Deep STFT-CNN for Spectrum Sensing in Cognitive Radio

Zhibo Chen, Yi-Qun Xu, Hongbin Wang, and Daoxing Guo

***Abstract*— Spectrum sensing is one of the crucial technologies used to solve the shortage of spectrum resources. In this letter, based on the short-time Fourier transform (STFT) and convolutional neural network (CNN), we firstly develop a STFT-CNN method for spectrum sensing. The proposed method exploits the time-frequency domain information of the signal samples and achieves the state of the art detection performance. In particular, the method is suitable for various primary users' signals and does not need any priori information. Besides, we also analyze the signal-to-noise ratio robustness and the generalization ability of the proposed algorithm. Finally, simulation results demonstrate that the proposed method outperforms other popular spectrum sensing methods. Notably, the proposed method can achieve a detection probability of 90.2% with a false alarm probability of 10% at SNR $= -15$dB.**

***Index Terms*— Spectrum sensing, deep learning, short-time Fourier transform.**

I. Introduction

As A promising method to solve the spectrum shortage, cognitive radio (CR) allows unlicensed secondary users (SU) to access the authorized primary user's (PU) spectrum, which effectively solves the problem of spectrum shortage [1]. Hence, detecting the presence of primary signals is critical for the realization of CR [2]. Many spectrum sensing methods that are suitable for kinds of scenarios have been proposed in the past decades. Concerning the semi-blind scenario in which noise power's priority information is known, the conventional detector is the maximum eigenvalue detector (MED) [3]. In scenarios where the noise power is not available or inaccurate, a blindly combined energy detection algorithm (BCED) and covariance absolute value (CAV) detection are proposed in [4], [5]. These methods are effective and low complexity. However, their test statistics are extracted from the model-based features. Once the scenario or model changes, the detection performance will be significantly affected.

In recent years, deep learning has shown its significant advantages in extracting data features. Such methods also have great potential in spectrum sensing. Lee *et al.* [6] have proposed a stacked autoencoder based spectrum sensing method (SAE-SS) for OFDM system. To further improve the sensing accuracy under low SNR, they used time-frequency domain signals and developed a new stacked autoencoder (SAE-TF). In [7], a method based on convolutional neural network (CNN) and cycle spectrum is proposed, which transforms the spectral sensing problem into an image processing problem. This approach applied the advantage of CNN in image processing and achieved a high detection probability. Liu *et al.* [8] use the sample covariance matrix as the test statistic and propose a covariance matrix-aware CNN (CM-CNN)-based spectrum sensing algorithm. The performance of this method is close to that of the optimal detector. In [9], Gao *et al.* adopt a new model composed of CNN, long-short term memory network LSTM, and deep neural network (DNN) named CLDNN to achieve spectrum sensing. Besides, to address the problem of labeled data shortage, Xie *et al.* have proposed an unsupervised deep learning-based spectrum sensing method (UDSS) [10] and achieved the performance of the benchmark deep supervised learning-based spectrum sensing algorithm. Although the existing deep-learning based methods further improve the detection performance, these methods have restrictions on the signal of PU. SAE-SS and SAE-TF are only suitable for OFDM system. CM-CNN method requires the signal correlation. And CLDNN algorithm needs the underlying structural information of the modulated signals.

To overcome these limitations, we need to propose a universal approach. The short-Time Fourier transform (STFT) is a widely used time-frequency analysis method, which integrates the signal into the time-frequency domain [11]. Using the short-time Fourier transform, we can obtain a time-frequency matrix and extract the features of the signal. On the other hand, CNN is a class of deep learning, which is designed to process the two-dimensional data. Motivated by this, we choose the time-frequency matrix as the input of the CNN and propose a STFT-CNN spectrum sensing algorithm. The main contributions of this work are as follows.

(1) We propose a universal deep learning-based spectrum sensing method that uses the time-frequency matrix as the test statistic and utilizes a threshold-based mechanism for online detection. This method has no requirements for the PU's signal and is used in various communication scenarios.

(2) The proposed STFT-CNN approach is data-driven and conveniently sets the desired probability of false alarm (P_f).

(3) We conduct extensive experiments using randomly generated samples. Simulation results verify the efficiency of the proposed method in terms of detection performance, SNR-robustness, and generalization.

II. System Model

In this letter, we consider a general spectrum-aware scenario. During the period of the k-th frame, we refer it to the k-th sensing period. In this period, the PU can decide to

Manuscript received October 15, 2020; accepted November 5, 2020. Date of publication November 11, 2020; date of current version March 10, 2021. This work was supported in part by the Jiangsu Provincial Natural Science Foundation of China under Grant BK20191328. The associate editor coordinating the review of this letter and approving it for publication was M. Chafii. *(Corresponding author: Daoxing Guo.)*

The authors are with the College of Communication Engineering, Army Engineering University of PLA, Nanjing 210000, China (e-mail: xyzgfg@sina.com).

Digital Object Identifier 10.1109/LCOMM.2020.3037273

transmit or keep silent, and the SU collects T signal samples. According to PU's status, spectrum sensing can be modeled as a binary hypothesis test problem and the received signal samples in the k-th sensing period are [12]:

$$\begin{cases} H_0 : \{x_k(t)\}_{t=1}^{T} = \{w_k(t)\}_{t=1}^{T} \\ H_1 : \{x_k(t)\}_{t=1}^{T} = \{h_k * s_k(t) + w_k(t)\}_{t=1}^{T}, \end{cases} \tag{1}$$

where H_0 and H_1 represent two hypotheses are denoting the absence and presence of primary signal in the detection frequency band, respectively. h_k represents the channel gain follows Rayleigh distribution. In general, the sensing period is shorter than the channel coherence time. Therefore, the h_k unchanged during k-th sensing period. $x_k(t)$ is the t-th received signal sample, $s_k(t)$, and $w_k(t)$ are the signal from PU and the noise in t-th sample.

In the simulations, we assume that the signal is modulated with unit energy, and the noise is independently complex Gaussian distributed. The STFT-CNN method is a totally-blind algorithm that does not assume any signal-noise prior information. Hence, we simplify the representation of received signal samples in the k-th sensing period as:

$$\mathbf{X}_k = [x_k(1), x_k(2), \ldots, x_k(T)]. \tag{2}$$

III. STFT-CNN Based Spectrum Sensing Method

A. Short Time Fourier Transform

The short-time Fourier transform (STFT) is a time-frequency domain analysis method. It divides a longer time signal into shorter segments of equal length and computes Discrete Fourier Transform (DFT) separately on each shorter segment. By losing a certain degree of frequency resolution, STFT helps us regain the time resolution to some extent. STFT breaks the signal samples up into chunks with a window function, and each chunk is Discrete-Fourier transformed [11],

$$STFT^{(\tau,s)}(X)_{[m,n]} = \sum_{t=1}^{T} x(t) W(t - s \cdot m) e^{-j\frac{2\pi n}{\tau}(t - s \cdot m)}, \tag{3}$$

where $STFT^{(\tau,s)}(X) \in \mathbb{C}^{M \times N}$, $1 \le m \le M$, $1 \le n \le N$, represents the short-time Fourier transform with width τ and sliding step s. M denotes the number of time chunks. N denotes the number of frequency components. $W(\cdot)$ is the sliding window with width τ only has non-zero values for $1 \le t \le \tau$. In this letter, we focus on sliding chunks with Gauss window and no overlaps to simplify the formulation, *i.e.*, $s = \tau$ and $M = T/\tau$. Then, the time-frequency matrix of the signal samples is given by:

$$\mathbf{R}(m, n) = \sum_{t=1}^{T} x(t) W(t - \tau \cdot m) e^{-j\frac{2\pi n}{\tau}(t - \tau \cdot m)}. \tag{4}$$

Due to the uncertainty principle, the transformed representation cannot achieve both a high frequency resolution and a high time resolution at the same time. The time-frequency resolution depends on the length of the window function. We have done a lot of simulation experiments and find the window length of 32 works best when the sampling length is 128. To simplify the handling of bandpass signals, we convert the bandpass signal into an equivalent low-pass signal. According to eq (4), we obtain the time-frequency matrix of the equivalent low-pass signal, where the rows denote time resolution, and the columns represent frequency components. Further, we normalize the matrix and get the time-frequency images in grayscale. As shown in Fig. 1, because the values of time-frequency matrices are different in the sensing frequency band. The values in the frequency band of the PU signals are more prominent than other bands. This difference forms a conspicuous bright band on the time-frequency images. In contrast, due to the noise spectrum is distributed in the sensing frequency band, the values of time-frequency matrix under the state of H_0 are not much different. Hence, Fig. 2 does not have apparent features. This visual difference in time-frequency under H_0 and H_1 makes the CNN suitable for learning the signal samples' time-frequency features. Finally, spectrum sensing is transformed into image recognition.

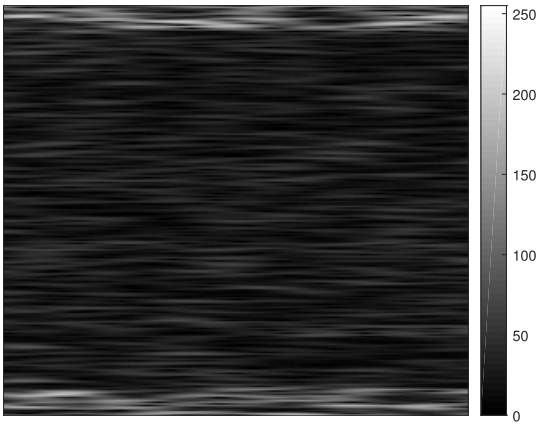

Fig. 1. Time-Frequency image of H_1.

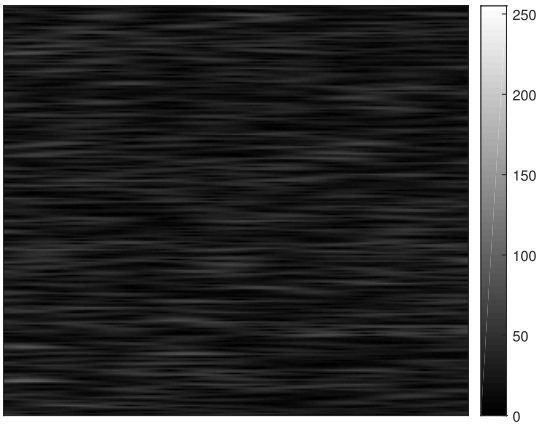

Fig. 2. Time-Frequency image of H_0.

Remark 1: Here, we briefly discuss how STFT is used in Multiple-input Multiple-output (MIMO) system. MIMO systems with multiple antennas at both the transmitter and receiver can improve the channel capacity and reliability of wireless communication systems, and we can collect multiple signal samples at the same time. Using the maximal-ratio combining (MRC) algorithm, we can get a higher signal-to-noise

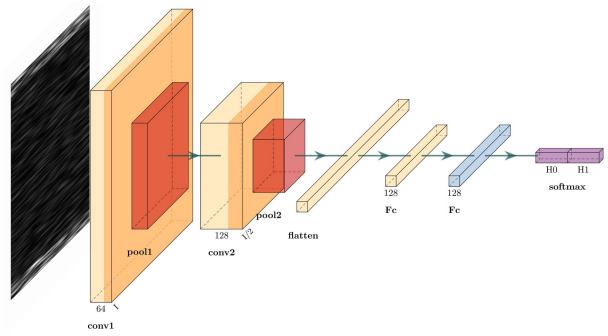

Fig. 3. Network architecture of CNN.

ratio (SNR) signal samples brought by Array Gain. According to eq (4), we obtain a time-frequency matrix. Although the use of STFT in the MIMO system increases the computational complexity, it improves the SNR of the signal samples and effectively improves detection performance.

B. CNN Architecture and Training

Considering the excellent performance of CNN in two-dimensional matrix processing and image recognition, we designed a CNN model for spectrum sensing and the network architecture illustrated in Fig. 3. For training the CNN model, we need to label the time-frequency matrix $\boldsymbol{R}$. Depending on the status of H_0 and H_1, we can express the labeled data set as:

$$Y = \{(\mathbf{R}_1, b_1), (\mathbf{R}_2, b_2), \ldots, (\mathbf{R}_k, b_k)\}, \tag{5}$$

where k is the size of the training data set, and $b_k \in (0, 1)$ is the label. When the status is H_0, b_k= 0, and when the status is H_1, b_k=1. Since spectrum sensing is a binary hypothesis test problem, training the model can be considered as a binary classification problem. Therefore, for the k-th single sample $(\mathbf{R}_k, b_k)$, the label can be converted to a one-hot vector:

$$b_k = \begin{cases} [0,1], & H_0 \\ [1,0], & H_1. \end{cases} \tag{6}$$

Then the labeled signal samples are put into CNN, which consists of two convolutional and pool layers. Flatten the output of the last layer and then put it into the full connection layers. Finally, the output of CNN is a 2×1 class score vector. We adopt the softmax function [13] to normalize it, and the result is:

$$h_\theta(\mathbf{R}_k) = \begin{bmatrix} h_{\theta|H_0}(\mathbf{R}_k) \\ h_{\theta|H_1}(\mathbf{R}_k) \end{bmatrix}, \tag{7}$$

with

$$h_{\theta|H_0}(\mathbf{R}_k) + h_{\theta|H_1}(\mathbf{R}_k) = 1, \tag{8}$$

where θ is the parameter of the CNN network, and $h_\theta(\cdot)$ is the expression of the entire CNN. $h_{\theta|H_i}(R_k)$ represents the class probability of H_i. Hence, the probability of two hypotheses can be expressed as:

$$\begin{cases} P(H_0|\theta) = h_{\theta|H_0}(\mathbf{R}_k) \\ P(H_1|\theta) = h_{\theta|H_1}(\mathbf{R}_k). \end{cases} \tag{9}$$

We adopt a general goal function based on maximizing the likelihood. The expression is:

$$L(\theta) = \prod_{k=1}^{K} \left(h_{\theta|H_0}(\mathbf{R}_k)\right)^{1-b_k} \left(h_{\theta|H_1}(\mathbf{R}_k)\right)^{b_k}, \tag{10}$$

and the log likelihood

$$rCll(\theta) = logL(\theta) = \sum_{k=1}^{K} (1-b_k) \log\left(h_{\theta|H_0}(\mathbf{R}_k)\right) + b_k \log\left(h_{\theta|H_1}(\mathbf{R}_k)\right). \tag{11}$$

The essence of model training is to obtain the optimal parameter, and the expression is:

$$\theta^* = \arg\max_{\theta} L(\theta). \tag{12}$$

We can minimize the loss function to obtain the optimal parameter, in this letter, we choose cross-entropy function as the loss function [14]:

$$H(\theta) = -\sum_{k=1}^{K} b_k \log h_\theta(\mathbf{R}_k) + (1-b_k) \log\left(1 - h_\theta(\mathbf{R}_k)\right). \tag{13}$$

We train the model by adopting the back propagation (BP) algorithm [15], which is widely used in deep learning. And the adaptive moment estimation (Adam) method is used to calculate the optimal parameter [16]. Now we have get the well-trained module and the outputs of well-trained CNN are $h_{H_0|\theta^*}(\mathbf{R}_k)$ and $h_{H_1|\theta^*}(\mathbf{R}_k)$, which are the probability of H_0 and H_1, respectively. Setting the CNN online, for a new received sensing sample, which can be named test data, we judge the state of the PU by comparing the value of the CNN output. However, this method cannot control the probability of false alarms (P_f). In fact, missed detection is a more serious problem than false alarm in cognitive radio. To achieve the given false alarm probability P_f, we use the Monte-Carlo method. Firstly, we select the H_0 label samples from the training data set to constitute a new set. This new data set $Y' = [(\mathbf{R}_1, 0), (\mathbf{R}_2, 0), \ldots, (\mathbf{R}_U, 0)]$ are put into the CNN, then we will get the result. Also we reorder the results

$$h_{\theta|H_1}(\mathbf{R}_u) \leq h_{\theta|H_1}(\mathbf{R}_v), \quad \forall 1 \leq u \leq v \leq U. \tag{14}$$

Finally, we obtained the threshold as

$$\gamma = h_{\theta|H_1}\left(\mathbf{R}_{round((1-P_f)U)}\right), \tag{15}$$

where $round(\cdot)$ is the rounding function which rounds a number to its nearest integer. Now we have obtained the threshold, when online test, for a new test data, we can judge the status of PU by

$$\gamma \underset{H_1}{\overset{H_0}{\gtrless}} h_{H_1|\theta^*}(\mathbf{R}_i). \tag{16}$$

The process of the STFT-CNN algorithm is summarized below

Algorithm 1 STFT-CNN Spectrum Sensing

1: Set $i = 0$, initialize θ with random weights.
2: collect the signal sample X, pre-process the data set into Y.
3: **while** $i \leq Iter_{Max}$ **do**
4: update θ by doing Adam on $H(\theta)$
5: $i = i + 1$
6: **end while**
7: determine the test threshold γ according to eq(15).
8: during the period k, receive the test sample and formulate $\mathbf{R}_k$ according to eq(4)
9: judge the status of the PU according to eq(16)

TABLE I
HYPERPARAMETER OF THE PROPOSED STFT-CNN.

Input : Sample STFT Matrix(Dimension: 128 × 128)	
Layers	**Kernel Size**
S0	NULL
C1+ReLu	64 @ 3 × 3
P1 (Max-Pooling)	2 × 2
C2+ReLu	128 @ 3 × 3
P2 (Max-Pooling)	2 × 2
F1 (Flatten)	NULL
F2+ReLu	128
F3+ReLu	128
F4+Softmax	2
Output: Score Vector (Dimension: 2 × 1)	

Remark 2: Here, we briefly discuss how the labeled training data set can be obtained in practical scenarios. To mark the data, we must know the underlying ground truth PU activity. Hence, we suggest building a database for the PU. The managers of the PU can sample signals in the vicinity and label the signal samples with their corresponding on-off activity recorded by the PU. In order to obtain different SNR levels data, the sampling should take place in various locations form near the PU to far from the PU. Then, we build a complete database. For each SU, it should first measure its sensing SNR and then choose the set of training data whose SNR level is nearest to its sensing SNR to train its deep learning-based spectrum sensing model.

IV. NUMERICAL SIMULATIONS

A. Simulation Environments

The simulation parameter settings, including the training parameters and neural network hyperparameters, are shown in Table I. Instead of sending the true value of the time-frequency matrix to the CNN, we normalize the matrix value before training or testing. The normalized matrix can more clearly reflect the energy information of the received signal in time and frequency domain, which is conducive to extracting signal features. Besides, the normalized matrix reduces the impact of different environments on the model. Even if the background noise changes, the model can still work well. According to simulation results, normalization can effectively improve the generalization ability of the model.

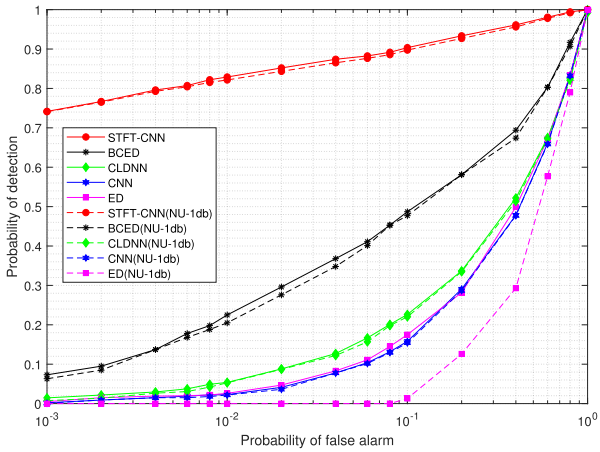

Fig. 4. ROC curves of different algorithms under Gaussian noise: SNR $= -15$dB.

B. Simulation Results

In this section, we provide extensive simulation results to demonstrate the performance of the STFT-CNN method. Furthermore, we evaluate the detection performance in terms of the PD versus SNR (PD-SNR) and Receiver Operating Characteristics (ROC), through comparing the proposed STFT-CNN method with the ED [3], BCED [4], CNN [7], and CLDNN [9].

In the simulations, we assume the PU signals are QPSK, and the sample length is 128. For a fair comparison, we performed extensive cross-validation for all the test models based on deep learning to determine the best hyperparameters through the fine-tune process. By changing different values of P_f, we calculate different thresholds to get the corresponding P_d. As shown in Fig. 4, it presents the ROC curves of different methods under SNR $= -15$dB. According to the results, it can be observed that the P_d of STFT-CNN has lager promotion than other methods. Considering the operating point with $P_f = 10\%$, the P_d of STFT-CNN is 90.2%, the BCED is 58.7%, which is 31.5% higher.

In addition, we note that the test statistic of the STFT-CNN method depends on the SNR of the data set. Hence, to further demonstrate the performance of the algorithm, we should evaluate the SNR-robustness of the proposed method. We introduce noise uncertainty (NU) in spectrum sensing [8]. Because of the existence of NU, the noise power is $\widehat{\sigma}_n^2 = \alpha\sigma_n^2$, where σ_n^2 is the actual noise power, α is the NU factor. α is uniformly distributed in the interval of $[-B, B]$. Fig. 4 shows the ROC curves of the various method under NU. It can be observed that the performance of ED has been greatly affected. However, the STFT-CNN with NU performs better. Because the NU brings more different samples, which makes the STFT-CNN learn more features. Besides, the CNN, CLDNN, and BCED are total-blind algorithms, which require neither prior knowledge about noise power. These methods keep almost the same detection performance when the NU exists.

To further evaluate the detection performance, Fig. 5 presents the PD-SNR curves of different methods, and the detection SNR in spectrum sensing is about -20dB$\sim$0dB.

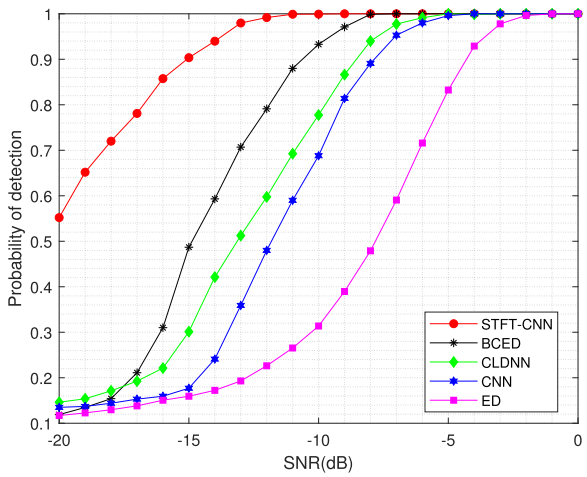

Fig. 5. Probability of detection versus SNR of different algorithms under Gaussian noise: Pf = 0.1.

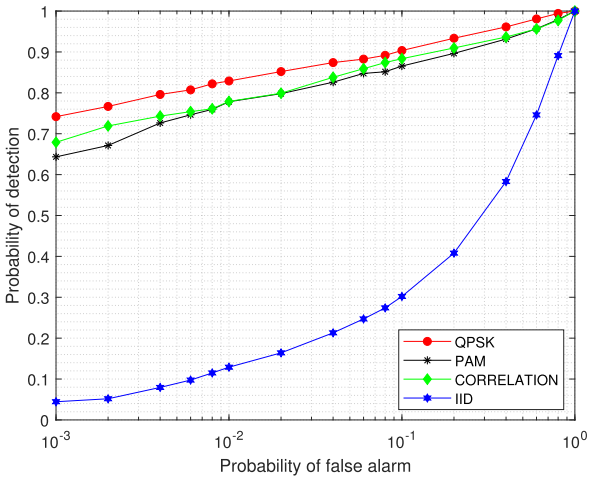

Fig. 6. Generalization ability to different PU signals.

It is shown that the detection performance of the proposed method still maintains the ahead. Particularly, the STFT-CNN method achieves a P_d of 98.1% under the P_f of 10%.

Moreover, we evaluate the generalization ability of the PU signal. We consider four scenarios where the received PU signals are i.i.d. Gaussian random vector, correlated Gaus- sian random vector, QPSK, and PAM in the presence of Gaussian noise. As shown in Fig. 6, the ROC curve of QPSK is very close to that of PAM and correlated signals at the whole P_f range. But the performance of i.i.d. signals are not good. Because the frequency spectrum of i.i.d. signals is distributed over the entire frequency band, which cannot extract the time-frequency features. As a summary, the STFT-CNN method is insensitive to the PU signals and provides a decent generalization ability.

V. Conclusion

In this letter, we proposed a novel blind spectrum sensing method based on short-time Fourier transform and deep learning. Different from the existing spectrum sensing methods based on deep learning, the STFT-CNN method exploits the information of signal in the time-frequency domain and is free of the signal and noise assumes. Moreover, it has no requirements for the PU signal and considerable SNR-robustness. With extensive simulations, the detection performance of the STFT-CNN method achieves significant improvement in the total-blind scenario.

References

[1] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1277–1304, 2nd Quart., 2017.

[2] Y. Chen and H.-S. Oh, "A survey of measurement-based spectrum occupancy modeling for cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 848–859, 1st Quart., 2016.

[3] Y. Zeng, C. L. Koh, and Y.-C. Liang, "Maximum eigenvalue detection: Theory and application," in *Proc. IEEE Int. Conf. Commun.*, 2008, pp. 4160–4164.

[4] Y. Zeng, Y.-C. Liang, and R. Zhang, "Blindly combined energy detection for spectrum sensing in cognitive radio," *IEEE Signal Process. Lett.*, vol. 15, pp. 649–652, 2008.

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

[6] Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Sensing OFDM signal: A deep learning approach," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7785–7798, Nov. 2019.

[7] G. Pan, J. Li, and F. Lin, "A cognitive radio spectrum sensing method for an OFDM signal based on deep learning and cycle spectrum," *Int. J. Digit. Multimedia Broadcast.*, vol. 2020, pp. 1687–7578, Mar. 2020.

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

[9] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1727–1730, Dec. 2019.

[10] J. Xie, J. Fang, C. Liu, and L. Yang, "Unsupervised deep spectrum sensing: A variational auto-encoder based approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5307–5319, May 2020.

[11] S. Yao *et al.*, "Stfnets: Learning sensing signals from the time-frequency perspective with short-time Fourier neural networks," in *Proc. World Wide Web Conf.*, 2019, pp. 2192–2202.

[12] M. Santhoshkumar and K. Premkumar, "Energy-throughput tradeoff with optimal sensing order in cognitive radio networks," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2020, pp. 602–605.

[13] P. Deng, H. Wang, S.-J. Horng, D. Wang, J. Zhang, and H. Zhou, "Softmax regression by using unsupervised ensemble learning," in *Proc. 9th Int. Symp. Parallel Archit., Algorithms Program. (PAAP)*, Dec. 2018, pp. 196–201.

[14] O. Vynokurova, D. Peleshko, Y. Borzov, S. Oskerko, and V. Voloshyn, "Hybrid multidimensional Wavelet-Neuro-System and its learning using cross entropy cost function in pattern recognition," in *Proc. IEEE 2nd Int. Conf. Data Stream Mining Process. (DSMP)*, Aug. 2018, pp. 305–309.

[15] Z. Man, H. R. Wu, S. Liu, and X. Yu, "A new adaptive backpropagation algorithm based on Lyapunov stability theory for neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1580–1591, Nov. 2006.

[16] Z. Chang, Y. Zhang, and W. Chen, "Effective adam-optimized LSTM neural network for electricity price forecasting," in *Proc. IEEE 9th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2018, pp. 245–248.