

## RESEARCH ARTICLE

# Spectrum Sensing in Cognitive Radio Using CNN-RNN and Transfer Learning

SURENDRA SOLANKI<sup>1</sup>, VASUDEV DEHALWAR<sup>1</sup>, (Senior Member, IEEE),  
JAYTRILOK CHOUDHARY<sup>1</sup>, (Member, IEEE), MOHAN LAL KOLHE<sup>2</sup>, (Senior Member, IEEE),  
AND KOKI OGURA<sup>3</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of CSE, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh 462003, India

<sup>2</sup>Department of Engineering Science, University of Agder, 4604 Kristiansand, Norway

<sup>3</sup>Department of Electrical Engineering, Faculty of Science and Engineering, Kyushu Sangyo University, Fukuoka 813-8503, Japan

Corresponding authors: Surendra Solanki (surendra1.iet@gmail.com) and Mohan Lal Kolhe (mohan.L.kolhe@uia.no)

**ABSTRACT** Cognitive radio has been proposed to improve spectrum utilization in wireless communication. Spectrum sensing is an essential component of cognitive radio. The traditional methods of spectrum sensing are based on feature extraction of a received signal at a given point. The development in artificial intelligence and deep learning have given an opportunity to improve the accuracy of spectrum sensing by using cooperative spectrum sensing and analyzing the radio scene. This research proposed a hybrid model of convolution and recurrent neural network for spectrum sensing. The research further enhances the accuracy of sensing for low SNR signals through transfer learning. The results of modelling show improvement in spectrum sensing using CNN-RNN compared to other models studied in this field. The complexity of an algorithm is analyzed to show an improvement in the performance of the algorithm.

**INDEX TERMS** Spectrum sensing, cognitive radio, deep neural network, convolutional neural network, recurrent neural network.


## I. INTRODUCTION

The wireless communication is at the forefront of subsequent generation development as it has many advantages. Improvements in last-mile connectivity, a short time to commission, and a high-speed data rate make wireless an important medium [1]. The exponential growth of mobile apps and gaming devices exerted pressure on wireless bandwidth. Spectrum scarcity in 3g and 4g are a bottleneck that hinders the development of more wireless applications [2].

In order to improve spectrum utilization, cognitive radio technology is recommended. Cognitive radio allows sharing of the licensed vacant spectrum to unlicensed/secondary users opportunistically based on radio scene analysis [3], [4]. Cognitive radio can be used for Big data in a machine to machine communication. Radio scene analysis comprises spectrum sensing, interference, channel estimation,

allocation patterns etc. [5], [6]. Many spectrum sensing techniques such as energy detection, match filter, cyclostationary feature detection etc., are used for spectrum sensing [7], [8]. These techniques require statistics of signals and noise for spectrum sensing. These methods are prone to the hidden terminal problem, multipath fading, and shadowing [9]. Due to the complex nature of signal and noise, Artificial Intelligence (AI) based deep learning methods are pursued for better spectrum utilization.

Deep learning is used in a wireless network for resource allocation, non-orthogonal multiple access, multiple input-multiple outputs, signal modulation recognition, and spectrum sensing [10], [11]. Deep learning-based dynamic spectrum sensing minimizes flaws in channel identification and classification as it learns from the historical radio scene of spectrum allocation [12], [13]. The Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) having learning capacity are used to develop a sensing model. The CNN is used to extract features, and RNN exploits the temporal characteristics of the spectrum.

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang .

This research intends to develop a deep learning-based spectrum sensing method based on data features, patterns and temporal characteristics to improve spectral efficiency. The accuracy of prediction using spectrum sensing and minimizing the algorithm's complexity is the objective of this research. The key contributions of this article are as follows:

- 1) A hybrid CNN-RNN model is proposed to detect the presence or absence of a signal at a given location. Spectrum sensing is a complex task, and sensing accurately can improve spectrum utilization.
- 2) Transfer learning is applied for the first time on the intermediate results of the CNN-RNN model to improve the accuracy of spectrum sensing.
- 3) The results demonstrate improvement in the probability of detection even at low SNR. It shows improvement in sensing as compared to other models.

The paper is organized as follows: Section 2 describes the related research work in this field. Section 3 presents the deep neural network-based architecture. The results are discussed in section 4, followed by a conclusion in section 5.

## II. RELATED WORK

Deep learning techniques have been used in a variety of applications recently. We are experimenting with these techniques for performing the task of spectrum sensing. The seminal research work in the field of deep learning is discussed here.

Gao et al. [14] introduced spectrum sensing models DetectNet and SoftCombinationNet. The author used the Radioml2016.10a dataset for experiments. In-phase and quadrature-phase of a signal are used for model building. Lee et al. [15] proposed cooperative spectrum sensing (CSS) based on deep CNN. Spatial and spectral correlations of received signals are used for model building. Liu et al. [16] proposed a deep learning model for spectrum sensing. The test statistic used by the method is the time-frequency matrix, and for online detection, a threshold-based mechanism is exploited. Extensive experiments were performed with samples generated randomly. The experiment results show the efficiency in terms of SNR-robustness and detection performance. Soni et al. [17] introduced a long short-term memory (LSTM) based spectrum sensing method. The implicit features like the temporal correlation could be learned from the spectrum data by this method. The statistics of primary user activity can be used to improve Cognitive Radio performance. Sarikhan et al. [18] proposed a deep reinforcement learning-based model for improving the detection performance of cooperative spectrum sensing. Reinforcement learning was employed for updating the required secondary user measurements. This model increases the CSS efficiently from a time and resource viewpoint. Pati et al. [19] proposed a CNN and transfer learning-based architecture for spectrum sensing. Transfer learning is used to decrease the training data, sensing time, and computational time in the proposed work. Xie et al. [20] proposed CNN-LSTM architecture for spectrum sensing. CNN is used to learn the features from the covariance matrix of sensing data. LSTM learns the

primary user activity pattern by energy and correlation features. Peng et al. [21] proposed spectrum sensing based on the CNN network model and transfer learning. Received signals at secondary users are sampled, filtered and then fed into CNN in the form of in-phase and quadrature-phase. Results prove the robustness of the proposed model. Chen et al. [22] proposed spectrum sensing based on a short-time Fourier transform and convolutional neural network. The authors used signals' time-frequency information and achieved the best detection performance. Proposed model achieved 90.2 percentage detection at  $-15$ dB SNR signals. Liu et al. [23] proposed a deep learning-based spectrum sensing for space air-ground integrated network (SAGIN). The input used was the sample covariance matrix. The author proposed CNN-based spectrum sensing, in which he derived the convolutional neural network-based Likelihood Ratio Test (LRT) using Neyman Pearson's theorem. Compared to the traditional spectrum sensing schemes, the detection performance of the proposed approach is improved under a low SNR regime. A blind threshold-setting scheme is proposed without using the system's preliminary information to eliminate the impact of noise uncertainty. Chandhok et al. [24] proposed SenseNet, a novel architecture for automatic modulation classification and wideband spectrum sensing. The model is trained on the basis of signals in-phase and quadrature-phase (IQ), and amplitude-phase. Xing et al. [25] proposed spectrum sensing based on 1-D CNN, bidirectional LSTM and self-attention. Experiments are performed on QAM16 modulated signals of the RadioML2016.10a dataset. Their study performs better than CNN, LSTM, ResNet and CLDNN models. It can be observed from this literature that the models which employed deep learning techniques performed better than traditional sensing methods.

## III. PROPOSED METHODOLOGY

Spectrum sensing is a vital part of cognitive radio. When a spectrum band is underutilized by the licensed user at a given time and location, the secondary users can use this underutilized spectrum [26]. The radio scene analysis of spectrum sensing includes a collection of statistics of occupancy pattern, primary user activity, idle/busy time, minimum duration, mean, higher-order moments and distribution [27]. This statistical information helps predict future spectrum occupancy trends, schedule spectrum sensing, selecting appropriate spectrum bands and channel operations. Spectrum sensing is described by a binary classification problem elaborated in section III. A of this paper.

The architecture of the proposed spectrum sensing model is illustrated in Fig.1. The model has three stages. In the first stage, the RadioML2016.10b dataset consisting of eight digital signals and two analog signals of QPSK, QAM16, QAM64 and GFSK is split into two parts, one in the range of  $-20$ dB to  $+18$ dB (source dataset) and the second in the range of  $-20$ dB to  $-2$ dB (target dataset) in decrements of 2dB. The purpose of the split is to tackle unstable training of the noisy signals and provide more efficient performance

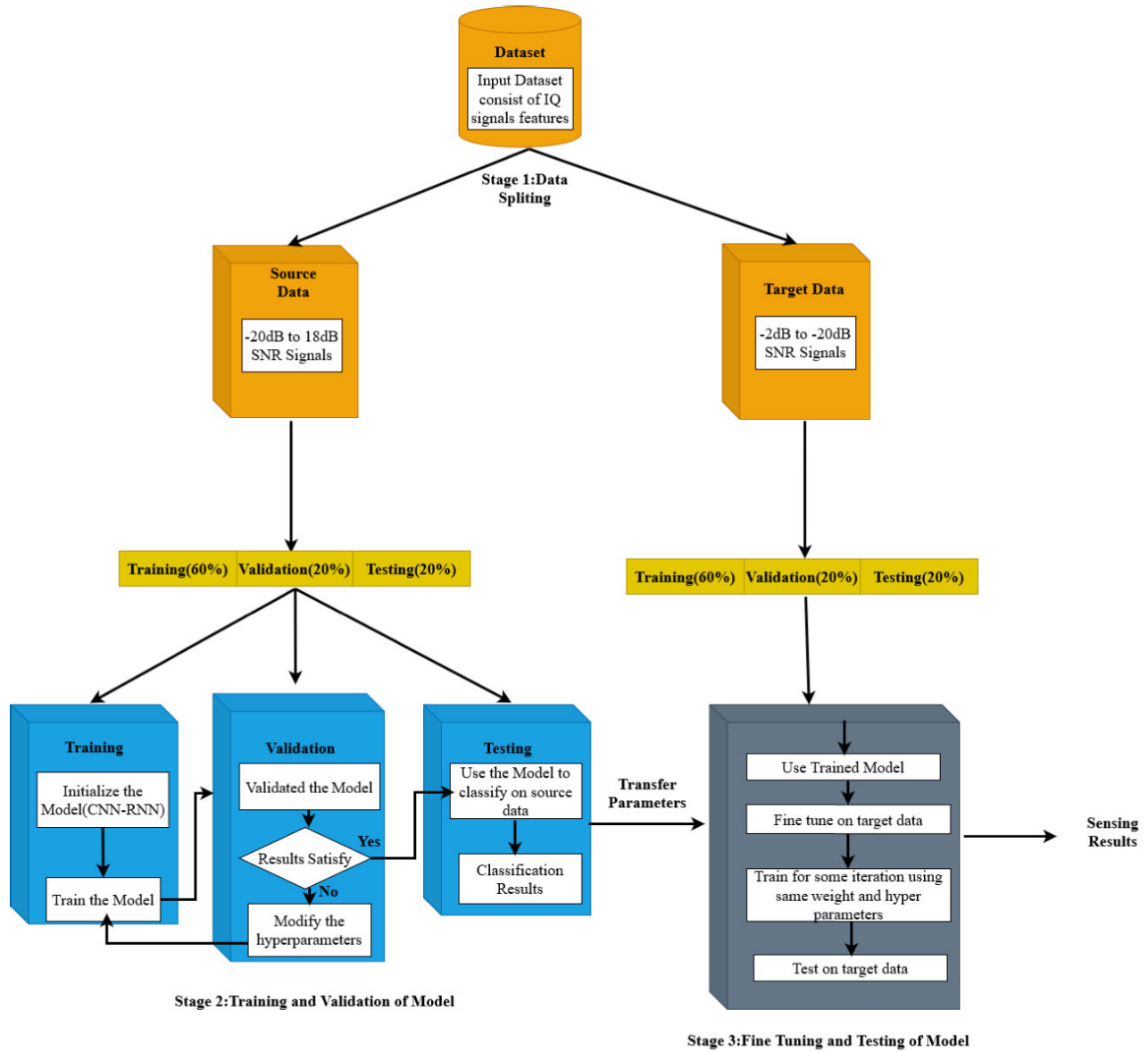


FIGURE 1. Stages of proposed methodology.

accuracy through transfer learning. The source data is used for training the model where positive values are paramount. Target data is used for fine-tuning the negative value from  $-20\text{dB}$  to  $-2\text{dB}$ , where a majority of values are negative.

In the second stage, signal components (in-phase, quadrature-phase) are selected as input for training, validation and testing on source data. The outcome of the second stage will be the optimal structure of the model (CNN-RNN) by tuning the hyperparameters. Finally, in the third stage, the model is fine-tuned on target data, and the efficiency of the model is compared with other neural network models (DNN, CNN, LSTM, DetectNet) for the probability of detection, false alarm, and sensing error. The methodology of the proposed model of spectrum sensing is discussed in the following subsections.

### A. NEURAL NETWORK MECHANISM FOR SPECTRUM SENSING

Spectrum sensing is represented as a binary hypothesis [11]:

$$\begin{aligned}
 H_1 &: rec(n) = prim(n) + nos(n), \\
 H_0 &: rec(n) = nos(n).
 \end{aligned} \tag{1}$$

where  $H_1$  represents the channel occupied (Signal and noise component present) and  $H_0$  represents the channel is idle (noise component present). Here,  $rec(n)$  is the received signal,  $prim(n)$  is the primary user signal and  $nos(n)$  represents the noise signal [28], [29].

The pseudo-code of the spectrum sensing model is given in algorithm 1. The model takes signal components as input and classifies the signals as either presence of primary user

**Algorithm 1** Spectrum Sensing Based on CNN-RNN and Transfer Learning**Input:** Received signal vector  $X$  with SNR values  $S_i$ ,  $i$  ranging from  $-20$  to  $+18$  and Label  $y$ ;**Output:** Sensing Result of primary user Vacancy;

- 1)  $X_I \leftarrow \text{Acos}(\theta)$ ,  $X_Q \leftarrow \text{Asin}(\theta)$
- 2)  $\hat{X} \leftarrow \text{Concatenate}(X_I, X_Q)$
- 3)  $\hat{X}_{norm} \leftarrow \text{EnergyNormalize}(\hat{X})$
- 4) for SNR in range  $-20$ - $18$ :
  - a)  $\hat{X}_1 = \hat{X}_{norm}^{S_{-20-18}}$
  - b)  $y_1 = y^{S_{-20-18}}$
- 5) for SNR in range  $-2$ - $(-20)$ :
  - a)  $\hat{X}_2 = \hat{X}_{norm}^{S_{-2-(-20)}}$
  - b)  $y_2 = y^{S_{-2-(-20)}}$
- 6) Initialize parameters  $w^1$  for Model  $f()$
- 7)  $w^1 \leftarrow f(\hat{X}_1^{train})$
- 8)  $t \leftarrow 0$
- 9) **while** not converge do
  - a)  $t \leftarrow t + 1$
  - b) Sample observation in batches  $(\hat{X}_1^{train}, y_1^{train})$
  - c)  $\hat{y}_1^{train} \leftarrow f(\hat{X}_1^{train}, w^1)$
  - d)  $Loss_{CCE}(\hat{y}_1^{train}, y_1^{train}) = -\sum_i (y_1^{train}[i] \log \hat{y}_1^{train}[i] + (1 - y_1^{train}[i]) \log(\hat{y}_1^{train}[i]))$
  - e)  $w^1 = w^1 - \eta(t) \nabla_{w^1} Loss_{CCE}$
- 10)  $\hat{y}_1^{test} \leftarrow f(\hat{X}_1^{test}, w^1)$
- 11) Initialize parameters  $w^2$  from  $w^1$  i.e., transfer weights
- 12)  $w^2 \leftarrow f(\hat{X}_2^{train})$
- 13)  $t \leftarrow 0$
- 14) **while** not converge do
  - a)  $t \leftarrow t + 1$
  - b) Sample observation in batches  $(\hat{X}_2^{train}, y_2^{train})$
  - c)  $\hat{y}_2^{train} \leftarrow f(\hat{X}_2^{train}, w^2)$
  - d)  $Loss_{CCE}(\hat{y}_2^{train}, y_2^{train}) = -\sum_i (y_2^{train}[i] \log \hat{y}_2^{train}[i] + (1 - y_2^{train}[i]) \log(\hat{y}_2^{train}[i]))$
  - e)  $w^2 = w^2 - \eta(t) \nabla_{w^2} Loss_{CCE}$
- 15)  $\hat{y}_2^{test} \leftarrow f(\hat{X}_2^{test}, w^2)$

signals or absence. For the given input dataset  $D = [X_I, X_Q]$ , where  $X_i$  denotes an input vector consisting of radio signal input features ( $X_I$  represent the in-phase component and  $X_Q$  represent the quadrature-phase component). The aim is to define a function in order to detect the primary user signal based on the signal, which is expressed as below equation 2.

$$\hat{y} = f(X_I, X_Q) \quad (2)$$

where  $\hat{y}$  denotes the classified signal,  $f(\cdot)$  is the combined mathematical function of the CNN-RNN model as shown by equation 3.

$$f(X) = [f_1(X), f_2(X)] \quad (3)$$

where  $f_1(X)$  denotes the CNN model and  $f_2(X)$  represents the RNN in proposed model [30].

$$y_{ij}^l = \sigma(b_j^l + \sum_{m=1}^M w_{m,j}^l x_{i+m-1,j}^0) \quad (4)$$

Equation 4 shows the results after the input is passed through a convolutional layer with  $y_{ij}^l$  being the output vector

of the  $l_{th}$  layer for input vector  $x_{ij}$  and output from the previous layer  $x_{ij}^l$ .  $b$  is the bias,  $w$  is the weight of the kernel with  $m$  filters and  $\sigma$  activation function.

The equation for an RNN in a typical forward pass is stated by equation 5, where  $x_t$  represents present input,  $h_{t-1}$  the previous state,  $w^{xh}$  as the weight between inputs to the hidden unit,  $w^{hh}$  being weight between hidden to the hidden unit, i.e., the recurrent weight and bias  $b$ .

$$z_t = w^{xh} \times x_t + w^{hh} \times h_{t-1} \quad (5)$$

$w$  denotes the weights that are optimized using categorical cross-entropy cost function which is calculated by equation 6 and equation 7 where  $Loss_{CCE}$  is the cost function [31].

$$Loss_{CCE}(\hat{y}^{ss}, y) = -\sum_i (y[i] \log \hat{y}^{ss}[i] + (1 - y[i]) \log(\hat{y}^{ss}[i])) \quad (6)$$

$$w^{ss} = w^{ss} - \eta(t) \nabla_{w^{ss}} Loss_{CCE} \quad (7)$$

**B. CNN-RNN MODEL ARCHITECTURE**

The proposed spectrum sensing architecture using CNN-RNN is described in Fig.2. It is comprised of (i) CNN layers and (ii) RNN layers. The model is designed as a CNN-RNN architecture in order to implicitly learn the important features in the spectrum data. The CNN component is used for feature extraction. The RNN component exploits the temporal characteristics in the spectrum data.

The model has two convolutional layers in parallel, the output of which is passed through the dropout layer. The purpose of the dropout layer is to provide effective regularization to prevent overfitting and improve generalization performance. Next, we used the concatenate layer, the output of which is passed through the flatten layer to adjust the dimensions for the layers further ahead. Next, we have the dense layer that forwards the output of the previous layer to the RNN layer through a linear activation function. The RNN layer processes this input and sends the output to the subsequent two dense layers. Next, we have activation functions (PReLU and softmax). Output from the activation function will be spectrum-sensing decision. Different variants of RNN were also experimented with to test the efficiency of other models. However, the vanilla RNN variant proved to be performing the best with less overhead of parameters. The convolution layer extracts implicit information in the time dimension [12] and also provides features of high concentration to the RNN layer [32]. The long-term memory ability of the RNN, along with CNN, makes it suitable for efficiently learning the features of signals. The architecture of the CNN component is inspired by the inception model [33]. The input enters the two convolution layers respectively, as shown in Fig.2. which has the feature of balancing and generalization. It allows creating deep architectures, maintaining the computational budget while also increasing the network's depth and width. [34].

In the table 1 CNN-RNN network parameters is presented [35]. The number of filters per convolution layer is 60, with a filter size of 10. The number of cells per RNN layer and

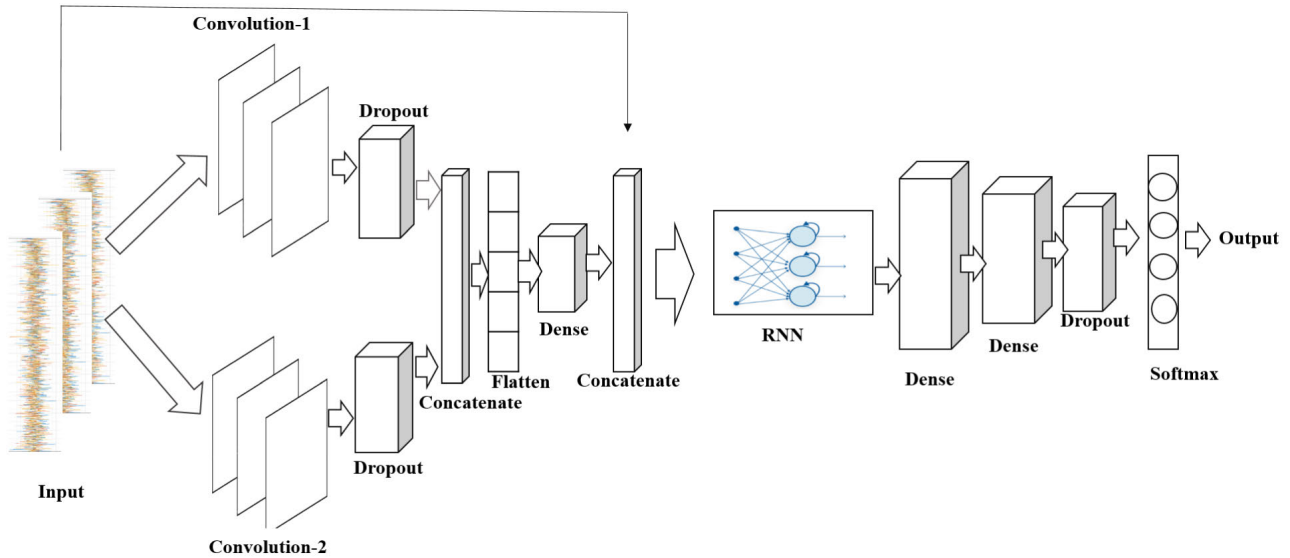


FIGURE 2. Proposed CNN-RNN model architecture.

the number of neurons per FC layer is 128. Adam [36] optimizer is used with an initial learning rate of 0.001 and batch size equals 128. Adam is an optimization algorithm used in place of stochastic gradient descent for training deep learning models. It handles sparse gradients on noisy problems. Adam optimization is used since it is computationally efficient, its implementation is straightforward with few memory requirements, and it is also suited for problems that are large in terms of data or parameters. In order to prevent overfitting, dropout regularization is used with a 0.3 ratio.

TABLE 1. The CNN-RNN network structure.

Layer	Output	Details
Input	(2, × 64)	
Conv1D	(2 × 60)	Filters 60, size=10
Conv1D	(2 × 60)	Filters 60, size=10
Dropout	/	Dropout=0.3
Dropout	/	Dropout=0.3
Concatenate	(2, × 120)	
Flatten	(240)	
Dense+Activation	(64)	Linear
Concatenate	(3, × 64)	
RNN	(128)	
Dense+Activation	(128)	PReLU
Dense+Activation	(128)	PReLU
Dropout	/	Dropout=0.3
Dense+Softmax	(Spectrum Sensing Decision)	One Hot

C. TRANSFER LEARNING

Transfer learning is applied to improve the performance of spectrum sensing. Transfer learning is used where prior knowledge from a given task is used to solve a new but similar task instead of learning from the initial. It saves significant time and computation burdens [19], [21].

The concept of transfer learning is applied by transferring knowledge, i.e., features learned across different SNRs. The model initially trains, validates, and tests a CNN-RNN model on source data (−20dB to 18dB SNR signal). The CNN-RNN model is the one as discussed in the previous section III-B. While training, learning is achieved by weight updates on all layers of the proposed network. This phase continues until the learned features are sufficient to attain accurate sensing decisions. This gives us the optimal model for spectrum sensing, which allows for the transfer of knowledge. The same weights, parameters, and trained (CNN-RNN) model are transferred as we have optimal parameters test results. Now, this trained model is used for fine-tuning target data (−2dB to −20dB SNR signals) for a specified number of iterations. Thus, transfer learning is achieved in order to improve the efficiency of the model.

IV. RESULTS AND DISCUSSION

The dataset, evaluation metrics and experimental results for the proposed scheme are presented in this section.

A. DATASET DESCRIPTION

The RadioML2016.10b is a radio signals dataset generated by authors O’Shea and Corgan [37]. Dataset consists of ten different modulated signals(BPSK, QPSK,8PSK, QAM16, QAM64, CPFSK, GFSK, PAM4, AM-DSB, AM-SSB). Eight are digital signals out of these ten variants, and two are analog signals. The signals are generated for different ranges of SNRs. These SNR values range from +18dB to −20dB in decrements of 2dB. The QAM16, QPSK, QAM64 and GFSK signals are used for our experiments. QPSK is a form of phase-shift keying in which two bits are modulated at once, selecting one of four possible carrier phase shifts (0, 90, 180, or 270 degrees). QAM is a composite modulation technique



that combines both phase modulation and amplitude modulation. In QAM16, each symbol represents 4 bits, while in QAM64, each symbol is represented by 6 bits. QAM benefits from the concept that two signal frequencies, one shifted by 90 degrees with respect to the other, can be transmitted on the same carrier. GFSK is a type of FSK modulation which uses a Gaussian filter to shape the pulses before they are modulated. The signal and noise components from the dataset are considered primary user signals. This noise is generated with dimensions exact as the signals. It follows the zero-mean circularly symmetric complex Gaussian (CSCG) distribution.

## B. EVALUATION METRICS

The performance of the proposed model is evaluated over different metrics, which consist of the probability of detection ( $P_d$ ), probability of false alarm ( $P_f$ ), and sensing error (SE) [15]. When a cognitive radio node, i.e., secondary user, speculates that a channel is occupied by a primary user when it is actually not is termed a false alarm. Missed detection occurs when a cognitive radio node speculates that the channel is not occupied by a primary user when it actually is occupied. False alarm leads to undiscovered spectral opportunities, which in turn decreases spectral efficiency while missed detection causes harmful interference between primary and secondary users. These two types of errors collectively result in spectrum sensing errors. The sensing error is calculated by averaging the probability of false alarms and missed detection.  $P_d$  is the probability of declaring the presence of the primary user in the event of the primary user occupying the spectrum, and  $P_f$  is the probability of declaring the presence of the primary user when the spectrum is vacant. The probability of miss detection is the probability of declaring the vacancy of the spectrum when the primary user is present. These parameters are discussed in detail in our previous research paper [2].

## C. EXPERIMENT RESULTS AND ANALYSIS

The research uses a 64-bit Windows computer equipped with Intel Core i5-9300H@2.90 GHz, 8GB of memory. All experiments are performed using Keras deep learning library on NVIDIA GPU Geforce GTX 1650. The RadioML2016.10b dataset is used for the evaluation of results and compared with the models of DNN, CNN, LSTM, and DetectNet.

Tables 2 and 3 represents the performance metrics on QAM16 signals having 64 and 128 sample length respectively. The probability of detection is shown for signals having an SNR value of  $-20$ dB. The  $P_f$  of our network model is low, which is desired to be achieved. The  $P_f$  of LSTM is the lowest; however, it is not able to achieve high  $P_d$  and low sensing error. The SE of the proposed model is the lowest as compared to other models. The proposed model achieves optimal performance in recognizing primary users' signals.

Tables 4 and 5 present the metrics for QPSK modulated signals with 64 and 128 sample lengths, respectively.  $P_f$  of our network model is not the lowest, but it is well within the suitable range (0 to 0.1). Other models with lower  $P_f$  are not

**TABLE 2. Performance metrics on QAM16 signals (64 sample length).**

Models	$P_f$ (%)	Sensing Error(%)	$P_d$ (-20 dB)(%)
CNN	00.19	14.31	26.35
DNN	18.28	21	38.67
LSTM	00.00	14.37	24.93
DetectNet	00.99	14.92	24.95
<b>Proposed</b>	<b>00.05</b>	<b>13.46</b>	<b>51.84</b>

**TABLE 3. Performance metrics on QAM16 signals (128 sample length).**

Models	$P_f$ (%)	Sensing Error(%)	$P_d$ (-20 dB)(%)
CNN	01.35	14.81	26.9
DNN	29.88	25.44	46.11
LSTM	00.00	14.36	24.41
DetectNet	00.86	14.71	23.24
<b>Proposed</b>	<b>00.25</b>	<b>13.53</b>	<b>50.22</b>

able to show better performance in terms of sensing error or  $P_d$ , i.e., high  $P_d$  and low sensing error. The sensing error of the proposed model is the lowest, and the  $P_d$  for  $-20$ dB SNR also denotes the ability of the proposed model for spectrum sensing. This proves that our model performs efficiently for the QPSK modulated signals.

**TABLE 4. Performance metrics on QPSK signals (64 sample length).**

Models	$P_f$ (%)	Sensing Error(%)	$P_d$ (-20 dB)(%)
CNN	00.18	14.61	25.29
DNN	32.90	26.11	47.19
LSTM	00.00	14.47	24.47
DetectNet	00.33	14.71	24.47
<b>Proposed</b>	<b>00.84</b>	<b>13.68</b>	<b>50.73</b>

**TABLE 5. Performance metrics on QPSK signals (128 sample length).**

Models	$P_f$ (%)	Sensing Error(%)	$P_d$ (-20 dB)(%)
CNN	00.02	15.19	25.59
DNN	29.31	24.96	46.06
LSTM	00.00	14.42	25.48
DetectNet	00.81	14.67	26.63
<b>Proposed</b>	<b>0.54</b>	<b>13.53</b>	<b>50.58</b>

From the tables, it can also be observed that the  $P_f$  value of DetectNet model is larger than the proposed method for 16QAM signals, but in the case of QPSK signals for 64 sample length, the  $P_f$  value of DetectNet is smaller. This is because QAM16 and QPSK both have different signal characteristics. The DetectNet model is getting a low false alarm for QPSK 64 sample length. However, the probability of detection is less than the proposed model, which is more important than a false alarm. Also, the  $P_f$  of the proposed model is in the accepted range which is 0 to 0.1 according to IEEE 802.22 standard [38]. When looking at the overall results, the probability of detection of the proposed model is better than all the models, and sensing error is lower than all.

Fig.3 to Fig.6 present the comparison with other models on the QAM16 and QPSK signals with sample lengths 64 and 128, respectively. The neural network models, namely, DNN, CNN, LSTM, and DetectNet [14] are compared with our

model to prove its efficiency and accuracy of sensing. Fig.3 and Fig.4 show the detection performance of 64 and 128 sample length QAM16 modulated signals respectively. Similarly, Fig.5 and Fig.6 present the model performance for QPSK modulated signals. First, our model is trained and tested on source data, and then the trained parameters are utilized as initializers for data with low SNR signals from  $-2\text{dB}$  to  $-20\text{dB}$  SNR signals. It is clear from the figures that the proposed and DNN models show better detection performance as compared to CNN, LSTM, and DetectNet. However, the  $P_f$  of DNN is higher than other models, making it unsuitable for spectrum sensing. The proposed model attained the least  $P_f$ , followed by LSTM, CNN, and DetectNet. It can also be observed that our proposed model can achieve better detection performance starting from low SNR signals. Integrating the convolutional neural networks with the recurrent networks provides an added advantage for understanding complex signal characteristics.

It can be observed that the probability of a false alarm for the DNN model is higher than the other models. This is because the radio signal data is sequential data, while DNN does not consider learning historical data when computing the output. It does not share parameters across different time steps. On the other hand, RNNs are specifically developed for sequential data with recurrent connections. Therefore, the other models seem to perform better as compared to the DNN model.

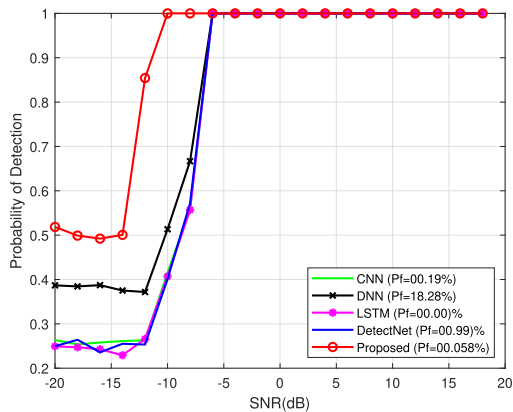


FIGURE 3. Performance on QAM16 signals (64 sample length).

To understand the generalization ability of the proposed model in terms of different signal characteristics, the experiments were performed on different types of signals modulated by various modulation schemes. Fig.7 represents the proposed network’s performance for the QPSK, QAM16, GFSK, and QAM64 modulated signals having a sample length of 128. The  $P_f$  is also denoted alongside the probability of detection. The detection performance implies that the network’s detection performance is almost similar in all the cases, which demonstrates that the model can adapt to the various signal characteristics.

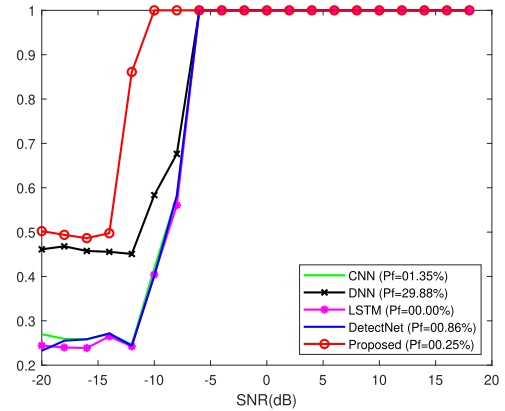


FIGURE 4. Performance on QAM16 signals (128 sample length).

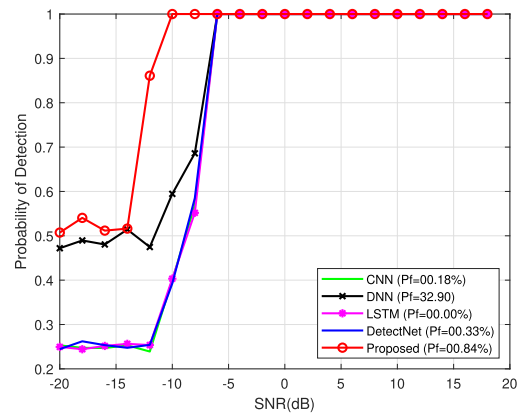


FIGURE 5. Performance on QPSK signals (64 sample length).

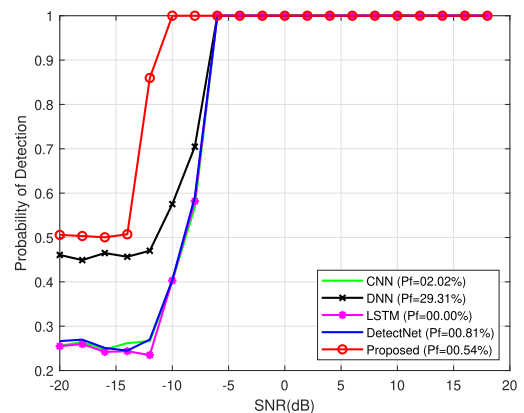


FIGURE 6. Performance on QPSK signals (128 sample length).

The different sample lengths(64 and 128) are considered to represent the performance of the models. It can be observed that there is a difference in the values of  $P_f$  and  $P_d$  as the sample length changes. Our proposed model tends to achieve optimal values in both sample lengths. From tables 2 to 5, it can be observed that the LSTM models gives better  $P_f$  values than the proposed model. This is because the LSTM mechanism is built for sequential data with additional components.

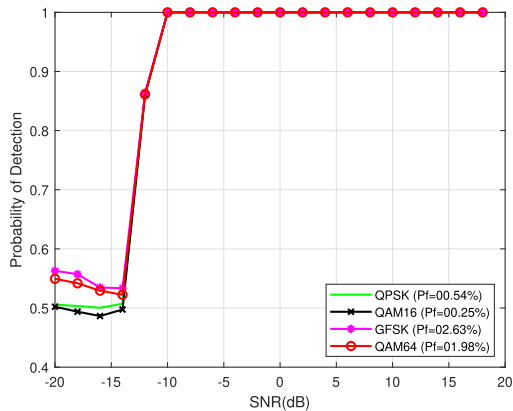


FIGURE 7. Performance on different signals.

However, the number of parameters and computations makes it much more complex. The proposed model can achieve  $P_f$  in the desired range with much less complexity due to the modelling technique and the exploitation of transfer learning. Hence, the different sample lengths are useful in analyzing the performance of the different sensing models.

In addition, a comparison of performance between the proposed model and other models is shown in table 6. The percentage of false alarm and detection probability is shown. It demonstrates that the proposed model performs better under similar conditions. The proposed model indicates a higher detection probability in all the scenarios. This research learns signal features with the help of a neural network which is desirable for spectrum sensing under similar conditions.

TABLE 6. Comparison of performance metrics with previous study.

Previous Study	Model	$P_f$ (%)	$P_d$ (-20 dB)(%)
QAM16 signals (64 sample length)			
Gao et al. [14]	CNN-LSTM	01.44	26.37
Xie et al. [20]	CNN-LSTM	00.45	24.10
Our Previous Study [2]	DLSenseNet	00.00	39.60
<b>Proposed</b>	<b>CNN-RNN</b>	<b>00.05</b>	<b>51.84</b>
QAM16 signals (128 sample length)			
Gao et al. [14]	CNN-LSTM	03.41	27.72
Xie et al. [20]	CNN-LSTM	01.57	25.28
Our Previous Study [2]	DLSenseNet	00.00	40.97
<b>Proposed</b>	<b>CNN-RNN</b>	<b>00.25</b>	<b>50.22</b>
QPSK signals (64 sample length)			
Gao et al. [14]	CNN-LSTM	01.73	26.38
Xie et al. [20]	CNN-LSTM	00.97	26.80
Our Previous Study. [2]	DLSenseNet	00.00	40.79
<b>Proposed</b>	<b>CNN-RNN</b>	<b>00.84</b>	<b>50.73</b>
QPSK signals (128 sample length)			
Gao et al. [14]	CNN-LSTM	02.87	26.32
Xie et al. [20]	CNN-LSTM	01.09	24.81
Our Previous Study. [2]	DLSenseNet	00.00	39.77
<b>Proposed</b>	<b>CNN-RNN</b>	<b>00.54</b>	<b>50.58</b>

The advantage of the proposed model is that it performs better than other models because of its capability to learn the signal features. The network architecture consists of convolution and RNN layers. The CNN layer investigates spatial relations by learning the internal representation of

the radio signal as input data, while RNN layers recognize temporal features in data. The proposed model efficiently combines the advantages of these two deep neural network architectures. Additionally, a transfer learning mechanism is applied to transfer knowledge, which improves the results while reducing complexity. The occupied or vacant spectrum is predicted based on the problem as a signal classification task. The presence of a primary user is accurately detected using the CNN-RNN model.

#### D. COMPLEXITY ANALYSIS

The complexity analysis of the proposed model is discussed here and compared with existing representative models (CNN, DNN, LSTM and DetectNet).

In model, the mini-batch size is  $b$ ,  $I_s$  is the size of the input sequence,  $N_{input}$  is number of input features and  $N_i$  denotes the neurons in  $i$ 'th layer. The expression to sensing model architecture is described as [39], [40]:

$$\begin{aligned}
 Comp_{Proposed} = & 2 \times N_{input} \times N_f \times N_k \\
 & \times \left[ \frac{I_s + 2 \times padding - N_k}{stride} + 1 \right] \\
 & + \left[ \frac{I_s + 2 \times padding - N_k}{stride} + 1 \right] \\
 & \times N_f \times N_{Dense1} + N_{Dense1} \\
 & \times N_{hidden} \times (N_f + N_{hidden} \times h) \\
 & + N_{hidden} \times N_{Dense2} + N_{Dense2} \\
 & \times N_{Dense3} + N_{Dense3} \times N_{output} \quad (8)
 \end{aligned}$$

Equation 8 denotes the contribution of the convolution, flatten, dense, and RNN layers to the complexity of the model. The input enters two convolution blocks which are then forwarded to the dense layer after flattening. Then, the RNN layer exists, the output of which is then forwarded to two dense layers and finally to the output layer. Here,  $N_f$  denotes the number of filters,  $N_k$  represent the kernel size,  $N_{hidden}$  is the number of hidden units and  $N_{output}$  being the number of outputs.  $padding$  is kept as the same, which denotes that enough padding is done so that the output size is the same as the input size. The purpose of padding is to avoid shrinking the output and avoid information loss.  $stride$  denotes the amount of movement the kernel makes upon the data.

Table 7 represents the complexities of all the compared models. The DNN model consists of four layers denoted by  $N_1, N_2, N_3$  and  $N_4$  respectively.  $Comp_{DNN}$  represents the complexity of this model. The next model for comparison is the LSTM model. It consists of one layer of LSTM with  $N_{hidden1}$  number of hidden units.

$Comp_{CNN}$  denotes the computational complexity of the developed CNN model. This model consists of two convolution layers followed by a flatten layer.  $N_{f1}, N_{k1}, padding1, stride1$  represent the filter, kernel, padding and strides for the first convolution layer respectively.  $N_{f2}, N_{k2}, padding2, stride2$  represent the filter, kernel, padding and strides for the second convolution layer



TABLE 7. Complexity comparison.

Model	Complexity
CNN	$Comp_{CNN} = N_{input} \times N_{f1} \times N_{k1} \times \left[ \frac{I_s + 2 \times padding1 - N_{k1}}{stride1} + 1 \right] + N_{f1} \times N_{f2} \times N_{k2} \times \left[ \frac{N_{s1} + 2 \times padding2 - N_{k2}}{stride2} + 1 \right] + \left[ \frac{N_{s1} + 2 \times padding2 - N_{k2}}{stride2} + 1 \right] \times N_{f2} \times N_{Dense1} + N_{Dense1} \times N_{output}$
DNN	$Comp_{DNN} = I_s \times N_{input} \times N_1 + N_1 \times N_2 + N_2 \times N_3 + N_3 \times N_4 + N_4 \times N_{output}$
LSTM	$Comp_{LSTM} = I_s \times N_{hidden1} (4 \times N_{input} + 4 \times N_{hidden1} + 3 + N_{output})$
DetectNet	$Comp_{Detectnet} = N_{input} \times N_{f1} \times N_{k1} \times \left[ \frac{I_s + 2 \times padding1 - N_{k1}}{stride1} + 1 \right] + N_{f1} \times N_{f2} \times N_{k2} \times \left[ \frac{N_{s1} + 2 \times padding2 - N_{k2}}{stride2} + 1 \right] + \left[ \frac{N_{s1} + 2 \times padding2 - N_{k2}}{stride2} + 1 \right] \times N_{f2} \times N_{Dense1} + I_s \times N_{hidden1} \times [4 \times N_{Dense1} + 4 \times N_{hidden1} + 3 + N_{Dense2}] + N_{hidden1} \times N_{Dense2} + N_{Dense2} \times N_{output}$
Proposed	$Comp_{Proposed} = 2 \times N_{input} \times N_f \times N_k \times \left[ \frac{I_s + 2 \times padding - N_k}{stride} + 1 \right] + \left[ \frac{I_s + 2 \times padding - N_k}{stride} + 1 \right] \times N_f \times N_{Dense1} + N_{Dense1} \times N_{hidden} (N_f + N_{hidden} \times h) + N_{hidden} \times N_{Dense2} + N_{Dense2} \times N_{Dense3} + N_{Dense3} \times N_{output}$

respectively. Here,  $N_s1$  denotes the output sequence from the first convolution layer, which enters the second convolution layer as the input sequence. After the flatten layer, one dense layer is present with  $N_{Dense1}$  number of neurons.

The DetectNet model consists of two convolution layers followed by a Dense layer. There is an LSTM layer after the dense layer in which the output from the dense layer and the input sequence is given as input. A dense layer then follows this LSTM layer. This complete computation and associated complexity are then denoted in  $Comp_{Detectnet}$ . Similar to  $Comp_{CNN}$ , the filter, kernel, padding and strides are denoted for the two convolution layers. Since the input sequence is again fed to the LSTM layer, the  $I_s$  term occurs again in the equation.

The sensing time depends upon the model complexity, the amount of data, and the system capacity. As such, the sensing time corresponding to the proposed model comes to approximately 1 second for evaluation data consisting of 29000 instances.

Table 7 depicts the complexities of all the models. Certain layers contribute significantly more to the complexity since the computations are more in number. However, these layers also contribute to better accuracy. In the proposed architecture, it has been tried to have the sensing task performed with minimal complexity and high accuracy. As such, instead of using the LSTM variant, the vanilla RNN architecture has been used. Also, the convolution operation is performed in parallel, which is inspired by the Inception architecture [33].

## V. CONCLUSION

Cognitive radio has become an important technology in wireless regional area networks for machine-to-machine communication. Spectrum sensing is an essential task of cognitive radio. The traditional spectrum sensing methods have some drawbacks in spectrum sensing. AI and deep learning methods have shown promising improvement in spectrum sensing. This research proposed a hybrid model of CNN-RNN for spectrum sensing. Transfer learning is also used for the first time to verify the enhancement in spectrum sensing. The result of the proposed model is compared with

CNN, DNN, LSTM, and DetectNet, and the proposed model showed improvement in spectrum sensing. The probability of detection is improved, and the probability of false alarm is reduced. The complexity of an algorithm is analyzed to show an improvement in performance characteristics.

## REFERENCES

- [1] R. N. Clarke, "Expanding mobile wireless capacity: The challenges presented by technology and economics," *Telecomm. Policy*, vol. 38, nos. 8–9, pp. 693–708, Dec. 2014.
- [2] S. Solanki, V. Dehalwar, and J. Choudhary, "Deep learning for spectrum sensing in cognitive radio," *Symmetry*, vol. 13, no. 1, p. 147, Jan. 2021.
- [3] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Commun.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
- [4] V. Dehalwar, A. Kalam, M. L. Kolhe, and A. Zayegh, "Review of IEEE 802.22 and IEC 61850 for real-time communication in smart grid," in *Proc. Int. Conf. Comput. Netw. Commun. (CoCoNet)*, Dec. 2015, pp. 571–575.
- [5] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [6] V. Dehalwar, A. Kalam, M. L. Kolhe, and A. Zayegh, "Compliance of IEEE 802.22 WRAN for field area network in smart grid," in *Proc. IEEE Int. Conf. Power Syst. Technol. (POWERCON)*, Sep. 2016, pp. 1–6.
- [7] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 116–130, Mar. 2009.
- [8] S. Solanki, V. Dehalwar, and J. Choudhary, "Cooperative spectrum sensing for PU detection in cognitive radio using SVM," in *Data Engineering and Communication Technology*. Cham, Switzerland: Springer, 2021, pp. 61–69.
- [9] W. Lee and D.-H. Cho, "Enhanced spectrum sensing scheme in cognitive radio systems with MIMO antennae," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1072–1085, Mar. 2011.
- [10] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, Oct. 2012.
- [11] C. Lin, Q. Chang, and X. Li, "A deep learning approach for MIMO-NOMA downlink signal detection," *Sensors*, vol. 19, no. 11, p. 2526, Jun. 2019.
- [12] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: A survey with a demonstration," *IEEE Access*, vol. 8, pp. 67366–67376, 2020.
- [13] C. Lin, Q. Chang, and X. Li, "A deep learning approach for MIMO-NOMA downlink signal detection," *Sensors*, vol. 19, no. 11, p. 2526, 2019.
- [14] 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.
- [15] W. Lee, M. Kim, and D. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3005–3009, Mar. 2019.

- [16] 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.
- [17] B. Soni, D. K. Patel, and M. López-Benítez, "Long short-term memory based spectrum sensing scheme for cognitive radio using primary activity statistics," *IEEE Access*, vol. 8, pp. 97437–97451, 2020.
- [18] R. Sarikhani and F. Keynia, "Cooperative spectrum sensing meets machine learning: Deep reinforcement learning approach," *IEEE Commun. Lett.*, vol. 24, no. 7, pp. 1459–1462, Jul. 2020.
- [19] B. M. Pati, M. Kaneko, and A. Taparugssanagorn, "A deep convolutional neural network based transfer learning method for non-cooperative spectrum sensing," *IEEE Access*, vol. 8, pp. 164529–164545, 2020.
- [20] J. Xie, J. Fang, C. Liu, and X. Li, "Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2196–2200, Oct. 2020.
- [21] Q. Peng, A. Gilman, N. Vasconcelos, P. C. Cosman, and L. B. Milstein, "Robust deep sensing through transfer learning in cognitive radio," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 38–41, Jan. 2020.
- [22] Z. Chen, Y.-Q. Xu, H. Wang, and D. Guo, "Deep STFT-CNN for spectrum sensing in cognitive radio," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 864–868, Mar. 2021.
- [23] R. Liu, Y. Ma, X. Zhang, and Y. Gao, "Deep learning-based spectrum sensing in space-air-ground integrated networks," *J. Commun. Inf. Netw.*, vol. 6, no. 1, pp. 82–90, Mar. 2021.
- [24] S. Chandhok, H. Joshi, A. V. Subramanyam, and S. J. Darak, "Novel deep learning framework for wideband spectrum characterization at sub-Nyquist rate," *Wireless Netw.*, vol. 27, no. 7, pp. 4727–4746, Oct. 2021.
- [25] H. Xing, H. Qin, S. Luo, P. Dai, L. Xu, and X. Cheng, "Spectrum sensing in cognitive radio: A deep learning based model," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 1, p. e4388, Jan. 2022.
- [26] S. Pandit and G. Singh, *Spectrum Sharing in Cognitive Radio Networks*. Cham, Switzerland: Springer, 2017.
- [27] C.-H. Liu, *Traffic-Aware Spectrum Sharing Protocols*. Los Angeles, CA, USA: Univ. California, Los Angeles, 2015.
- [28] E. Hossain, D. Niyato, and Z. Han, *Dynamic Spectrum Access and Management in Cognitive Radio Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [29] Z. Wang and W. Zhang, *Opportunistic Spectrum Sharing in Cognitive Radio Networks*. Cham, Switzerland: Springer, 2015.
- [30] B. Brahma and R. Wadhvani, "Solar irradiance forecasting based on deep learning methodologies and multi-site data," *Symmetry*, vol. 12, no. 11, p. 1830, Nov. 2020.
- [31] A. K. Nandanwar and J. Choudhary, "Semantic features with contextual knowledge-based web page categorization using the GloVe model and stacked BiLSTM," *Symmetry*, vol. 13, no. 10, p. 1772, Sep. 2021.
- [32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.
- [33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014, *arXiv:1409.4842*.
- [34] X.-F. Xu, L. Zhang, C.-D. Duan, and Y. Lu, "Research on inception module incorporated Siamese convolutional neural networks to realize face recognition," *IEEE Access*, vol. 8, pp. 12168–12178, 2020.
- [35] P. Kumar, S. Batra, and B. Raman, "Deep neural network hyper-parameter tuning through twofold genetic approach," *Soft Comput.*, vol. 25, no. 13, pp. 8747–8771, Jul. 2021.
- [36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [37] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," *Commun. Comput. Inf. Sci.*, vol. 629, pp. 213–226, Sep. 2016.
- [38] *IEEE Draft Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 22: Cognitive Radio Wireless Regional Area Networks (WRAN) Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the Bands That Allow Spectrum Sharing Where the Communications Devices May Opportunistically Operate in the Spectrum of the Primary Service*, IEEE P802.22/D6.0.0, May 2019, pp. 1–1455.
- [39] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1553–1565, Aug. 2014.
- [40] P. J. Freire, Y. Osadchuk, B. Spinnler, A. Napoli, W. Schairer, N. Costa, J. E. Prilepsky, and S. K. Turitsyn, "Performance versus complexity study of neural network equalizers in coherent optical systems," *J. Lightw. Technol.*, vol. 39, no. 19, pp. 6085–6096, Oct. 1, 2021.



**SURENDRA SOLANKI** received the B.E. degree in information technology from the IET DAVV, Indore, India, and the M.Tech. degree from the MANIT, Bhopal, India, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. His research interests include wireless communications, cognitive radio, machine learning, and deep learning.



**VASUDEV DEHALWAR** (Senior Member, IEEE) received the B.E. degree from the MANIT, Bhopal, India, the M.Tech. degree from the Indian Institute of Technology, Kharagpur, India, and the Ph.D. degree from Victoria University, Melbourne, VIC, Australia. He has more than 25 years of teaching and research experience. He currently works as an Assistant Professor at the MANIT. He was the Principal Investigator of the Information Security Educational and Awareness Project,

from 2006 to 2013. He organized many short-term courses for working professionals in Bhopal. His research interests include wireless communications and networking, information security, smart grid and the IoT, and software-defined networking. He is a reviewer of many IEEE and Elsevier journal publications.



**JAYTRILOK CHOUDHARY** (Member, IEEE) received the B.E. degree from MITM Indore, the M.Tech. degree from the IIT Roorkee, India, and the Ph.D. degree from the MANIT, Bhopal, India. He currently works as an Assistant Professor at the MANIT. His research interests include information retrieval, machine learning, image processing, and cognitive radio networks.



**MOHAN LAL KOLHE** (Senior Member, IEEE) is currently a Full Professor of smart grid and renewable energy at the Faculty of Engineering and Science, University of Agder, Norway. He is a leading renewable energy technologist with three decades of the academic experience at an international level. He has held various academic positions at prestigious universities, such as University College London, U.K., a branch in Australia; the University of Dundee, U.K.;

the University of Jyväskylä, Finland; and the Hydrogen Research Institute, Trois-Rivières, QC, Canada. He was also a Board Member of the Government of South Australia's Renewable Energy Board (2009–2011) as an advisor on renewable energy policies. He is an Expert Evaluator of projects for funding at various international research councils, such as European Commission; Erasmus+ Higher Education–International Capacity Building, Royal Society London, U.K., and the Engineering and Physical Sciences Research Council (EPSRC U.K.). He has delivered keynote addresses and expert lectures at many conferences and workshops.



**KOKI OGURA** (Senior Member, IEEE) received the B.S.Eng., M.S.Eng., and Dr.Eng. degrees in electrical engineering from Yamaguchi University, Yamaguchi, Japan, in 2000, 2002, and 2005, respectively. During his career at Kawasaki Heavy Industries, Ltd., Kobe, Japan, from 2005 to 2016, he was engaged in the development of the next-generation, renewable energy systems through battery-driven, low-floor light rail vehicle (LRV), and the advanced wayside energy storage system

for electric railways in New York City Transit (NYCT). He is currently an Associate Professor of electrical engineering at Kyushu Sangyo University, Fukuoka, Japan. His current research interests include circuit design and integration of high-frequency soft switching power converters, power electronics applications in home appliances, transportation, and renewable energy systems.

• • •