output, we want the adjustable network to be a "dictionary like" module to generate the "modification signal" according to the extra conditional input. Then the features from the input signal and the features from the "modification signal" are concatenated together. After extracting features from concatenated information, a fully connected layer is used to classify the output value.

IV. Proposed Framework

A. Problem Formulation

Traditional compressed sensing methods use the results of MWC to select the most relevant element for the signal or residual by a greedy algorithm in each iteration. Then run multiple iterations to restore the support set:

$$j_k = \arg\max_j |\langle r_{k-1}, v_j \rangle|, \Lambda_k = \Lambda_{k-1} \cup \{j_k\}, \tag{14}$$

where j_k is the column index most associated with the residual vector r_{k-1} in the dictionary matrix $\mathbf{V}$, and Λ_k is the support set, which is updated through iteration.

The deep learning method also uses the MWC results to reconstruct the support set. Considering that in our proposed model, the frequency bands occupied by PUs are related to the time period, we add information about the time dimension. Therefore, we propose a system framework as shown in Fig. 2 to map the signal to the spectrum state. The one-hot encoding uses c state representation bits to represent z time periods, and each bit represents a time period. The bit value is 0 or 1, where 1 represents that the current state is valid. The pre-processed data $\mathbf{I}$ and the one-hot encoding $\mathbf{H} \in \Re^z$ of the time dimension obtained by artificially dividing the time period are sent to the TFF_aDCNN network, and the estimated signal is mapped to the frequency band state B_{state} using the TFF_aDCNN model ψ, which can be formulated as:

$$B_{state} = \psi(\mathbf{I}|\mathbf{H}, \boldsymbol{\theta}), \tag{15}$$

where $\boldsymbol{\theta}$ represents the parameters of the network. Meanwhile, deep learning transforms the original support set reconstruction problem into a multi-label classification problem. Only by

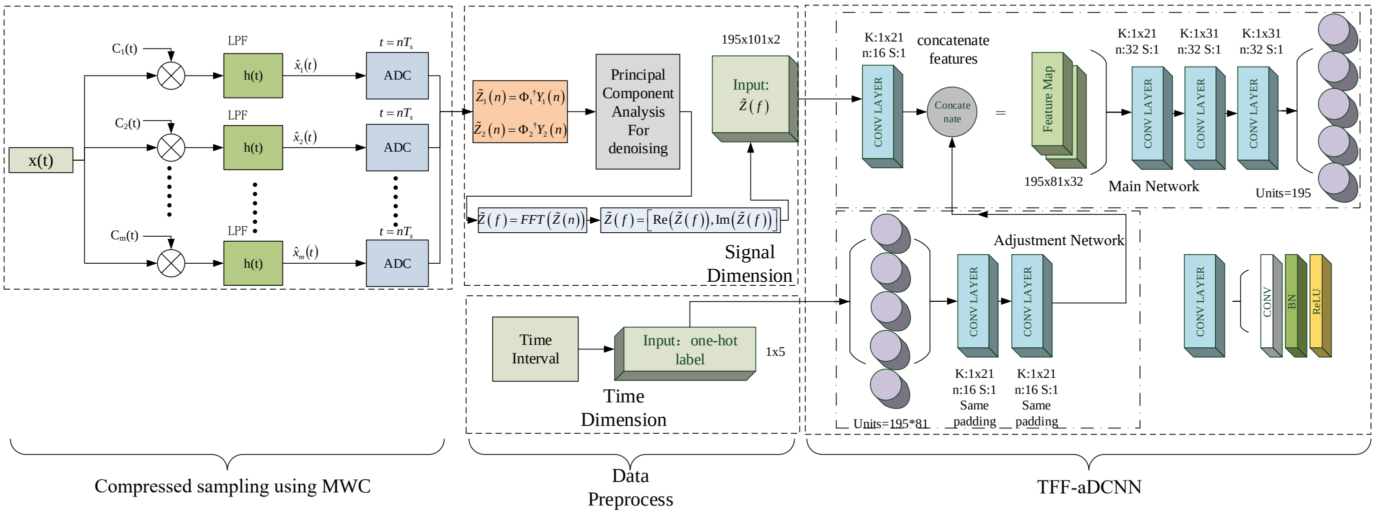


Fig. 2: System network framework.

judging the state of all frequency bands, the final frequency band state can be obtained. The multi-label classification problem continues to be transformed into multiple single-label binary classification problems. Finally, the *Sigmoid* activation function is used to solve the binary classification problem of judging L spectrum states.

B. Proposed TFF_aDCNN Model

A dual-input convolutional neural network is proposed. The two inputs of the network represent the signal dimension and the time dimension respectively. For the time dimension input, when a certain $\mathbf{H}$ is selected, the specific status bit is one, and the other status bits are zeros. The output of the fully connected layer in the adjustment network will only be affected by the current status bit being one, and the status bit of zero cannot affect the output of the fully connected network. During detection, the main network is regulated for spectrum sensing of specific data distribution by inputting specific one-hot encoding. Then the training set of the network can be defined as:

$$\chi = \left\{ \left[\left(\mathbf{I}^{(1)}, \mathbf{H}^{(1)} \right), \mathbf{y}^{(1)} \right], \cdots, \left[\left(\mathbf{I}^{(w)}, \mathbf{H}^{(w)} \right), \mathbf{y}^{(w)} \right] \right\}, \tag{16}$$

where w represents the number of samples in the training set. The input of each sample is composed of input signal $\mathbf{I} \in \Re^{L \times d \times 2}$, one-hot encoding $\mathbf{H} \in \Re^{1 \times z}$ and labels $\mathbf{y} \in \Re^{L}$.

Assume that the network parameters are $\boldsymbol{\theta}$, and the TFF_aDCNN model is ψ. We use supervised training, the output is activated by the *Sigmoid* function, and the value range is limited to (0,1). The cross-entropy loss function ℓ is:

$$\begin{aligned} \ell &= \frac{1}{w} \sum_{n=1}^{n=w} \sum_{i=1}^{n=L} \ell_i^n \\ &= \frac{1}{w} \sum_{n=1}^{n=w} \sum_{i=1}^{i=L} \left[y_i^{(n)} \cdot \log \left(\psi_i \left(\mathbf{I}^{(n)} \mid \mathbf{H}^{(n)}, \boldsymbol{\theta} \right) \right) \right. \\ &\left. + \left(1 - y_i^{(n)} \right) \cdot \log \left(1 - \psi_i \left(\mathbf{I}^{(n)} \mid \mathbf{H}^{(n)}, \boldsymbol{\theta} \right) \right) \right], \end{aligned} \tag{17}$$

where $y_i^{(n)}$ is the value of the i-th spectrum slice in the n-th training label, with band occupied as 1 and band idle as 0. $\psi_i \left(\mathbf{I}^{(n)} | \mathbf{H}^{(n)}, \boldsymbol{\theta} \right)$ represents the output of the network to the i-th spectrum slice state of the n-th training sample, and predicts the probability of band occupied in the form of 0 to 1. The training of the TFF_aDCNN network is mainly driven by data. During training, the TFF_aDCNN network is optimized by the Adaptive Moment Estimation (Adam) optimizer.

When the one-hot encoding representing a certain time period is input to the adjustment network, the data with the distribution characteristics of the current time period are input to the main network at the same time. After training, the loss of the network under the current input combination (the one-hot encoding of the current time period and the data that obeys the current time period distribution characteristics) is reduced. Under the current one-hot encoding input, the main network passively learns the distribution characteristics of the input data. When a conditional input $\mathbf{H}$ is selected for training, the adjustment network is able to update the parameters connected to the status bit 1 of one-hot encoding by backpropagation. A large amount of training makes the main network more sensitive to the data distribution in the current time period when the one-hot encoding of the current time period is input into the adjustment network.

After training, by inputting the one-hot encoding representing different time periods into the adjustment network, the focus of the TFF_aDCNN main network detection is actively adjusted. Therefore, under a low SNR, the network can still perform spectrum sensing that focuses on the current data distribution by inputting the one-hot encoding to achieve performance improvement. The training of the network can be regarded as an optimization problem:

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \ell \left(\chi, \boldsymbol{\theta} \right). \tag{18}$$

The algorithm flow is shown in **ALGORITHM 1**.

Algorithm 1 **Training and application of TFF_aDCNN.**

Input: Training dataset $\chi = \left\{\left[\left(\mathbf{I}^{(1)},\mathbf{H}^{(1)}\right),\mathbf{y}^{(1)}\right],\cdots,\left[\left(\mathbf{I}^{(w_1)},\mathbf{H}^{(w_1)}\right),\mathbf{y}^{(w_1)}\right]\right\}$; Test dataset $\tilde{\chi} = \left\{\left(\tilde{\mathbf{I}}^{(1)},\tilde{\mathbf{H}}^{(1)}\right),\left(\tilde{\mathbf{I}}^{(2)},\tilde{\mathbf{H}}^{(2)}\right),\cdots,\left(\tilde{\mathbf{I}}^{(w_2)},\tilde{\mathbf{H}}^{(w_2)}\right)\right\}$; Number of epochs E_{ep}; Batch Size E_B; Learning rates α_l.
Output: Sensing result B_{state}

Initialization:
Randomly initialize parameter $\boldsymbol{\theta}$: $\boldsymbol{\theta} \leftarrow random$.
Training:
for $i = 1, 2, \cdots, E_{ep}$ **do**
  for $j = 1, 2, \cdots, \lfloor W/E_B \rfloor$ **do**
    Choose E_B data from training dataset χ without replacement
    Calculate gradient:$grad \leftarrow \nabla_\theta \ell\left(\chi, \boldsymbol{\theta}\right)$
    Update the $\boldsymbol{\theta}$ by Adam with learning rate α_l: $\boldsymbol{\theta} \leftarrow Adam\left(\alpha_l, grad, \boldsymbol{\theta}\right)$
Application:
begin
Get TFF_aDCNN's parameters $\boldsymbol{\theta}$
Input test dataset $\tilde{\chi}$ into TFF_aDCNN
Get sensing result $B_{state} = \psi\left(\tilde{\chi}, \boldsymbol{\theta}\right)$
return result

C. Structure of TFF_aDCNN

As shown in Fig. 2, the proposed network model consists of two parts: the main network and the adjustment network. The main network is composed of a convolution part and a fully connected part. The convolution part includes four convolution units. Each convolution unit is composed of a convolutional layer with strides 1, a batch normalization layer and a ReLU layer. The fully connected part is composed of a fully connected neural network of L neurons, and uses the *Sigmoid* activation function to limit the output range to (0,1) to represent the frequency band state.

Connect the input data $\mathbf{I}$ of the main network with the convolution part. After the data passes through the first convolution unit with a $(1, 21)$ filter $\mathbf{K}_1$, 16 channels, the effective information is extracted and the first feature map is obtained:

$$\mathbf{R}^1 = \left[\mathbf{R}_1^1, \cdots, \mathbf{R}_i^1, \cdots, \mathbf{R}_{16}^1\right], \mathbf{R}_i^1 \in \Re^{a \times b}. \tag{19}$$

Since the filter is set to 16 channels, the output has 16 matrices from $\mathbf{R}_1^1$ to $\mathbf{R_{16}^1}$, and each matrix is as follows:

$$\mathbf{R}_i^1 = \begin{pmatrix} R_{1,1}^i & \cdots & R_{1,d-20}^i \\ \vdots & R_{a,b}^i & \vdots \\ R_{L,1}^i & \cdots & R_{L,d-20}^i \end{pmatrix}. \tag{20}$$

The value in the matrix is obtained by the convolution of the input matrix and the convolution filter:

$$\begin{aligned} R_{a,b} &= \xi(\mathbf{I} * \mathbf{K}_1 + \mathbf{W})_{a,b} \\ &= \xi\left(\sum_{i=1}^{1}\sum_{j=1}^{21}\sum_{l=1}^{2}\underbrace{I_{a+i-1,b+j-1,l}K_{i,j,l}}_{\mathbf{I}*\mathbf{K}_1} + \underbrace{w_{a,b,l}}_{\mathbf{W}}\right), \end{aligned} \tag{21}$$

where $\xi\left(\right)$ represents the ReLU activation function. With the subscripts a and b, it represents the value of a row and b column in a certain channel of the feature map, $\mathbf{W}$ is bias matrix. Because the size of the input data $\mathbf{I}$ is $(L, d, 2)$ and L is the number of spectrum slices, the first dimension of the input data has the structure and position information of the spectrum slices. In order to ensure that the information of the data is not lost, the first dimension of the filter is 1. In the output feature map of each subsequent convolution unit, the first dimension remains L unchanged.

The adjustment network consists of a fully-connected part and a convolution part. Considering that 16-channel feature maps with the same dimension and shape are required for feature concatenation, the fully connected part consists of a fully connected layer with $L \times (d - 20)$ neurons. The parameters are not only set to satisfy the subsequent feature concatenation, but also map the low-dimensional temporal information to higher dimensions, increase the learnable parameters of the network, and improve the learning ability of the network.

The convolution part consists of two convolution units, each consisting of a convolutional layer with a $(1, 21, 16)$ filter, the same padding, a batch normalization layer and an activation layer using the *ReLU* activation function. The one-hot encoding representing the time dimension information is connected to the fully connected part of the adjustment network. The data after the added dimension cannot be directly input to the convolution part, so the fully connected output data is reshaped into a two-dimensional matrix, and then connected to the convolutional layers. The convolutional layers finally output a feature map with temporal information $\mathbf{R}^2$, which will be concatenated with the feature map $\mathbf{R}^1$ containing the input data information on the channel dimension to form a new feature map with a channel number of 32:

$$\mathbf{R}_{concatenate} = concatenate\left[\underbrace{\mathbf{R}^1}_{signal-part}, \underbrace{\mathbf{R}^2}_{time-part}\right], \tag{22}$$

where *concatenate*[] denotes concatenation on channel dimension. The concatenated feature map $\mathbf{R}_{concatenate}$ is input to the rest of the main network. The feature map output by the convolution part is connected to the fully connected part through a flatten layer. The fully connected part obtains the scores of L spectrum slices through the *Sigmoid* activation function. In order not to lose information, no pooling layer is used in the convolution unit, and in order to reduce the parameters, the convolution filter and the number of channels are designed according to input data. The final layer of convolution outputs a feature map of size $(L, 1, 32)$.

D. Transfer Learning Description of the Base Model

According to the literature on spectrum measurement [35]–[39], we know that the spectrum usage patterns of different cities are different, and even the same frequency band has different activity patterns in different regions. Therefore, it is difficult to have a general model that can learn the regularity of spectrum usage everywhere.

The model obtained by training TFF_aDCNN is to explore the effectiveness of the proposed spectrum sensing framework and use it as a pre-trained base model. Obviously, this base model cannot be used as a general and specific deep learning model, and it cannot cope with wideband spectrum sensing tasks in all electromagnetic environments.

In practical applications, for a specific task, spectrum measurement is first performed on the spot for at least one day, and the distribution law of the authorized spectrum is found according to the results of the spectrum measurement. Then, the distribution is modeled to produce a sizable amount of training set data through simulation. Next, the training set is used for transfer learning based on the pre-trained base model. In the process of transfer learning, we freeze the main network in the pre-trained base model, and only activate the adjustable auxiliary sub-network, so that we can quickly train to obtain a new and practical model suitable for new scenarios, that is, we can quickly adapt to local spectrum usage patterns.

V. Performance Evaluation

In this section, we consider a single SU to realize spectrum sensing as mentioned in *System Modeling*. Because of the hardware limitation, our simulations are conducted on a computer including signal simulation, MWC downsampling, data preprocessing and model evaluation. To evaluate the performance of our proposed model, a Gaussian channel is used. Additive White Gaussian Noise (AWGN) is added to PUs signal to simulate the contaminated signal received by the receiver.

A. Parameter Settings

1) MWC Parameter Settings and Dataset: Table I lists the values of the parameters used in the MWC experiments and their meanings. According to the origin MWC paper [14], they generate PUs' signals by formula:

$$p(t) = \sum_{i=1}^{N} \sqrt{E_i B_{\max}} \operatorname{sinc}\left(B_{\max}(t-\tau_i)\right) \cos\left(2\pi f_{ci}(t-\tau_i)\right), \tag{23}$$

where $\operatorname{sinc}(x) = \sin(\pi x)/\pi x$, and other variables can be found in Table I. We also use this formula to simulate the PUs' signals and AWGN is added. To simplify the experiment, the day is divided into $z = 5$ time periods and we randomly generate the PUs' signal frequency f_{ci} obeying $z = 5$ specific Gaussian distribution with mean μ(considering the max frequency is 1) and variance σ_i. When we get the frequency f_{ci}, the occupied bands' index is represented as one-hot training label. After that, using MWC to implement compressed sampling with parameters in Table I.

2) Data Preprocessing: $\tilde{\mathbf{Z}}(n) = \Phi^{\dagger}\mathbf{Y}(n)$ is used to get the estimated data. After two successive estimated data are obtained, PCA is used to reduce dimension to get one estimated data like in Section III *Data Preprocessing*. The scikit-learn package is used for implementing PCA, and we reserve one main component to realize dimension reduction, which means setting the parameter n_components equal to 1 in function PCA(). Since spectrum information is more intuitive, we use FFT to get signals' spectrum as a training dataset.

The software environment for the simulation experiments is as follows: *64-bit Win10 operating system, MATLAB 2018a, TensorFlow 1.15.5, Cuda11.4. The hardware environment is as follows: 6-core Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz, RAM 15GB, NVIDIA GeForce GTX 1080 Ti.*

B. Support Set Reconstruction Performance under Different SNRs

The comparison of model performance with different SNRs is performed. The data are denoised by PCA first. The SNR of the received signal $x(t)$ is $SNR = \{-10, -8, -6, -4, -2, 0\}$dB, and other conditions are the same as the simulation settings. Under each SNR condition, the training set has 8,500 pieces of data, and the test set has 1,500 pieces of data. Each set is divided into 5 groups, and each group of data obeys a specific distribution. Experiments are conducted using the proposed TFF_aDCNN network, the TFF_aDCNN network without an adjustment network, the SwSOMP [17], the SOMP [18], and DWLSS [29].

The ablation experimental results are as shown in Fig. 3. When $m = 25$, $N = 6$, $SNR = \{-10, -8, -6, -4, -2, 0\}$dB, the average reconstruction probability of the proposed model is 3.5% higher than that of the TFF_aDCNN model without the adjustment network. The DLWSS [29] performance is similar to that of TFF_aDCNN without adjustment network. Compared with traditional methods such as SOMP and SwSOMP, the proposed model has an average improvement of 65% and 69%, respectively. The results also prove that the TFF_aDCNN can learn the data characteristics of each time period, so that under low SNR, the model can also use the input one-hot encoding to obtain a higher support set reconstruction probability. Meanwhile, we set the standard deviation condition of the data distribution to $\sigma_2 = [0.04, 0.04, 0.04, 0.04]$ and conduct the support set reconstruction probability experiment with $m = 20$ and $N = 6$. From Fig. 3, when the data standard deviation is set to σ_2, the network performance is better, which indicates that the more obvious the usage pattern of the main user in the current time period, the more the network can find the pattern and learn the data distribution characteristics.

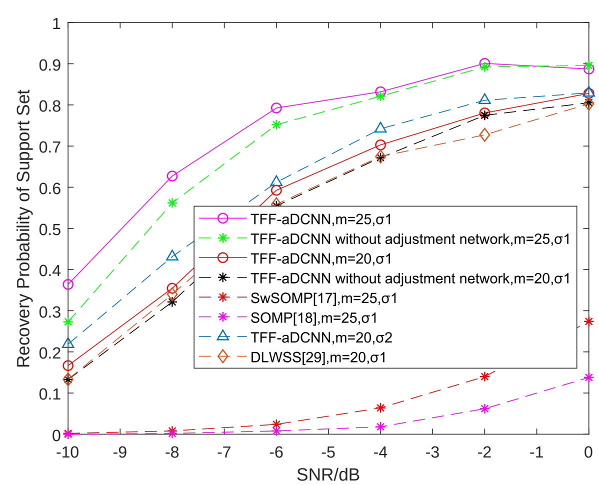


Fig. 3: Comparison of recovery probabilities of support set under different channel conditions.

TABLE I: Settings of simulation parameters.

Symbols	Value	Meanings
N	6	Number of sub-bands with energy
m	[15, 35]	Number of parallel sampling channels
E_i	{1, 2, 3}	Energy coefficient of the ith sub-band
$B_{\max}$	50MHz	Maximal width within sub-bands
τ_i	{0.4, 0.7, 0.2}	Time offset of the ith sub-band
f_{nyq}	10GHz	Nyquist rate
L	195	Aliasing rate, or the spectrum slice number
f_p	51.28MHz	Spectral slice width, $f_p = f_{nyq}/L$
f_s	51.28MHz	Sampling rate at each channel, $f_s = qf_p$, with odd q
μ	{0.2, 0.4, 0.6, 0.8}	The mean of normal distribution
σ_1	{0.1, 0.1, 0.1, 0.1}	The variance of normal distribution
σ_2	{0.04, 0.04, 0.04, 0.04}	The variance of normal distribution
d	101	The number of downsample points
$w(t)$	$w(t)$	Gaussian White Noise

PCA denoising makes the TFF_aDCNN network have a great performance improvement. As shown in Fig. 4, we compared the networks' support set reconstruction probability performance with and without PCA processing data in $m = 20$, $N = 6$, σ_1. When the SNR is very low, PCA cannot effectively remove noise. However, as the SNR increases, TFF_aDCNN network performs better with the data after PCA denoising. Compared with using data that have not been processed by PCA, denoising can improve the performance of the network's support set reconstruction, especially when $SNR = -2$dB, the support set reconstruction rate is increased by 24.66%.

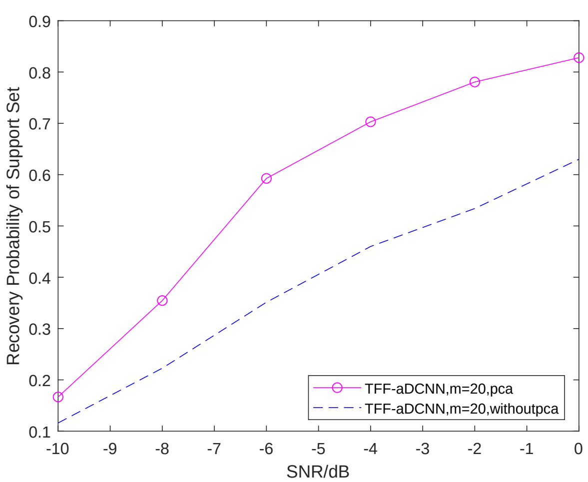


Fig. 4: Comparison of recovery probabilities of support set between the data with and without PCA denoising ($m = 20$).

Then, the estimated signals without PCA denoising are fed into the TFF_aDCNN network and the DLWSS [29] network for support set reconstruction performance comparison. The data without PCA denoising contains more Gaussian white noise, which is more able to show the performance of the model itself for low SNR conditions. Since DLWSS [29] is also designed for specific MWC parameters, we adapt the convolution filter size in the DLWSS [29] network to our dataset and reproduce the network in line with the idea in the original paper [29]. At $N = 6$, $m = 20$, σ_1 and $SNR = \{-10, -8, -6, -4, -2, 0\}$dB, the experimental results are shown in the Fig. 5. The experiment proves, without PCA denoising, our proposed TFF_aDCNN network support set reconstruction probability is on average 9.73% higher than the DLWSS [29] at $SNR = \{-10, -8, -6, -4, -2, 0\}$dB, just in terms of the performance of the network itself in a low SNR environment. The data after PCA denoising improves the SNR, while the data without PCA processing can better reflect the performance of the network itself, which strongly demonstrates that the TFF_aDCNN network successfully fuses the time of the inputs. It can also show that the TFF_aDCNN is able to learn the data distribution characteristics at each $\mathbf{H}$ input and use the learned data distribution to assist the network for spectrum sensing under a low SNR.

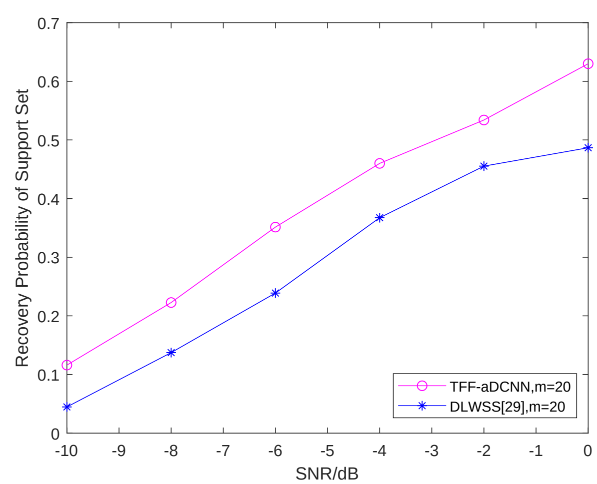


Fig. 5: Comparison of recovery probabilities of support set between DLWSS [29] and TFF_aDCNN by using data without PCA processing ($m = 20$ and $N = 6$).

C. Performance under Different Numbers of Parallel Channels

The more channels the MWC has, the better the model performs. We investigate the performance of the TFF_aDCNN network when the number of channels is taken in steps of 2 between the interval $[15, 35]$ for $N = 6$, σ_1, $SNR = -6$dB and $SNR = -8$dB, using the data after PCA denoising and the data without PCA.

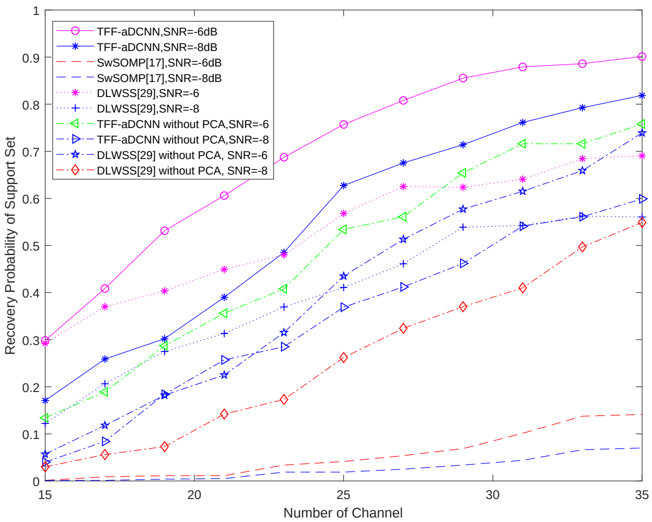


Fig. 6: Support set reconstruction probability of TFF_aDCNN with different numbers of channels.

After comparison, it is found that the TFF_aDCNN network is able to obtain a high support set reconstruction probability under the conditions of low channel number and low SNR. While, the conventional method SwSOMP reconstruction accuracy is less than 20%. From Fig. 6, the TFF_aDCNN reconstruction probability reaches more than 90% under the $SNR = -6$dB, $N = 35$, which is a high probability reconstruction performance. This reduces the hardware requirements(MWC parallel channels) for WSS, making the network more suitable for deployment, less hardware consumption, and greener.

Importantly, as shown in Fig. 6, the TFF_aDCNN network and the DLWSS network [29] are also compared with respect to the reconstruction accuracy of the support set. When the number of parallel channels is close to the theoretical lower limit (i.e., $m = 15$ and the theoretical limit is $2N + 1 = 13$ [14], [41]), the performance improvement of TFF_aDCNN compared to DLWSS is not very obvious. However, as the number of channels m is raised (i.e., $m > 15$), the performance of TFF_aDCNN has a very large improvement. When $m > 27$ and $SNR = -6$dB, the performance improvement of TFF_aDCNN tends to level off compared to DLWSS, and its average performance improvement can reach 22.1%.

Also, using the data without PCA denoising, the performance is decreased for both TFF_aDCNN and DLWSS. However, no matter what data we use(with or without PCA), our proposed network outperform DLWSS under each channel.

Therefore, the TFF_aDCNN model has better reconfiguration performance than the same type of models and conventional optimization models at low SNR and low hardware complexity.

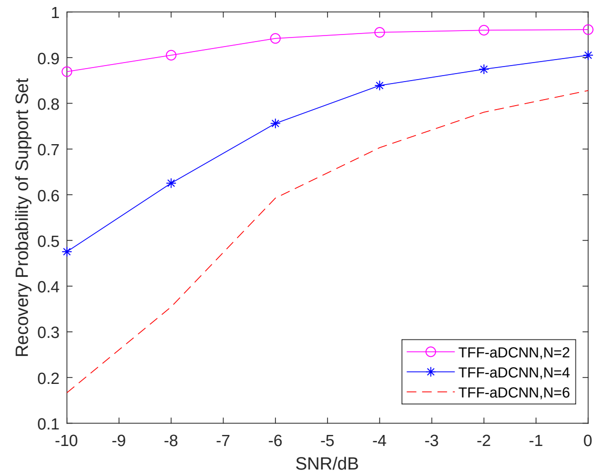


Fig. 7: The performance of the network under different SNRs and different signal frequency bands.

D. Performance under Different Signal Frequency Bands

The influence of different N on the support set reconstruction ability of TFF_aDCNN network support set is investigated. Considering the symmetrical frequency bands, the value of the number of frequency bands must be an even number, so we create a new dataset with N=2, 4, and 6, the dataset size is the same as that in Subsection V-B. We use PCA to denoise the signal. The value of SNR is $\{-10, -8, -6, -4, -2, 0\}$dB, the number of parallel MWC channels is $m = 20$, and other parameters are the same as in the simulation settings. As shown in Fig. 7, as the number of signal frequency bands increases, the performance of the network under low SNR is severely degraded.

When $N = 2$ and the number of channels $m = 20$, the support set reconstruction rate is still close to 90% when $SNR = -10$dB. When $N = 4$ and the number of channels $m = 20$, the performance drops more severely than when $N = 2$, but the support set reconstruction rate can still reach 90% when $SNR = 0$dB. When $N = 6$ and the number of channels $m = 20$, the performance is further degraded, and it is impossible to obtain a high reconstruction probability lower than $SNR = 0$dB. The number of signal frequency bands greatly affects the performance of TFF_aDCNN.

E. TFF_aDCNN Structure Discussion

In order to discuss the rationality of the network structure, we compare the structure of two different adjustment networks and compare the performance of the two structures in terms of reconstruction ability, we use the data after PCA denoising with the number of parallel channels $m = 25$, σ_1 and $N = 6$.

The experimental results are shown in Table II, from which we can find that when using two layers CNN in our proposed TFF_aDCNN structure as the adjustment network, it has advantages compared with the adjustment network using only

TABLE II: Performance of the two adjustment network structures (σ_1, $m = 25$ and $N = 6$).

Structure	-10	-8	-6	-4	-2	0
2-layer-CNN	36.4%	62.73%	79.26%	83.20%	90.13%	88.73%
1-layer-CNN	33.6%	57.00%	77.20%	82.06%	84.60%	85.06%

TABLE III: Performance of three kinds of models (σ_1, $m = 20$ and $N = 6$).

Models	-10	-8	-6	-4	-2	0
Pre-trained base model	23.7%	51.9%	59.8%	62.2%	72.2	77.8%
Training from scratch (norm+norm)	22.3%	47.6%	60.7%	62.2%	69.1%	78.9%
Fine-tuned model (norm+norm)	20.4%	47.0%	56.5%	60.4%	71.8%	77.6%

one layer CNN. Considering the number of parameters and the support set reconstruction probability of the frequency band, our proposed TFF_aDCNN network has an appropriate number of parameters and an advantageous performance.

F. TFF_aDCNN Classification Capability

For the classification ability of TFF_aDCNN, we use Receiver Operating Characteristic (ROC) curve and confusion matrix to evaluate it. The confusion matrix is a matrix that reflects the classification ability of the model, and since we use a binary classification approach to determine the spectrum state, we get a matrix of size (2,2), as shown in Fig. 12. The vertical 0 and 1 denote the true label, and the horizontal 0 and 1 denote the predicted label by the model, then the significance of the four regions is defined as: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

Since the frequency spectrum is sparse, the positive and negative samples are out of proportion. Recall and precision are needed to describe the model's ability to judge positive examples.

$$Recall = TP/(TP + FN), \tag{24}$$

$$Precision = TP/(TP + FP), \tag{25}$$

where *Recall* in Eq. (24) reflects the sensitivity of the model to positive cases, while *Precision* in Eq. (25) reflects the accuracy of the model in judging positive cases. However, the precision rate is not as important as the recall rate in spectrum sensing. When the false alarm rate of the network is high, the reconstructed support set is likely to contain the true support set, which also does not affect the use of the network by the PUs. Conversely, when the recall rate is low, the reconstructed support set lacks the truly occupied frequency bands, which will greatly affect the normal communication of the PUs.

The dataset in Subsection V-B is used to perform the recall and precision calculations with $N = 6$ and σ_1. Looking at the recall images in Fig. 8, we can find that the recall rates of the two models are similar when the $SNR < -4$dB, while our proposed TFF_aDCNN network can show better performance at lower SNRs. It indicates that the TFF_aDCNN network can be more sensitive to band occupancy at low SNRs and can capture the band being occupied with a higher probability at low SNRs.

The TFF_aDCNN can judge PUs' state more accurately for the number of channels $m = 25$ according to the precision rate figure. With $m = 20$, the precision rate performs worse compared to the TFF_aDCNN without the adjustment network, however, TFF_aDCNN is able to obtain a higher support set reconstruction probability at a worse precision rate, indicating that after reducing the channel number, TFF_aDCNN is still able to use the learned relationship between the band occupancy characteristic and the input $\mathbf{H}$ to assist the main network in determining the spectrum state correctly. The experiments demonstrate that the existence of the adjustment network is necessary.

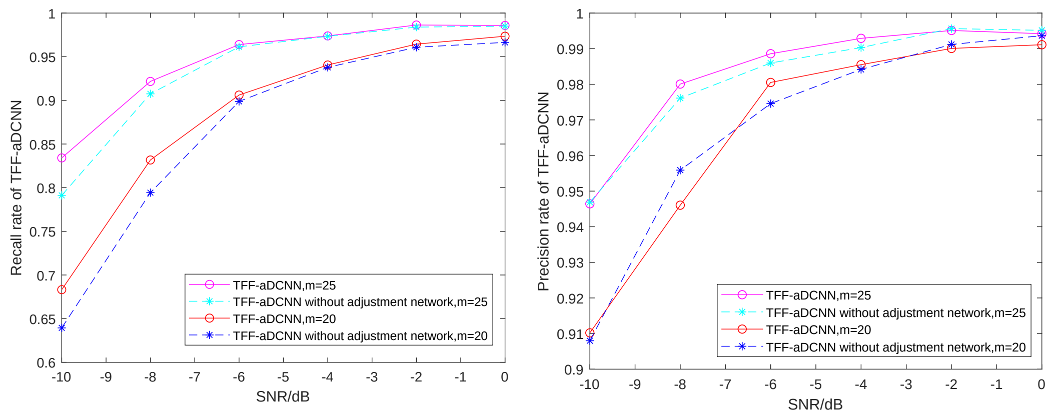
Since the author of DCSS-GAN did not release their code, we reproduce the model and conduct the experiment with our model. A new dataset is generated because of the model difference. The dataset consists of 0dB and 20dB data with σ_1, $m = 20$ and $N = 6$, processed by PCA. The results in Table IV shows that our model performs better in recovering probability, precision and recall.

The ROC can reflect the classification ability of the model. We treat the spectrum sensing problem as a binary classification problem for each frequency band, so using the ROC curve shown in Fig. 9 can reflect the classification ability of the model. We use the data under $SNR = -10$dB, $m = 20$ and $N = 6$ for the classification test and the first four-time periods obey a normal distribution with standard deviation $\sigma_1 = [0.1, 0.1, 0.1, 0.1]$. The results show that our network has improved band state classification ability compared to the TFF_aDCNN without the adjustment network. We also use reproduced DCSS-GAN model to present in ROC curve in Figs. 10 and 11. It is found that 20dB is so high that our model almost can classify each band state.

Through the experiment of the confusion matrix in Fig. 12, we can clearly find that the TFF_aDCNN network can predict more positive classes correctly under low SNR, and the recall rate and precision rate are higher than the TFF_aDCNN without the adjustment network. The TFF_aDCNN model correctly predicted 738 more positive cases than the TFF_aDCNN model without the adjustment network.

G. Model Transferability

To validate the transferability of our proposed pre-trained base model, we produced a new dataset for a new electro-



(a) Recall curve.

(b) Precision curve.

Fig. 8: The recall and precision performance of TFF_aDCNN model for the cases of $m = 20$ and $m = 25$.

TABLE IV: Spectrum recovery performance of TFF_aDCNN and DCSS-GAN using new made dataset.

channel	Metrics	SNR	TFF_aDCNN	DCSS-GAN
m=15	Recovery probability	0dB	**44.6%**	1.9%
		10dB	**76.2%**	9.4%
		20dB	**81.4%**	10.9%
	Precision	0dB	**96.96%**	77.69%
		10dB	**99.08%**	87.17%
		20dB	**99.32%**	88.14%
	Recall	0dB	**89.47%**	50.72%
		10dB	**96.31%**	77.64%
		20dB	**97.10%**	79.72%
m=20	Recovery probability	0dB	**81.2%**	6.1%
		10dB	**89.7%**	22.2%
		20dB	**93.4%**	25.4%
	Precision	0dB	**96.97%**	79.62%
		10dB	**99.67%**	91.85%
		20dB	**99.11%**	90.36%
	Recall	0dB	**99.61%**	83.70%
		10dB	**98.41%**	89.21%
		20dB	**99.77%**	90.16%
m=25	Recovery probability	0dB	**85.1%**	11.4%
		10dB	**92.7%**	23.0%
		20dB	**94.1%**	30.6%
	Precision	0dB	**99.74%**	89.29%
		10dB	**99.72%**	90.8%
		20dB	**99.84%**	94.56%
	Recall	0dB	**97.54%**	78.22%
		10dB	**99.09%**	88.03%
		20dB	**99.16%**	89.15%

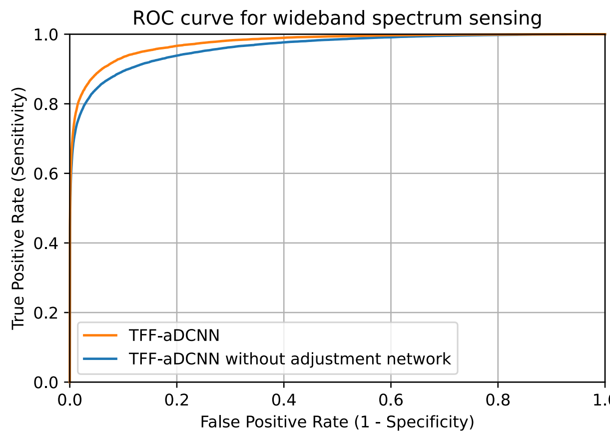


Fig. 9: ROC curve reflecting the classification ability of the model ($SNR = -10$dB, σ_1, $m = 20$ and $N = 6$).

magnetic environment. The parameters were set as $SNR = [-10, -8, -6, -4, -2, 0]$dB, $m = 20$ and $N = 6$.

The pre-trained base model assumes that the sub-bands' activity in each time period exhibits normal distributed. By changing the distribution of the spectrum, we also simulated an unfamiliar electromagnetic environment, that is, we changed from a single normal distribution for each time period to two normal distributions for each time. After loading the pre-trained base model, the trained main network is frozen and only the model parameters of the adjustable network are trained. Therefore, we can obtain a wideband spectrum sensing fine-tuned model that can be adapted to the new electromagnetic distribution environment by transfer learning.

Table III shows that the pre-trained base model performs well in a single-distributed electromagnetic environment. For sensing tasks in a complex electromagnetic environment, we use the pre-trained base model for transfer learning, during which only the adjustable sub-network is retrained. From Table III, we can know that the fine-tuned model also has good performance. Compared with the fully trained model from scratch, we can conclude that the fine-tuned model can adapt to different electromagnetic environments. Meanwhile, the pre-trained base model has the ability of transfer learning.

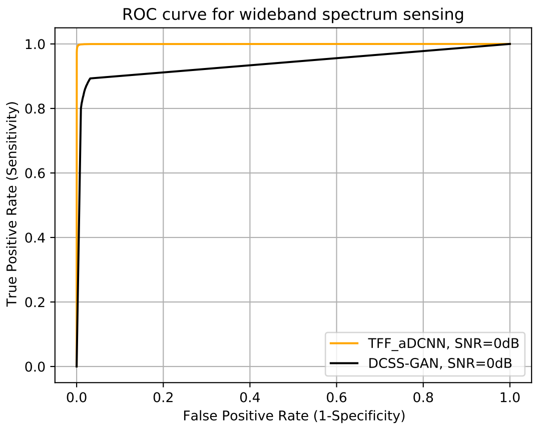


Fig. 10: ROC curve reflecting the classification ability of the model ($SNR = 0$dB, σ_1, $m = 20$ and $N = 6$).

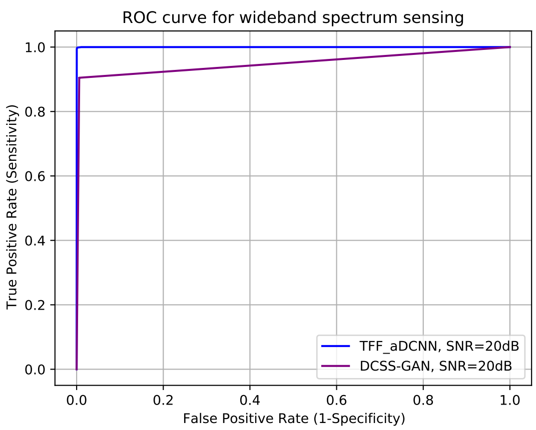


Fig. 11: ROC curve reflecting the classification ability of the model ($SNR = 20$dB, σ_1, $m = 20$ and $N = 6$).

H. Complexity and Time Overhead

Considering that the spectrum sensing task requires real-time performance, we conduct a complexity analysis of the model. The model is mainly composed of CNN and a fully connected layer, where the complexity of CNN is:

$$O\left(M_h \cdot M_w \cdot K_h \cdot K_w \cdot C_{in} \cdot C_{out}\right), \tag{26}$$

where M_h and M_w represent the height and width of the output feature map, respectively, K_h and K_w represent the height and width of the kernel, respectively, and C_{in} and C_{out} are the number of input channels and output channels, respectively.

The complexity of a fully connected layer is:

$$O\left(D_{in} \cdot D_{out}\right) \tag{27}$$

where D_{in} is the input dimension and D_{out} is the output dimension.

TABLE V: Time cost analysis with $m = 25$ and 100 iterations.

N	MWC	Data Preprocess	TFF_aDCNN
2	0.3740s	0.0226s	0.0702s
4	0.3698s	0.0212s	0.0701s
6	0.3689s	0.0328s	0.0701s

TABLE VI: Time cost analysis for 50 epochs training.

Fully trained model from scratch	Fine-tuned model
571.2s	471.9s

In our model, $M_h = L$, $K_h = 1$. The complexity of the adjustable network is expressed as:

$$O\left(\begin{array}{l}\underbrace{z}_{D_{in}}\underbrace{L\left(d-K_w+1\right)}_{D_{out}}+\underbrace{L}_{M_h}\underbrace{\left(d-K_w+1\right)}_{M_w}K_w\cdot\underbrace{2}_{C_{in}}C_{out}\\+\underbrace{L}_{M_h}\underbrace{\left(d-K_w+1\right)}_{M_w}K_w\underbrace{C_{out}^2}_{C_{in},C_{out}}\end{array}\right) \tag{28}$$

where z is the number of time periods, and d is the number of sample points by MWC.

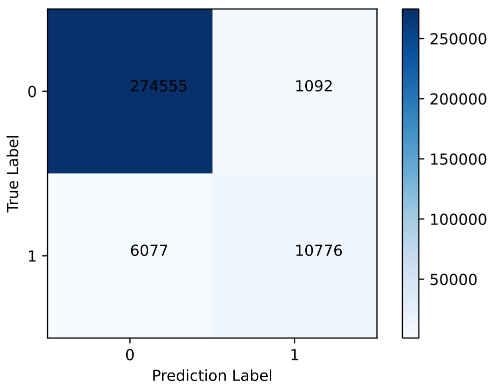
The complexity of the main network is expressed as:

$$O\left(\begin{array}{l}2L\left(d-K_w+1\right)K_wC_{out}+L\left(d-2K_w+2\right)K_wC_{out}^2+\\L\left(d-3K_w-7\right)\left(K_w+10\right)C_{out}^2+\\\ L\left(d-4K_w-16\right)\left(K_w+10\right)C_{out}^2+\\\ 2L^2\left(d-4K_w-16\right)C_{out}\end{array}\right) \tag{29}$$

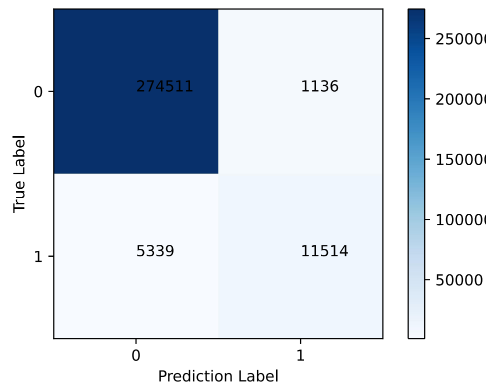
The overhead of the whole system framework is simply estimated. The whole system can be divided into an MWC sampling stage, a data preprocessing stage and a TFF_aDCNN network stage. We use MWC to sample a signal 100 times and calculate the average time. Then, record the time of data preprocessing. Finally, the trained model is used for predicting and the time is recorded. The tensor of the prediction results is obtained.

Through experiment results in Table V, we find that the time cost of MWC is the highest, followed by TFF_aDCNN network, and finally, the data preprocessing part. In order to denoise the signal during data preprocessing, we use two MWC samples, which consume some time, but the time is traded for a higher probability of support set reconstruction. It is worth noting that the IEEE802.22 standard has proposed two sensing methods, fast sensing and fine sensing for the PUs' service. The spectrum uses a band space that varies on a larger time scale, and the real-time requirements of the sensing cycle are not as stringent.

It can be seen from Table VI that the transfer learning way based on the pre-trained base model can obtain a faster training speed, which also verifies the correctness of the theoretical analysis of the TFF_aDCNN model. Therefore, in practical application deployment, the grounded wideband spectrum sensing model can be obtained at a fast speed.



(a) TFF_aDCNN without the adjustment network.



(b) TFF_aDCNN.

Fig. 12: Confusion matrix of the TFF_aDCNN ($SNR = -10$dB, $m = 20$ and σ_1).

VI. Conclusion and Future Work

In this paper, we proposed a novel intelligent WSS framework based on TFF_aDCNN model. TFF_aDCNN can learn the data distribution features in different time periods during training and use these features as auxiliary information to assist in determining the current spectrum state. By training TFF_aDCNN in this framework we can obtain a pre-trained base model under a simple distribution pattern. Then, in the actual specific application scenario, transfer learning was performed on the basis of the base model to obtain an intelligent wideband spectrum sensing model, that is, a fine-tuned model. The fine-tuned model obtained after transfer learning has adapted to the actual electromagnetic environment. Experimental results showed the proposed sensing framework can achieve better sensing performance than other models under very low SNR and a few sampling channels.

In the future, we can use cooperative spectrum sensing based on TFF_aDCNN to obtain better performance. If we consider the space and time relationship and combine the frequency relationship, we can further improve the performance of the single-node WSS network. At present, Transformer performs very well in the field of natural language processing. Many works use the multi-head attention mechanism, position vector embedding mechanism and multilayer perceptron of the Transformer model to replace the LSTM structure and the CNN structure. The transformer model has a powerful ability to extract the potential relationship among all inputs, which may improve the support set reconstruction probability of WSS based on the time-frequency-related model proposed in this paper.

References

[1] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless personal communications*, 58(1):49–69, 2011.

[2] Ping Yang, Yue Xiao, Ming Xiao, and Shaoqian Li. 6g wireless communications: Vision and potential techniques. *IEEE Network*, 33(4):70–75, 2019.

[3] Baiqing Zong, Chen Fan, Xiyu Wang, Xiangyang Duan, Baojie Wang, and Jianwei Wang. 6g technologies: Key drivers, core requirements, system architectures, and enabling technologies. *IEEE Vehicular Technology Magazine*, 14(3):18–27, 2019.

[4] Amir Ghasemi and Elvino S Sousa. Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs. *IEEE Communications magazine*, 46(4):32–39, 2008.

[5] Joseph Mitola and Gerald Q Maguire. Cognitive radio: making software radios more personal. *IEEE personal communications*, 6(4):13–18, 1999.

[6] An He, Kyung Kyoon Bae, Timothy R Newman, Joseph Gaeddert, Kyouwoong Kim, Rekha Menon, Lizdabel Morales-Tirado, Youping Zhao, Jeffrey H Reed, William H Tranter, et al. A survey of artificial intelligence for cognitive radios. *IEEE transactions on vehicular technology*, 59(4):1578–1592, 2010.

[7] Simon Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE journal on selected areas in communications*, 23(2):201–220, 2005.

[8] Yonghong Zeng and Y-C Liang. Eigenvalue-based spectrum sensing algorithms for cognitive radio. *IEEE transactions on communications*, 57(6):1784–1793, 2009.

[9] Paul D Sutton, Keith E Nolan, and Linda E Doyle. Cyclostationary signatures in practical cognitive radio applications. *IEEE Journal on selected areas in Communications*, 26(1):13–24, 2008.

[10] Yonghong Zeng and Ying-Chang Liang. Spectrum-sensing algorithms for cognitive radio based on statistical covariances. *IEEE transactions on Vehicular Technology*, 58(4):1804–1815, 2008.

[11] Zhi Quan, Wenyi Zhang, Stephen J Shellhammer, and Ali H Sayed. Optimal spectral feature detection for spectrum sensing at very low snr. *IEEE Transactions on Communications*, 59(1):201–212, 2010.

[12] Kyouwoong Kim, Ihsan A Akbar, Kyung K Bae, Jung-Sun Um, Chad M Spooner, and Jeffrey H Reed. Cyclostationary approaches to signal detection and classification in cognitive radio. In *2007 2nd ieee international symposium on new frontiers in dynamic spectrum access networks*, pages 212–215. IEEE, 2007.

[13] Shree Krishna Sharma, Eva Lagunas, Symeon Chatzinotas, and Björn Ottersten. Application of compressive sensing in cognitive radio communications: A survey. *IEEE communications surveys & tutorials*, 18(3):1838–1860, 2016.

[14] Moshe Mishali and Yonina C Eldar. From theory to practice: Sub-nyquist sampling of sparse wideband analog signals. *IEEE Journal of selected topics in signal processing*, 4(2):375–391, 2010.

[15] Babar Aziz, Samba Traoré, Amor Nafkha, and Daniel Le Guennec. Spectrum sensing for cognitive radio using multicoset sampling. In *2014 IEEE Global Communications Conference*, pages 816–821. IEEE, 2014.

[16] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[17] Zhuhua Hu, Yong Bai, Yaochi Zhao, and Yiran Zhang. Support recovery for multiband spectrum sensing based on modulated wideband converter with swsomp algorithm. In *International Conference on 5G for Future Wireless Networks*, pages 146–159. Springer, 2017.

[18] Joel A Tropp, Anna C Gilbert, and Martin J Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal processing*, 86(3):572–588, 2006.

[19] Zhuhua Hu, Yong Bai, Yaochi Zhao, Chong Shen, and Mingshan Xie. Adaptive and blind wideband spectrum sensing scheme using singular

value decomposition. *Wireless Communications and Mobile Computing*, 2017, 2017.

[20] Zhuhua Hu, Yong Bai, Mengxing Huang, Mingshan Xie, and Yaochi Zhao. A self-adaptive progressive support selection scheme for collaborative wideband spectrum sensing. *Sensors*, 18(9):3011, 2018.

[21] Yulong Gao, Yanping Chen, and Yongkui Ma. Sparse-bayesian-learning-based wideband spectrum sensing with simplified modulated wideband converter. *IEEE Access*, 6:6058–6070, 2017.

[22] Jiai He, Wei Chen, Lu Jia, and Tong Wang. An effective reconstruction algorithm based on modulated wideband converter for wideband spectrum sensing. *IEEE Access*, 8:152239–152247, 2020.

[23] Tianyi Xiong, Hongbin Li, Peihan Qi, Zan Li, and Shilian Zheng. Predecision for wideband spectrum sensing with sub-nyquist sampling. *IEEE Transactions on Vehicular Technology*, 66(8):6908–6920, 2017.

[24] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4580–4584. IEEE, 2015.

[25] Kai Yang, Zhitao Huang, Xiang Wang, and Xueqiong Li. A blind spectrum sensing method based on deep learning. *Sensors*, 19(10):2270, 2019.

[26] Jiabao Gao, Xuemei Yi, Caijun Zhong, Xiaoming Chen, and Zhaoyang Zhang. Deep learning for spectrum sensing. *IEEE Wireless Communications Letters*, 8(6):1727–1730, 2019.

[27] Jiandong Xie, Chang Liu, Ying-Chang Liang, and Jun Fang. Activity pattern aware spectrum sensing: A cnn-based deep learning approach. *IEEE Communications Letters*, 23(6):1025–1028, 2019.

[28] Xiangyue Meng, Hazer Inaltekin, and Brian Krongold. End-to-end deep learning-based compressive spectrum sensing in cognitive radio networks. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[29] Shivam Chandhok, Himani Joshi, A. V. Subramanyam, and Sumit J. Darak. Novel deep learning framework for wideband spectrum characterization at sub-nyquist rate. *Wireless Netw*, 27(7):4727–4746, aug 2021.

[30] Woongsup Lee, Minhoe Kim, and Dong-Ho Cho. Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks. *IEEE Transactions on Vehicular Technology*, 68(3):3005–3009, 2019.

[31] Markus Leinonen, Marian Codreanu, and Markku Juntti. Sequential compressed sensing with progressive signal reconstruction in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 14(3):1622–1635, 2015.

[32] Francesco Marino, Luigi Paura, and Roberto Savoia. On spectrum sensing optimal design in spatial-temporal domain for cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 65(10):8496–8510, 2015.

[33] Yonghong Zeng, Ying-Chang Liang, Anh Tuan Hoang, and Rui Zhang. A review on spectrum sensing for cognitive radio: challenges and solutions. *EURASIP journal on advances in signal processing*, 2010:1–15, 2010.

[34] Wang Wei-Gang, Yang Zhen, and Hu Hai-Feng. A method of space-frequency compressed sensing on wideband spectrum detection. *Journal of Electronics & Information Technology*, 35(2):255–260, 2013.

[35] Václav Valenta, Roman Maršálek, Geneviève Baudoin, Martine Villegas, Martha Suarez, and Fabien Robert. Survey on spectrum utilization in europe: Measurements, analyses and observations. In *2010 Proceedings of the fifth international conference on cognitive radio oriented wireless networks and communications*, pages 1–5. IEEE, 2010.

[36] Roger B Bacchus, Antoni J Fertner, Cynthia S Hood, and Dennis A Roberson. Long-term, wide-band spectral monitoring in support of dynamic spectrum access networks at the iit spectrum observatory. In *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 1–10. IEEE, 2008.

[37] Tanim M Taher, Roger B Bacchus, Kenneth J Zdunek, and Dennis A Roberson. Long-term spectral occupancy findings in chicago. In *2011 IEEE international symposium on dynamic spectrum access networks (DySPAN)*, pages 100–107. IEEE, 2011.

[38] Adnan Ahmad Cheema and Sana Salous. Spectrum occupancy measurements and analysis in 2.4 ghz wlan. *Electronics*, 8(9):1011, 2019.

[39] Yunfei Chen and Hee-Seok Oh. A survey of measurement-based spectrum occupancy modeling for cognitive radios. *IEEE Communications Surveys & Tutorials*, 18(1):848–859, 2014.

[40] Soraya Contreras, Gabriel Villardi, Ryuhei Funada, and Hiroshi Harada. An investigation into the spectrum occupancy in japan in the context of tv white space systems. In *2011 6th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 341–345. IEEE, 2011.

[41] HJ Landau. Necessary density conditions for sampling and interpolation of certain entire functions. *Acta Mathematica*, 117:37–52, 1967.

Xianghui Li received the B.S. degree from Hainan University, Haikou, China, in 2022, where he is currently pursuing the master's degree in Tianjin University. His research interests include deep learning and wireless communications.

Zhuhua Hu (Senior Member, IEEE) received the BEng degree from the Jilin University in 2002, the MEng degree from the Jilin University in 2005, and the Ph.D. degree from the Hainan University in 2019. He was a software engineer at Ningbo BIRD Research Institute of China from 2005 to 2006. He was a software engineer at Nanjing Research Institute of ZTE from 2006 to 2007. He was a minister of the software department at Shanghai Aoxun Information Technology Co., Ltd. from 2007 to 2009. He is a professor at the School of Information and Communication Engineering, Hainan University since 2020. He is IEEE Senior Member and CCF Senior Member. He is currently a high-level talent in Hainan Province. His current research interests include artificial intelligence, signal and information processing.

Chong Shen was born in 1981. He received the Ph.D. degree from the Cork Institute of Technology, Cork, Ireland, in 2008. He was a full-time Post-doctoral Research Fellow with the Tyndall National Institute, Ireland from 2008 to 2009. He is currently a Professor with the School of Information and Communication Engineering, Hainan University, Haikou, China. His major research interests include array signal processing, ultra-wideband communications and wireless communications.

Huaming Wu (Senior Member, IEEE) received the B.E. and M.S. degrees from Harbin Institute of Technology, China in 2009 and 2011, respectively, both in electrical engineering. He received the Ph.D. degree with the highest honor in computer science at Freie Universität Berlin, Germany in 2015. He is currently an associate professor at the Center for Applied Mathematics, Tianjin University, China. His research interests include wireless and mobile network systems and deep learning.

Yaochi Zhao received her M.S. degree in pattern recognition and intelligent system from Central South University, Changsha, China, in 2005, and the Ph.D. degree from the Tianjin University, Tianjin, China, in 2023. She worked in Ningbo BIRD Research Institute and Shanghai Wingtech Communication Co., Ltd. for three years. Later, she was engaged in teaching and research work at the College of Information Science & Technology, Hainan University, Haikou, China. Now, she is an associate professor at the School of Cyberspace Security at Hainan University. Her current research interests include deep learning and intelligent information processing.