Ain Shams Engineering Journal 15 (2024) 102505

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

Analysis of spectrum sensing using deep learning algorithms: CNNs and RNNs

Arun Kumar [a], Nishant Gaur [b], Sumit Chakravarty [c], Mohammed H. Alsharif [d], Peerapong Uthansakul [e,*], Monthippa Uthansakul [e]

[a] Department of Electronics and Communication Engineering, New Horizon College of Engineering, Bengaluru, India
[b] Department of Physics, JECRC University, Jaipur, India
[c] Department of Electrical Engineering and Computer Engineering, Kennesaw State University, Kennesaw, GA, USA
[d] Department of Electrical Engineering, College of Electronics and Information Engineering, Sejong University, Seoul 05006, Republic of Korea
[e] School of Telecommunication Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand

ARTICLE INFO

Keywords:
CNNs
RNNs
Spectrum sensing
Probability of false alarm

ABSTRACT

Spectrum sensing is a critical component of cognitive radio systems, enabling the detection and utilization of underutilized frequency bands. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), in particular, have showed promise in recent years for enhancing the precision and effectiveness of spectrum sensing. This paper presents an overview of spectrum sensing using CNNs and RNNs and their performance in cognitive radio systems. Furthermore, the paper delves into the spectrum sensing performance of RNNs, particularly for processing time-series data. RNNs are capable of capturing temporal dependencies in sequential data, which is essential for spectrum sensing tasks where signals evolve over time. Further, the parakeets such as probability of detection (Pd), Probability of false alarm (PFA), Bite Error rate (BER) and Power spectral density (PSD) are analysed for spectrum sensing algorithms. RNNs and CNNs, two examples of deep learning methods (DLM), performed better than more traditional approaches.

1. Introduction

The wireless communication landscape is rapidly evolving, with an ever-increasing demand for bandwidth due to a surge in connected devices and data-intensive applications. Traditional spectrum sensing methods, pivotal for the optimal use of radio frequency (RF) spectrum in cognitive radio networks, often fall short in dynamic and noisy environments. Deep learning, with its inherent ability to decipher complex patterns, presents a transformative solution [1]. CNNs, initially designed for image processing, can adeptly handle the spatial characteristics of RF signals, turning them into detectable patterns. RNNs, on the other hand, excel in processing sequences, making them ideal for tracking the temporal dynamics of spectrum use. Together, these algorithms promise to revolutionize spectrum sensing, enabling more efficient and adaptive use of the RF spectrum, meeting the demands of modern wireless communication while optimizing spectrum resources [2]. Spectrum sensing is a pivotal operation in the world of cognitive radio networks. Its primary goal is to detect vacant frequency bands, known as spectral holes, in real-time so that secondary (unlicensed) users can utilize them without causing interference to primary (licensed) users. This process is crucial for optimizing the use of the limited RF spectrum, especially given the rapid expansion of wireless devices and services [3]. Traditional spectrum sensing techniques, such as energy detection, matched filtering, and cyclostationary feature detection, have their set of challenges. For instance, energy detection, although simple, struggles in environments with uncertain noise levels [4]. Matched filtering provides better reliability but requires a priori knowledge of the primary user's signal. Cyclostationary feature detection, while robust against noise, is computationally intensive. Pilot-based sensing relies on the transmission of pilot signals periodically. These known signals are used to approximate the channel characteristics and detect the existence of primary users. By comparing the received pilot signals with the expected pilot signals, devices can identify the availability of the active users in the spectrum [5]. This technique is commonly used in cellular systems, where base stations periodically transmit pilot signals to facilitate channel estimation and signal detection. Statistical methods, such as

* Corresponding author.
E-mail addresses: nishant.gaur@jecrcu.edu.in (N. Gaur), schakra2@kennesaw.edu (S. Chakravarty), sharif@sejong.ac.kr (M.H. Alsharif), uthansakul@sut.ac.th (P. Uthansakul), mpt@sut.ac.th (M. Uthansakul).

https://doi.org/10.1016/j.asej.2023.102505
Received 13 July 2023; Received in revised form 22 August 2023; Accepted 16 September 2023
Available online 30 September 2023

Bayesian methods and hypothesis testing, are also employed in spectrum sensing. These techniques use statistical models and decision theory to make informed decisions about the existence or time off of the signals based on observed data and prior knowledge [6]. Bayesian methods, for example, use prior probabilities and update them based on new evidence to estimate the probability of signal presence. Hypothesis testing involves formulating null and alternative hypotheses and making decisions based on the statistical significance of observed data. It's important to note that spectrum sensing faces various challenges. The wireless environment is dynamic, with signal characteristics varying over time and space. Moreover, the presence of noise, interference, and fading further complicates accurate signal detection. Spectrum sensing algorithms must be robust and adaptable to handle these challenges. In recent years, spectrum sensing has gained significant attention due to the emergence of cognitive radio networks [7]. In such a landscape, the emergence of deep learning techniques, specifically CNNs and RNNs, offers a geat potential. Rooted in image processing, CNNs have become a cornerstone in various applications due to their ability to detect hierarchical patterns in data. The core concept behind CNNs is the work of convolutional layers, where a set of learnable filters slide (or convolve) around the input data (in this case, RF signals) to produce feature maps [8]. These feature maps undergo pooling operations, which reduce their spatial dimensions while retaining crucial information. The stacked convolutional and pooling layers enable the network to recognize progressively complex patterns, making it adept at identifying intricate characteristics of signals in spectrum sensing [9]. In the context of spectrum sensing, RF signals can be visualized as one-dimensional 'images' or sequences, allowing CNNs to extricate spatial attributes effectively. This capability facilitates the recognition of nuanced frequency usage patterns, thereby enhancing the detection exactness of primary users in various spectral conditions. RNNs are intended to handle sequential data, a property that makes them incredibly applicable for tasks that necessitate memory or the consideration of past information [10]. In RNNs, the output from one node is fed back into the network as an input, enabling the network to maintain a form of 'memory' about previous inputs [11]. This looped feedback mechanism allows RNNs to process sequences of data, making them invaluable for applications that involve time-series or sequential data. When applied to spectrum sensing, RNNs focus on the temporal aspect of spectrum usage. Given their ability to remember past states, they are adept at predicting future spectrum utilization based on historical data [12,13]. Variants of RNNs, such as Long Short-Term Memory (LSTM) networks, further enhance this memory capability, allowing for the recognition of long-term dependencies in the time-series data of spectrum usage. Marrying the strengths of CNNs and RNNs offers a comprehensive approach to spectrum sensing. While CNNs efficiently process the spatial attributes of the RF spectrum, identifying signal patterns across different bands, RNNs capture the temporal dynamics of spectrum usage, providing insights into how frequency bands are used over time [14,15]. This holistic methodology paves the way for a more intelligent, adaptive, and efficient spectrum sensing mechanism, capable of navigating the complexities of modern wireless communication landscapes. In essence, the blend of traditional spectrum sensing with the prowess of CNNs and RNNs is set to usher in a new era of cognitive radio networks, optimizing spectrum usage and accommodating the ever-growing demands of wireless communication [16]. In this article, we aim to identify the availablity or non- availablity of signals within a particular frequency band, enabling cognitive radios or other wireless devices to opportunistically access and utilize vacant or underutilized slices of the spectrum. This procedure enhenced the spectrum efficiency, increased network capacity, and better utilization of available resources. Further, the proposed algorithm adaptively allocates bands to where demand is highest and distinguishing between genuine signals and noise, especially in low signal-to-noise ratio environments. The key contributions of the article are given below:

- Traditional spectrum sensing techniques often falter in rapidly changing or non-static wireless environments. The proposed algorithms, with their capability to learn and adapt from vast amounts of data, have showcased significant improvement in detection accuracy, even under hostile circumstances like low SNR, fading, and shadowing. They can discern intricate patterns within data, enabling them to differentiate between actual signals and noise with higher precision.
- Traditional methods, such as matched filtering, require a priori knowledge of primary user signals. the proposed model, once trained, can generalize their learning to new, unseen conditions, reducing the need for explicit prior knowledge. They can dynamically adapt to different scenarios, making them mainly beneficial for real-world deployments where the RF environment is unpredictable.
- With the proliferation of wireless devices and the emerging spectrum bands (like millimeter-wave bands), the amount of spectral data to be processed has grown exponentially. The proposed algorithms can efficiently process this vast data, extracting relevant features and making timely decisions. This scalability ensures that as the wireless landscape evolves, deep learning-based spectrum sensing methods remain relevant and effective.

1.1. *Motivation for deep learning method (DLM)*

The motivation for using DLM techniques in spectrum sensing stems from several advantages and potential benefits that these methods offer. Some key motivations are given below [17-19]:

- Improved Feature Representation: DLM have the ability to mechanically acquire complex and hierarchical feature representations from raw data. Traditional spectrum sensing techniques often rely on handcrafted feature extraction algorithms that may not acquire all the relevant information in the signals. DLM can study directly from the data and capture both low-level and high-level features, potentially leading to improved detection performance.
- Adaptability and Flexibility: DLM are extremely adaptable and can be trained to detect a wide range of signals. They can learn from diverse datasets and adapt to different signal features, making them appropriate for dynamic and heterogeneous wireless environments. This adaptability is particularly valuable in scenarios where signal characteristics change over time or in environments with multiple types of signals.
- Robustness to Noise and Interference: DLM have the potential to be more robust to noise and interference compared to traditional spectrum sensing techniques. They can learn to differentiate between signal patterns and noise patterns, and their ability to capture complex features can help in mitigating the effects of interference on signal detection.
- Scalability and Generalization: Deep learning models have demonstrated good scalability and generalization capabilities. These models can be generalised to many circumstances once they have been developed by using previously unknown data with similar properties. This scalability is predominantly beneficial when dealing with large-scale spectrum sensing applications or deployments.
- End-to-End Learning: DLM enable end-to-end learning, which means that they can directly learn the mapping from raw input signals to desired outputs, without relying on manual feature engineering or intermediate processing steps. This eliminates the requirement for domain expertise in signal processing and simplifies the overall spectrum sensing pipeline.
- Potential for Real-Time Processing: DLM can be optimized for efficient inference, allowing for real-time or near-real-time processing of incoming signals. This is essential in applications where timely spectrum sensing is required, such as dynamic spectrum access and cognitive radio systems.

- It's crucial to remember that while DLM have potential for spectrum sensing, they also have their own set of difficulties. These include the need for large labelled datasets, potential overfitting, interpretability of the learned representations, and computational complexity. Nonetheless, the motivations for utilizing deep learning in spectrum sensing arise from their potential to improve detection accuracy, adaptability to dynamic environments, and ability to handle complex signal characteristics. Table.1. indicate the abbreviation used in the proposed article.

2. Related work

We conduct a thorough literature study on deep learning algorithms for spectrum sensing and provide a summary of the pertinent papers and research in Table 2.

3. Proposed system model

CNNs are a type of DLM that have confirmed to be effective in various domains, including image and signal processing. When applied to spectrum sensing, CNNs can extricatet important attributes from raw signal data and accurately classify whether a particular frequency band is occupied or unoccupied. CNNs are particularly well-suited for analyzing signals because they can robotically acquire local spatial dependencies within the data [28]. The key components of a CNN are convolutional layers, pooling layers, and fully connected layers. Convolutional layers are responsible for extracting features from input data. In the context of spectrum sensing, the input data would typically be the raw signal samples or a transformed representation of the signal. Each convolutional layer consists of multiple filters or kernels, which are small-sized matrices that slide over the input data to perform convolutions. These convolutions help capture local patterns and correlations in the signal. Pooling layers are used to down-sample the output of convolutional layers, reducing the dimensionality of the data [29]. This down-sampling helps in retaining the important features while discarding redundant information. Max pooling is a common type of pooling operation, where the maximum value within each pooling region is selected. Fully connected layers are responsible for making the final classification decision based on the learned features. They take the output from the previous layers and apply a set of weighted connections to produce a final prediction. The fully connected layers in the CNN can have multiple neurons, and the weights associated with each neuron are learned during the training phase [30]. The process of using CNNs for spectrum sensing involves the following steps [31,32]:

- Data Pre-processing: The raw signal data is typically pre-processed to normalize amplitudes, rescale the data, or perform other necessary transformations. This step ensures that the input data is suitable for the CNN model.
- Training Data Preparation: A large dataset of labeled signals is required for training the CNN. This dataset should contain examples of both occupied and unoccupied frequency bands. The labeled data is divided into training and validation sets.
- Model Architecture Design: The architecture of the CNN is designed based on the specific spectrum sensing task and requirements. The number and arrangement of convolutional layers, pooling layers, and fully connected layers are determined. The selection of appropriate activation functions and regularization techniques is also important.
- Training: The CNN model is trained using the labeled dataset. The training process involves forward propagation, where the input data passes through the network layers to generate predictions. The loss or error between the predicted outputs and the ground truth labels is calculated. Backpropagation is then used to update the model's weights and optimize the loss function using gradient descent or other optimization algorithms.
- Validation and Fine-tuning: During training, the model's performance is evaluated using the validation set to ensure that it

Table 1
Abbreviations used in the paper.

CNNs	Convolutional neural networks
RNNs	Recurrent neural networks
Pd	Pprobability of detection
PFA	Probability of false alarm
BER	Bite error rate
PSD	Power spectral density
DLM	Deep learning methods
RF	Radio frequency
CSS	Cooperative spectrum sensing
ELM	Extreme learning machine
ED	Energy detection
MF	Matched filter

Table 2
Literature Review.

References	Remarks
[20]	The authors in [20] suggested a deep learning-based signal detector that delivers state-of-the-art detection performance with no prior knowledge of channel state information or background noise by using the underlying structure information of the modulated signals. It should be highlighted that the provided approach significantly outperforms other established cooperative sensing methods.
[21]	The spectrum-sensing problem is turned into an image identification challenge in [21], and machine learning is used to differentiate between noise and the real world. signal's existence Using just a short training set of a few hundred samples. The efficacy of the CNN detector is then contrasted with previous published results of machine learning employed for signal identification as well as the throughput of more conventional energy detection.
[22]	The article in [22] makes an effort to evaluate the effectiveness of cooperative spectrum sensing (CSS), which uses deep learning as a method for data fusion. We investigate how well convolutional neural network-based CSS performs under dynamic channel conditions.
[23]	In [23], a hybrid deep learning (DL)-based spectrum sensing method that substantially learns the statistical time series spectrum data was developed. The innovative testbed that gets the raw spectrum data under various frequency technologies and SNR conditions. Performance indicators such as Pd, and Pfa, are measured and compared to other learning model-based spectrum sensing methods that are currently available. Researches show that the suggested framework beat the other approach in-terms of high sensing accuracy and a high detection ratio, even when the SNR was low.
[24]	For cognitive radio systems, this study in [24] offered a brand-new spectrum sensing method. The suggested strategy makes use of the recurrent neural network (RNN), a well-liked deep learning methodology, to ascertain the spectrum's emptiness. The suggested method calculates the principal user's (PU) spectrum occupancy. Any knowledge of the PU signal characteristic is not used; instead, the received signal's energy is observed.
[25]	In [25], an Extreme Learning Machine (ELM)-based process for cooperative spectrum sensing (CSS) is suggested. ELMs are feedforward neural networks in which only the output weights are optimised and not the hidden layer parameters. Both a fading environment and a non-fading environment were simulated. These findings show that ELM may be better than conventional techniques.
[20]	In [20], the authors suggest using DRL to solve the cognitive radio network problem of CSS. The simulation outcomes establish that, while greatly decreasing computing time, the suggested problem formulation employing the CQL algorithm may achieve similar detection accuracy to existing state-of-the-art approaches for CSS.
[26]	The authors in [26] conduct trials with various signal-to-noise ratios to validate the improved performance of the suggested approach. Finding the occurrence of a modulation format demonstrates that the PU signal will have an associated modulation format.
[27]	The findings in [27] demonstrate that the suggested technique is successful at suppressing the sensing cost by shortening the sensing time and utilising conventional fusion principles. Additionally, the Using changed sensing samples as the basis for a global decision, the fusion centre (FC) achieves reduced energy consumption, better throughput, and enhanced detection with a low mistake probability.

generalizes well to unseen data. If necessary, hyperparameters can be fine-tuned to optimize the model's performance.

- Testing and Inference: Once trained, the CNN model can be used for spectrum sensing on unseen data. The input signal samples are fed into the model, and the output predictions indicate whether the frequency band is occupied or unoccupied.

CNNs have shown promising results in spectrum sensing tasks by automatically learning discriminative features directly from the signal data. They can handle complex signal patterns, noise, and interference, making them suitable for real-world spectrum sensing applications. However, it's important to carefully design the CNN architecture, select appropriate hyperparameters, and validate the model's performance to ensure accurate and reliable spectrum sensing.

3.1. Mathematical CNNs

A mathematical model for CNNs in a spectrum sensing system can be characterized by means of the following equations and notations:

Let $x \in \mathbb{R}^{(}N \times M)$ represent the input signal, where N is the number of samples and M is the number of channels (e.g., for a single channel signal, M = 1). Let $W^l \in \mathbb{R}^{K \times K \times D^{l-1}} \times D^l$ denote the weight matrix of the l-th convolutional layer, where K is the filter size $, D^{l-1})$ is the input channels, and D^l is the output channels. The convolution operation for the $l-th$ layer can be written as:

$$z^l = f\sum_{i=1} D^{l-1}\left(x^* W_i^l\right) + b^l \tag{1}$$

where * represents the convolution operation, f is the activation function, and b^l is the bias term for the l_{th} layer. The pooling operation, typically max pooling, down-samples the feature maps to reduce dimensionality and extract key features. Let P^l denote the pooling operation for the l_{th} layer, and P_{Size^l} be the size of the pooling window.

The output of the pooling operation can be expressed as:

$$x^{'l} = P^l\left(z^l, P_{Size^l}\right). \tag{2}$$

Let $W_{F_{cl-1}} \in \mathbb{R}^{(}W_{F_{cl-1}}) \times H_{F_{cl}})$ represent the weight matrix for the l_{th} fully connected layer, where $H_{F_{cl-1}}$ and $H_{F_{cl}}$ are the sizes of the input and output layers, respectively. The output of the l-th fully connected layer is computed as:

$$Z_{F_{cl-1}} = W_{F_{cl}} {}^* x_{F_{cl-1}} + b_{F_{cl}} \tag{3}$$

where $x_{F_{cl-1}}$ represents the input to the l_{th} fully connected layer, $b_{F_{cl}}$ is the bias term, and f is the activation function. For binary classification in spectrum sensing, the output layer typically uses a sigmoid activation function. The predicted output for a given input sample × is given by:

$$\widehat{y} = \sigma\left(\left(Z_{F_{cL}}\right)\right), \tag{4}$$

where L is the index of the last fully connected layer and σ is the sigmoid activation function. The selection of loss function depends on the precise problem and can include cross-entropy loss, mean squared error, or others.

Let $L(y, \widehat{y}))$ represent the loss function that measures the discrepancy between the predicted output $\widehat{y}$ and the true label y. The CNN model is trained by minimizing the loss function using techniques such as gradient descent or its variants. The weights and biases are updated iteratively by computing the gradients of the loss function with respect to these parameters and adjusting them accordingly. This mathematical model provides a framework for understanding the operations and computations involved in CNN-based spectrum sensing systems. However, note that the specific details of the architecture, activation functions, and loss function may fluctuate reliant on the design choices and requirements of the spectrum sensing application. The algorithm for the suggested CNNs is indicated in Table 3.

3.2. RNNs

RRNs are a form of neural network architecture specifically designed for sequential data processing. They have been successfully applied to various tasks involving time-series data, including natural language processing and speech recognition [33]. RNNs are also suitable for spectrum sensing, where the input signals are often represented as a sequence of samples over time. The key feature of RNNs is their capability to acquire temporal needs and maintain a memory of past information. This is attained through recurrent connections, where the hidden state of the network is fed back into itself at each time step. This recurrent connection allows RNNs to process sequential data by considering both the current input and the previous states [34]. In the framework of spectrum sensing, the input to an RNN can be a time series of signal samples, representing the changing characteristics of the signal over time. Here's a high-level explanation of how RNNs work in spectrum sensing [35,36]:

Input Representation:

- The raw signal data is typically divided into fixed-length time windows or frames, where each frame contains a sequence of samples.
- Each sample in the sequence is considered as an input at a particular time step.

Hidden State Update:

- At each time step t, the RNN takes the input sample at that time step, along with the previous hidden state, and computes a new hidden state.
- The hidden state is updated using a combination of the current input and the previous hidden state, applying an activation function (such

Table 3
CNNs Algorithm.

Algorithm for CNN based spectrum sensing
Data Pre-processing:
Normalize the raw signal data to a specific range (e.g., [-1, 1]).
Divide the dataset into training and validation sets.
Model Architecture Design:
Determine the number of layers based on the problem requirements.
Specify the number of filters, filter sizes, and pooling sizes for each layer.
Choose appropriate activation functions (e.g., ReLU) for the hidden layers.
Define the output layer with a sigmoid or softmax activation function.
Initialize the CNN Model:
Initialize the weights and biases of the CNN layers.
Training Loop:
Iterate over the training dataset multiple times (epochs).
Shuffle the training data before each epoch to avoid biased training.
For each training sample:
Put forward propagation to use.
Use the specified filters and activation function to apply the convolution operation on the input signal.
Utilise pooling procedures to downsample the feature maps, such as max pooling*.*
Make a 1D vector out of the obtained feature maps.
Provide the completely connected layers with the flattened vector and the necessary activation functions.
Determine the difference in output between the projected result and the ground truth label.
Apply backpropagation using an optimisation technique (such as stochastic gradient descent) to update the weights and biases of the CNN layers.
Validation:
Evaluate the model's performance on the validation dataset at the end of each epoch.
To evaluate the effectiveness of the model, compute metrics such as accuracy, precision, recall, or F1 score.
Testing and Inference:
Once training is complete, use the trained model for spectrum sensing on unseen data.
Pre-process the test signal data in the similar method as the training data.
Perform forward propagation through the trained CNN model to obtain estimates for each test sample.

as tanh or ReLU) to capture the non-linear relationships between the inputs and the hidden state.

Output Generation:

- Depending on the specific task, the RNN may generate an output at each time step or only at the final time step.
- The output can be a prediction of whether the frequency band is occupied or unoccupied, or it can represent some other relevant information related to spectrum sensing.

Training and Backpropagation:

- RNNs are trained using a variant of backpropagation called Backpropagation Through Time (BPTT).
- BPTT involves propagating the gradients of the loss function through each time step of the RNN, adjusting the weights and biases to minimize the loss.
- The gradients are computed by comparing the predicted output with the ground truth at each time step.

Long Short-Term Memory (LSTM) Networks:

- To address the vanishing gradient problem in traditional RNNs, LSTM networks have been introduced.
- LSTM networks have a more sophisticated architecture that includes memory cells and gating mechanisms, enabling them to capture long-term dependencies more effectively.
- The use of RNNs in spectrum sensing allows the network to consider the temporal dynamics of the signal and make predictions based on the sequence of samples. By maintaining a hidden state that represents the historical information, RNNs can capture dependencies and patterns in the time series data, enhancing the accuracy of spectrum sensing tasks.
- It's important to note that the specific design choices, hyperparameter tuning, and network architecture may vary depending on the application and dataset. Nonetheless, RNNs provide a powerful framework for modeling sequential data in spectrum sensing systems.

3.2.1. *Mathematical model of RNNs*

A mathematical model for RNNs in spectrum sensing involves capturing the computations performed at each time step and the update rules for the hidden state. Here's a mathematical formulation of a basic RNN model for spectrum sensing:

Let $x_t \in \mathbb{R}^D$ represent the input signal at time step t, where D is the number of input features or dimensions. The hidden state $h_t \in \mathbb{R}^H$ at time step t is computed as a function of the input signal and the previous hidden state:

$$h_t = f(W_{H\times H}{}^*h_{t-1} + W_{hx}{}^*x_t + b_{th}), \tag{5}$$

where $W_{HxH} \in \mathbb{R}^(H \times H)$ is the weight matrix for the recurrent connections, $W_{hx} = \in \mathbb{R}^{(H\times H)}$ is the weight matrix for the input connections, $b_h \in b_{th}\mathbb{R}^{(H)}$ is the bias term, and f is the activation function (e.g., tanh or ReLU). The output $y_t \in \mathbb{R}^{(O)}$ at time step t is computed based on the hidden state:

$$y_t = g\left(W_{yh}{}^*h_t + b_y\right) \tag{6}$$

where $W_{yh} = \in \mathbb{R}^{(O\times H)}$ is the weight matrix connecting the hidden state to the output, $b_y = \in \mathbb{R}^O$, b is the bias term, and g is the activation function.

The choice of the loss function depends on the specific problem and can include cross-entropy loss, mean squared error, or others. Let $L(y, y_t)$ represent the loss function that measures the discrepancy between the predicted output y_t and the true label y. RNNs are trained using variants of backpropagation through time (BPTT) or gradient descent algorithms. The weights and biases are updated iteratively by computing the gradients of the loss function with respect to these parameters and adjusting them accordingly. This mathematical model represents the forward pass of a basic RNN for spectrum sensing. It captures the update of the hidden state at each time step and the generation of the output based on the hidden state. During training, the model parameters are adjusted to minimize the loss function. The algorithm for RNNS is given in Table.4. Table 5 highlights the characteristics of the proposed CNNs nad RNNs methods.

3.3. *Energy detection (ED)*

ED is a well-liked method for detecting the existence or absence of principal users in a frequency range in spectrum sensing. It is a non-coherent technique that doesn't necessitate prior knowledge of the fundamental signal properties [37]. Instead, it makes use of a predetermined threshold to compare the received signal's power or energy. The basic principle of ED is straightforward. The received signal is sampled and squared to obtain the instantaneous power values. These power samples are then averaged over a specific time interval to obtain the estimated energy. The energy is compared to a predetermined threshold, and if the energy exceeds the threshold, it is classified as the presence of a primary user in the spectrum [38]. To implement energy detection, the following steps are typically involved, given in Fig. 1: (see Fig. 2).

Signal Acquisition:

- The received signal is captured using a receiver or a software-defined radio (SDR).

Table 4
RNNS algorithm.

Data Pre-processing:
Divide the raw signal data into fixed-length time windows or frames.
Normalize the signal samples within each frame to a specific range (e.g., [-1, 1]).
Model Architecture Design:
Choose the type of RNN architecture, such as Vanilla RNN, Long Short-Term Memory (LSTM), or Gated Recurrent Unit (GRU).
Determine the number of recurrent layers and the number of units (hidden state size) in each layer.
Specify the activation function for the recurrent layers.
Initialize the RNN Model:
Initialize the weights and biases of the recurrent layers.
Training Loop:
Iterate over the training dataset multiple times (epochs).
Shuffle the training data before each epoch to avoid biased training.
Initialize the hidden state of the RNN.
For each training sample:
Perform forward propagation:
Feed the input sequence into the RNN, updating the hidden state at each time step.
Calculate the loss between the predicted output and the ground truth label.
Perform backpropagation through time to compute the gradients of the loss with respect to the RNN's parameters.
Update the weights and biases of the RNN using an optimization algorithm (e.g., gradient descent or its variants).
Validation:
After each epoch, evaluate the model's performance on the validation dataset.
Calculate metrics such as accuracy, precision, recall, or F1 score to assess the model's performance.
Testing and Inference:
Once training is complete, use the trained RNN model for spectrum sensing on unseen data.
Pre-process the test signal data in the same way as the training data.
Initialize the hidden state of the RNN.
Feed the input sequence into the RNN and obtain the predicted output at each time step.
Apply a threshold to the output predictions to determine if a frequency band is occupied or unoccupied.

Table 5
Highlights the characteristics and advantages of the cnns and rnns.

Parameters	Characteristics
Data Type Suitability	While CNNs excel at tasks involving spatial hierarchies, like image or signal processing, RNNs shine with sequential data, making them a top choice for time-series analysis or any task involving sequences.
Processing Style	NNs focus on spatial features through convolution operations, creating a depth of layers to understand data hierarchically. In contrast, RNNs rely on loops to process sequences, keeping track of prior information to make decisions about current data.
Scalability and Complexity	CNNs, with their deep architectures, often involve a significant number of parameters, especially in deeper networks. RNNs, especially variants like LSTMs, can also be complex but are more focused on retaining and processing temporal information over long sequences.
Advantages of proposed CNNs	• Feature Learning: Automatically learns features from data without manual feature extraction. Spatial Invariance: Recognizes patterns irrespective of their position in the input. Efficient for Image Data: Particularly powerful for tasks like image recognition and classification.
Advantages of proposed CNNs	• Temporal Dynamics: Captures temporal patterns and dependencies in data, ideal for time-series analysis. Variable Length Sequences: Can handle sequences of varying lengths, beneficial for tasks like speech recognition or text analysis. Context Understanding: Due to its memory capability, RNNs can understand context in sequences, which is pivotal for applications like natural language processing.

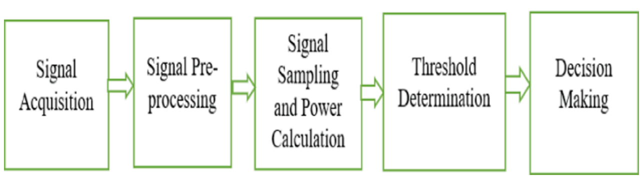


Fig. 1. Energy Detection.

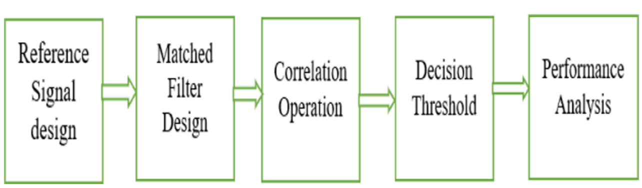


Fig. 2. MF Spectrum Sensing.

- The received signal can be either analog or digital, depending on the implementation.

Signal Pre-processing:

- The received signal is usually passed through a bandpass filter to isolate the frequency band of interest.
- Any unwanted noise or interference outside the band of interest is filtered out.

Signal Sampling and Power Calculation:

- The filtered signal is sampled at a sufficiently high rate to capture the desired frequency components.
- Each sample is squared to obtain the instantaneous power value.

Energy Estimation:

- The squared power samples are averaged over a specific time interval.
- The averaging can be performed using simple moving average or other windowing techniques.

Threshold Determination:

- The threshold is determined based on the desired detection performance and the noise characteristics.
- It is typically set to a level that provides a trade-off between detection probability and false alarm probability.

Decision Making:

- The estimated energy is compared to the threshold.
- If the energy exceeds the threshold, it is classified as the presence of a primary user.
- Otherwise, it is classified as the absence of a primary user.

Energy detection has several advantages in spectrum sensing. It is a simple and computationally efficient technique that can be implemented with relatively low complexity. It is also applicable to various types of signals and modulation schemes since it does not rely on prior knowledge of the primary signal characteristics. Moreover, energy detection can be used in wideband sensing scenarios where the spectrum is not occupied continuously by primary users. However, energy detection also has limitations. It is susceptible to noise uncertainty and variations in the noise power, which can lead to detection errors. Additionally, energy detection may struggle in the presence of weak primary signals or when the noise level is high, as the energy of the primary signal may not significantly exceed the noise floor [39]. To overcome some of these limitations, advanced techniques such as cooperative sensing, cognitive radio networks, or machine learning algorithms can be combined with energy detection to enhance the overall spectrum sensing performance [40].

3.4. Matched filter (MF)

MF spectrum sensing is a technique utilized to sense the presence of a known signal waveform, called the reference or template signal, in a received signal [41]. It relies on correlating the received signal with the reference signal to enhance the detection of the desired signal in the presence of noise and interference [42]. This system is widely used in various communication systems for spectrum sensing, radar applications, and wireless communications [43,44]. To implement MF, the following steps are typically involved, given in Fig. 1:

Reference Signal Design

- The first step in matched filter spectrum sensing is to design the reference signal, which represents the known waveform that is expected to be present in the received signal. The reference signal is typically chosen based on the knowledge of the transmitted signal or by training the system using known signals.

Matched Filter Design

- The matched filter is designed based on the reference signal. It is a filter that maximizes the signal-to-noise ratio (SNR) at its output when the received signal matches the reference signal.
- The matched filter is designed to have an impulse response that is a time-reversed and conjugate of the reference signal.

Correlation Operation:

- The received signal is passed through the matched filter to perform the correlation operation.

- The correlation operation involves convolving the received signal with the time-reversed and conjugated reference signal.

Decision Threshold:

- After the correlation operation, a decision threshold is applied to determine the presence or absence of the desired signal.
- If the correlation output exceeds a certain threshold, it is classified as the presence of the desired signal.

Detection Performance:

- A number of variables, such as the SNR, similarity between the received signal and the reference signal, and noise and interference characteristics, affect the detection performance of matched filter spectrum sensing.
- Statistical signal processing methods can be used to analyse the probabilities of detection and false alarm. When the received signal is tainted with additive white Gaussian noise (AWGN) or when the interference is not associated with the reference signal, matched filter spectrum sensing is effective. In terms of maximising the SNR, it can offer the best detection performance. However, accurate information of the reference signal, which may not always be available, is necessary for matching filter spectrum sensing. Additionally, it is susceptible to signal changes like phase noise and frequency offset, which can impair performance.

3.5. Decision threshold for CNNs and RNNs algorithm

The decision threshold in spectrum sensing represents the value at which a decision is made regarding the presence or absence of a PU signal in a given frequency band. The role of the decision threshold is crucial, as it determines whether the band is considered 'occupied' or 'free,' which in turn affects subsequent actions in cognitive radio systems. When employing CNNs and RNNs in spectrum sensing, the decision threshold acquires a new dimension. Unlike traditional methods, such as energy detection, where the threshold is often set based on the ambient noise level, in deep learning models, this threshold is typically derived from the output of the neural network. For instance, CNNs and RNNs may output a probability score indicating the likelihood of PU presence. The decision threshold can then be set such that if the probability surpasses this value, the band is deemed occupied; otherwise, it's considered free. Adjusting this threshold can manipulate the trade-off between false alarms and missed detections. A lower threshold might result in more false alarms (falsely declaring a channel occupied) but fewer missed detections. Conversely, a higher threshold may reduce false alarms but increase the risk of missing an active PU. With deep learning-based sensing, optimizing this threshold is crucial, given that the algorithms can be highly sensitive to the nuances in the data, and the decision-making process heavily leans on the certainty expressed by these models.

Let us denote the output of the proposed CNNs and RNNS as P_{PU}. This P_{PU} is the probability that the band is occupied by a PU. Let θ be the decision threshold, a value between 0 and 1. This threshold is the cut-off probability. The decision rule based on the threshold can be represented as:

$$D = \begin{cases} 1, if P_{PU} \geq \theta, (BandisOccupied) \\ 0, if P_{PU} < \theta, (Bandisfree) \end{cases}$$

In essence:

If P_{PU} (probability output from the model) is greater than or equal to θ, the algorithm decides that the band is occupied.

If P_{PU} is less than θ, the algorithm decides that the band is free.

The value of θ plays a crucial role in system performance. If θ is set too low, the system might have many false alarms, thinking the band is occupied when it's not. Conversely, if θ is set too high, the system may miss detecting a PU, leading to potential interference.

Fine-tuning θ in the context of CNNs and RNNs for spectrum sensing is crucial. It's often achieved empirically by evaluating system performance on validation datasets and aiming to strike a balance between false alarm rate and detection probability.

4. Simulation results

In this work, we have utilised Matlab 2016 to analyse the performance of different parameters such as Pd, PFA, BER, PAPR, and PSD for several spectrum sensing techniques. We have considered N = 3000 training samples, 512-FFT, 5 MHz of channel bandwidth, and 0.32 rolls of factor. The Pd is a statistical measure that indicates the likelihood of correctly detecting the occurrence of a signal in the availability of noise. It is typically used in scenarios where a system needs to identify or detect specific signals. The SNR is varied over a range of values, and for each SNR value, the pd is calculated or estimated. In Fig. 3, the probability of detection is analysed with respect to SNR. It is seen that DLMs such as CNNs and RNNs obtained a lower SNR of 1.2 dB and 1.6 dB as compared with MF (4 dB) and ED (4.8 dB). Hence, it is concluded that the proposed CNNs and RNNs outperform the conventional method.

In spectrum sensing, PFA vs. PD in Fig. 4 is a graphical representation that illustrates the trade-off between the probability of falsely detecting the occupancy of a signal (false alarm) and the probability of correctly detecting the availability of a signal (detection) in a spectrum sensing system. From the graph, it is clear that conventional algorithms such as ED and MD are not able to identify the noise and represent noise as a detected signal at low SNR. However, the deep learning algorithms have an optical sensing characteristic and take a long sensing time for noisy signals.

In spectrum sensing, the PSD graph is usually used to analyse and characterise the properties of the received signal, especially in CR schemes where secondary users aim to exploit unused frequency bands. The PSD graph in Fig. 5 helps in identifying the presence or absence of primary users' signals and distinguishing them from noise. The spectrum leakage is greatly reduced by using a DLM. The out-of-band radiation is given as −1800 for ED, −2200 for MF, −2600 for RNNs, and −3000 for CNNS, respectively. Hence, it is concluded that the proposed algorithm effectively enhanced the spectral performance of the framework.

The BER is used to evaluate the performance of the framework in terms of error rates at different SNR levels. As revealed in Fig. 6, for each SNR value, the system's efficacy is estimated by measuring the number of bit errors that occur during transmission or reception. At the BER of 10–3, the SNRs of 3.8 dB, 4.7 dB, 6.9 dB, and 8 dB are obtained by the CNNs, RNNs, MF, and ED, respectively. Hence, the DLM gave optimal

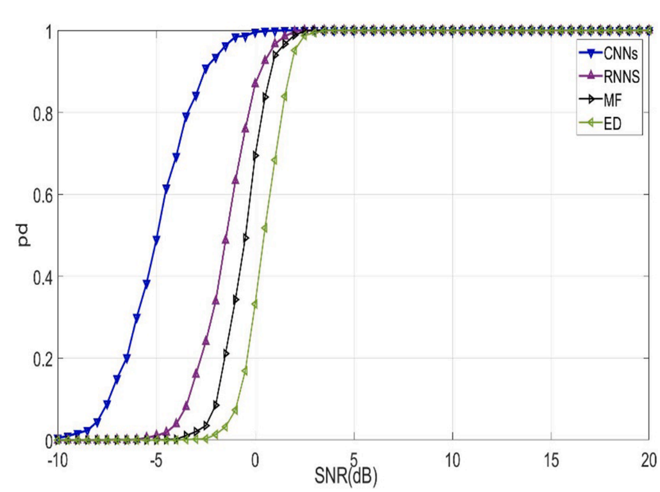


Fig. 3. SNR Vs Pd.

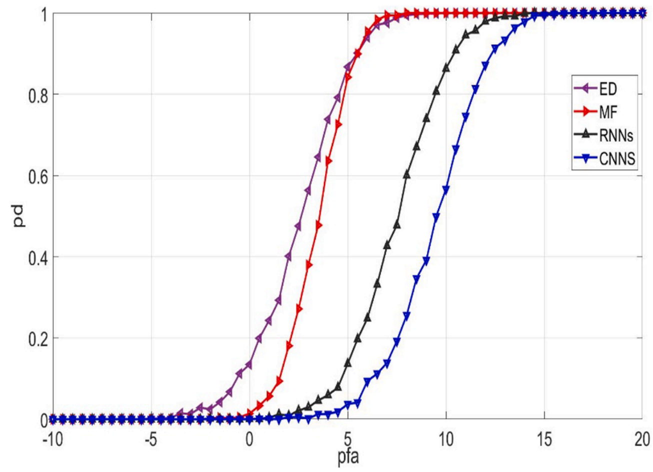


Fig. 4. Pfa vs pd.

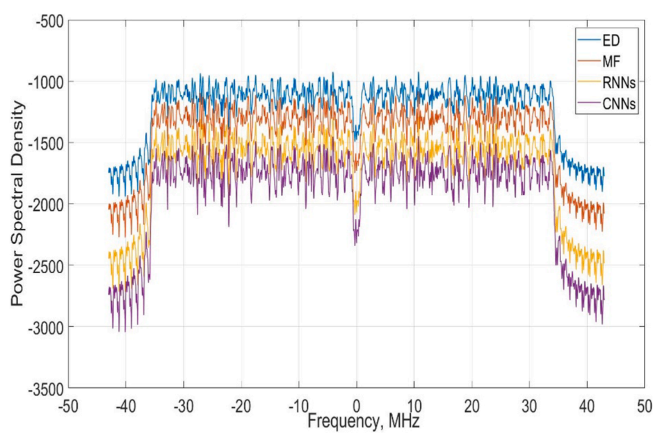


Fig. 5. PSD.

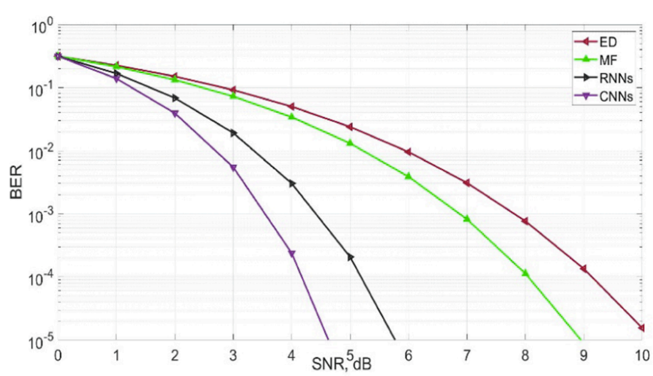


Fig. 6. BER.

performance as compared with conventional algorithms.

The CCDF vs. PAPR graph provides insights into the distribution of the peak power levels in the signal. The CCDF represents the probability that the PAPR exceeds a given threshold, while the PAPR indicates the peak power levels relative to the average power. As shown in Fig. 7, At the CCDF of 10–3, the PAPR values of 6.4 dB, 7.6 dB, 8.3 dB, and 12.1 dB are obtained by the CNNs, RNNs, MF, and ED methods, respectively. Hence, a great deal of power performance can be achieved by utilising deep learning spectrum sensing algorithms.

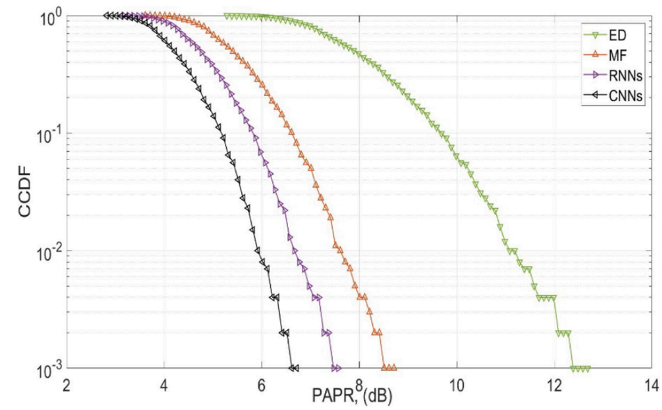


Fig. 7. PAPR analysis.

5. Challenges and limitation of the proposed work

Spectrum sensing is a key component of cognitive radio systems, which are designed to detect available frequency bands (spectral holes or white spaces) to maximize the practice of the radio spectrum. As the wireless communication world has evolved, the necessity for efficient and accurate spectrum sensing techniques has grown. To meet this demand, deep learning algorithms such as CCNNs and RNNs have been investigated for spectrum sensing tasks. Despite the fact that they have several advantages over conventional techniques, applying them might be difficult.

Here are some of the challenges of spectrum sensing using CNNs and RNNs:

- Training Data Requirement: DLM require vast amounts of labeled data to train. Obtaining this data for spectrum sensing tasks can be challenging, especially when considering dynamic real-world environments.
- Computational Complexity: Both CNNs and RNNs have a higher computational requirement than traditional methods. This complexity can make real-time processing and decision-making more challenging, especially in devices with limited computational resources.
- Overfitting: Due to their high modeling capacities, deep learning networks can overfit to training data, leading them to perform poorly on unseen or new data. Regularization techniques are required to mitigate this.
- Model Interpretability: Unlike some outmoded approaches, DLM are often considered “black boxes,” making them hard to interpret. Understanding why a certain spectrum sensing decision was made can be crucial in certain applications.
- Latency Issues: Especially with RNNs, which process data sequentially, there might be latency issues that can affect real-time spectrum sensing tasks.
- Hardware Requirements: Deep learning algorithms, especially when deployed in real-time, require specialized hardware (like GPUs) for efficient operation. This can be a challenge for small devices with limited capabilities.
- Robustness: The wireless environment is filled with uncertainties, noise, and interference. Ensuring that deep learning-based spectrum sensing models are robust in such scenarios is challenging.
- Adaptability: The radio environment is dynamic, with conditions and patterns that might change over time. Deep learning models need to be adaptive to these changes, which might necessitate continuous retraining or fine-tuning.
- Architecture Choices: Deciding on the right architecture, be it CNN, RNN, or hybrid models, can be challenging. Each has its strengths

and weaknesses, and the choice might heavily influence performance.

- Integration with Other Systems: Integrating deep learning models into the larger cognitive radio framework, especially with decision-making and communication modules, can be intricate and require careful engineering.

Utilizing the proposed CNNs and RNNs for spectrum sensing offer several limitations.

- Firstly, the proposed models demand substantial amounts of labeled data for effective training, a challenging feat in dynamic wireless environments.
- The inherent computational complexity is higher as compared with the existing algorithms which can impede real-time processing, especially in resource-constrained devices.
- The “black box” nature of CNNs and RNNs renders their decision-making processes opaque, which contrasts with the more transparent traditional methods.
- The latency of the proposed algorithm is high due to sequential data processing.

6. Conclusion and future work

In this work, we have estimated different parameters such as Pd, Pfa, PSD, and BER to examine the presentation of the CNNs, RNNs, ED, and MF spectrum sensing algorithms. It is seen that the proposed CNNs sense the signal at a low SNR of 1.2 dB and are also able to distinguish between noise and the original signal. Hence, the sensing time of the PFA DLM algorithm is high as compared with conventional methods. Further, it is seen that the out-of-band radiation in CNNs is quite low, approximately −3000, which enhances the spectral utilization efficiency of the framework. DLM offer a highly promising avenue for advancing spectrum sensing techniques in wireless communication systems. Their capability to inevitably study and excerpt eloquent attributes from raw spectrum data, coupled with their adaptability and robustness, make them well-suited for addressing the dynamics of modern spectrum environments. One of the key advantages of deep learning algorithms is their ability to handle complex and non-linear relationships within the spectrum data. Traditional spectrum sensing techniques often rely on handcrafted features and assumptions about the statistical characterisctics of the signals, which can lower their efficiency in dynamic and heterogeneous environments. Deep learning algorithms, on the other hand, have the capacity to adapt and generalize well to different signal types and environmental conditions. This flexibility leads to improved detection accuracy, even in scenarios with challenging and diverse spectrum conditions.

The integration of DLM like CNNs and RNNs into spectrum sensing showcases promising results, but the horizon for future exploration in this domain remains vast. One immediate area of development is the fusion of these networks, possibly through hybrid models, to jointly exploit spatial and temporal features in wireless signals. Such a composite approach could achieve higher detection accuracies, even in challenging low signal-to-noise ratio conditions. Another significant advancement will be in the realm of real-time implementations. While these networks show efficacy in laboratory settings, their real-world deployment in dynamic and unpredictable wireless environments will necessitate optimized architectures and reduced computational complexities. Edge computing paradigms, where the processing happens close to the data source, may become invaluable in this context. Adapting DLM to operate efficiently on edge devices without compromising performance will be crucial. Transfer learning is another prospective avenue. Training DLM can be computationally intensive and requires substantial data. But what if a model trained in one environment could be adapted to another with minimal retraining. Exploring how transferable the learned features are across different wireless scenarios and regions could fast-track the deployment of these models. There’s also the challenge of adversarial attacks in wireless communication. As DLM become pervasive in spectrum sensing, they might become targets for malicious entities aiming to exploit their vulnerabilities. Research into developing robust models that can withstand such attacks, possibly by integrating adversarial training, will be of paramount importance. Lastly, the fusion of old-style signal processing procedures with deep learning might offer the finest of both worlds. Harnessing the strengths of classical methods, while filling their gaps with the adaptability and intelligence of deep learning, could lead to even more resilient and efficient spectrum sensing systems. The evolution of spectrum sensing with CNNs and RNNs is poised to witness exciting breakthroughs, setting the stage for revolutionary advancements in cognitive radio networks.

Funding

This work was supported by (i) Suranaree University of Technology (SUT), (ii) Thailand Science Research and Innovation (TSRI) and (iii) National Science, Research and Innovation Fund (NSRF) (NRIIS number 179284).

Code availability (software application or custom code)

NA.

Authors’ contributions (optional)

A.K wrote the paper and perform the experiments, N.G and S.C edited the paper, M.H.A, P.U and M.U perform the analysis of the work.

Availability of data and material (data transparency)

This article does not contain any studies with human participants or animals performed by any of the authors.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Kumar A, Sharma MK, Sengar K, Kumar S. NOMA based CR for QAM-64 and QAM-256. Egyp Informat J 2020;21(2):67–71.

[2] Cai P, Zhang Y. Intelligent cognitive spectrum collaboration: Convergence of spectrum sensing, spectrum access, and coding technology. Intelligent Converged Networks June 2020;1(1):79–98. https://doi.org/10.23919/ICN.2020.0006.

[3] Guimarães DA. Pietra-Ricci Index Detector for Centralized Data Fusion Cooperative Spectrum Sensing. IEEE Trans Veh Technol Oct. 2020;69(10):12354–8. https://doi.org/10.1109/TVT.2020.3009440.

[4] Kumar A, Venkatesh J, Gaur N, Alsharif MH, Jahid A, Raju K. Analysis of Hybrid Spectrum Sensing for 5G and 6G Waveforms. Electronics 2023;12(1):138. https://doi.org/10.3390/electronics12010138.

[5] Xiong T, Yao Y-D, Ren Y, Li Z. Multiband Spectrum Sensing in Cognitive Radio Networks With Secondary User Hardware Limitation: Random and Adaptive Spectrum Sensing Strategies. IEEE Trans Wirel Commun May 2018;17(5):3018–29. https://doi.org/10.1109/TWC.2018.2805729.

[6] Arun Kumar J, Venkatesh NG, Alsharif MH. Peerapong Uthansakul, Monthippa Uthansakul. Cyclostationary and energy detection spectrum sensing beyond 5G waveforms. Electron Res Arch 2023;31(6):3400–16. https://doi.org/10.3934/era.2023172.

[7] Lee D-J, Jang M-S. Optimal spectrum sensing time considering spectrum handoff due to false alarm in cognitive radio networks. IEEE Commun Lett December 2009;13(12):899–901. https://doi.org/10.1109/LCOMM.2009.12.091448.

[8] Shen F, Wang Z, Ding G, Li K, Wu Q. 3D compressed spectrum mapping with sampling locations optimization in spectrum-heterogeneous environment. IEEE Trans Wirel Commun Jan. 2022;21(1):326–38. https://doi.org/10.1109/TWC.2021.3095342.

[9] Kumar A, Sharma H. Intelligent cognitive radio spectrum sensing based on energy detection for advanced waveforms. Radioelectron Commun Syst 2022;65:149–54. https://doi.org/10.3103/S0735272722030049.

[10] Wu H, Yu B, Yan K, Zhang X, Wu Y. Spectrum sensing based on novel blind pilot detection algorithm. In: 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Valencia, Spain, 2018, pp. 1-6, doi: 10.1109/BMSB.2018.8436750.

[11] Li F, Xu Z. Sparse Bayesian hierarchical prior modeling based cooperative spectrum sensing in wideband cognitive radio networks. IEEE Signal Process Lett May 2014;21(5):586–90. https://doi.org/10.1109/LSP.2014.2311902.

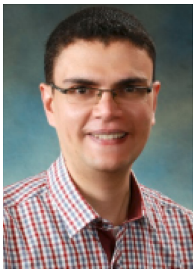
[12] Islam H, Das S, Bose T, Ali T. Diode based reconfigurable microwave filters for cognitive radio applications: a review. IEEE Access 2020;8:185429–44. https://doi.org/10.1109/ACCESS.2020.3030020.

[13] Chae K, Kim Y. DS2MA: A deep learning-based spectrum sensing scheme for a multi-antenna receiver. IEEE Wireless Commun Lett 2023;12(6):952–6. https://doi.org/10.1109/LWC.2023.3250257.
[14] Xie J, Fang J, Liu C, Li X. Deep learning-based spectrum sensing in cognitive radio: a CNN-LSTM approach. IEEE Commun Lett Oct. 2020;24(10):2196–200. https://doi.org/10.1109/LCOMM.2020.3002073.
[15] Liu S, Liu L, Yi Y. Quantized reservoir computing for spectrum sensing with knowledge distillation. IEEE Trans Cogn Devel Syst March 2023;15(1):88–99. https://doi.org/10.1109/TCDS.2022.3147789.
[16] Hamedani K, Liu L, Yi Y. Energy efficient MIMO-OFDM spectrum sensing using deep stacked spiking delayed feedback reservoir computing. IEEE Trans Green Commun Networking March 2021;5(1):484–96. https://doi.org/10.1109/TGCN.2020.3046725.
[17] Bokobza Y, Dabora R, Cohen K. Deep reinforcement learning for simultaneous sensing and channel access in cognitive networks. IEEE Trans Wireless Commun 2023;22(7):4930–46. https://doi.org/10.1109/TWC.2022.3230872.
[18] Park H, Lee J-W, Min D-J. Improvement of spectrum suppression-based deep learning interpolation technique. IEEE Trans Geosci Remote Sensing 2023;61: 1–19. https://doi.org/10.1109/TGRS.2023.3282619.
[19] Xie J, Liu C, Liang Y-C, Fang J. Activity pattern aware spectrum sensing: a CNN-based deep learning approach. IEEE Commun Lett June 2019;23(6):1025–8. https://doi.org/10.1109/LCOMM.2019.2910176.
[20] Gao J, Yi X, Zhong C, Chen X, Zhang Z. Deep learning for spectrum sensing. IEEE Wireless Commun Lett Dec. 2019;8(6):1727–30. https://doi.org/10.1109/LWC.2019.2939314.
[21] Chew D., Cooper AB. Spectrum sensing in interference and noise using deep learning. In: 2020 54th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 2020, pp. 1-6, doi: 10.1109/CISS48834.2020.1570617443.
[22] Shachi P, Sudhindra KR, Suma MN. Deep learning for cooperative spectrum sensing. In: 2020 2nd PhD Colloquium on Ethically Driven Innovation and Technology for Society (PhD EDITS), Bangalore, India, 2020, pp. 1-2, doi: 10.1109/PhDEDITS51180.2020.9315306.
[23] Mohanakurup V, Baghela VS, Kumar S, Srivastava PK, Doohan NV, Soni M, et al. 5G Cognitive Radio Networks Using Reliable Hybrid Deep Learning Based on Spectrum Sensing. Wirel Commun Mob Comput 2022;2022:1–17. https://doi.org/10.1155/2022/1830497.
[24] Jung T-Y, Jeong E-R. Spectrum Sensing Based On Deep Learning To Increase Spectrum Utilization. Turkish J Comput Mathe Educat 2021;12(6):538–43.
[25] Giri MK, Majumder S. Cooperative spectrum sensing using extreme learning machines for cognitive radio networks. IETE Tech Rev 2022;39(3):698–712.
[26] Uppala Appala Raju, Narasimhulu Venkata, Prasad Satya. Improved convolutional neural network based cooperative spectrum sensing for cognitive radio. KSII Trans Internet Informat Syst 2021;15(6):2128–47.
[27] Gul N, Ahmed S, Kim SM, Khan MS, Kim J. Reconfigurable sensing time in cooperative cognitive network using machine learning. Comput Mater Continua 2023;74(3):5209–27.
[28] Jiang F, Chang D-W, Ma S, Hu Y-J, Xu Y-H. A Residual Learning-Aided Convolutional Autoencoder for SCMA. IEEE Commun Lett May 2023;27(5): 1337–41. https://doi.org/10.1109/LCOMM.2023.3260881.
[29] Liu C, Liu X, Liang Y. -C. Deep CNN for Spectrum Sensing in Cognitive Radio. ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761360.
[30] Tan Y, Jing X. Cooperative Spectrum Sensing Based on Convolutional Neural Networks. Appl Sci 2021;11(10):4440. https://doi.org/10.3390/app11104440.
[31] Dewangan N, Kumar A, Patel RN. Performance analysis of LSTM-CNN for spectrum sensing in cognitive radio networks. Mathe Statistician Eng Appl 2022;71(4): 6218–29.
[32] Alqahtani AS, Changalasetty SB, Parthasarathy P, Thota LS, Mubarakali A. Effective spectrum sensing using cognitive radios in 5G and wireless body area networks. Comput Electr Eng 2023;105:108493. https://doi.org/10.1016/j.compeleceng.2022.108493.
[33] Cai S, Lau VKN. RNN-based learning of nonlinear dynamic system using wireless IIoT networks. IEEE Internet Things J 2021;8(14):11177–92. https://doi.org/10.1109/JIOT.2021.3052925.
[34] Ye Z, Gilman A, Peng Q, Levick K, Cosman P, Milstein L, et al. IEEE globecom workshops (GC Wkshps). Waikoloa, HI, USA 2019;2019:1–6. https://doi.org/10.1109/GCWkshps45667.2019.9024482.
[35] Xu M, Yin Z, Zhao Y, Wu Z. Cooperative Spectrum Sensing Based on Multi-Features Combination Network in Cognitive Radio Network. Entropy 2022;24(1):129. https://doi.org/10.3390/e24010129.
[36] Li G, Sun C, Zhang J, Jorswieck E, Xiao B, Hu A. Physical Layer Key Generation in 5G and Beyond Wireless Communications: Challenges and Opportunities. Entropy 2019;21:497.
[37] Kumar A, Gupta Manisha. A review on activities of fifth generation mobile communication system. Alex Eng J 2018;57(2):1125–35.
[38] kockaya K, Develi I. Spectrum sensing in cognitive radio networks: threshold optimization and analysis. J Wireless Com Network 2020;25(1). https://doi.org/10.1186/s13638-020-01870-7.
[39] Pandya P, Durvesh A, Parekh N. Energy Detection Based Spectrum Sensing for Cognitive Radio Network. In: 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 2015, pp. 201-206, doi: 10.1109/CSNT.2015.264.
[40] Wu W, Wang Zi, Yuan Lu, Zhou F, Lang F, Wang B, et al. IRS-enhanced energy detection for spectrum sensing in cognitive radio networks. IEEE Wireless Commun Lett 2021;10(10):2254–8. https://doi.org/10.1109/LWC.2021.3099121.
[41] Kumar A, NandhaKumar P. OFDM system with cyclostationary feature detection spectrum sensing. ICT Express 2019;5(1):21–5.
[42] Dhananjaya S, Yuvaraju BN. A novel method in matched filter spectrum sensing to minimize interference from compromised secondary users of cognitive radio networks. In: 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Msyuru, India, 2018, pp. 228-231, doi: 10.1109/ICEECCOT43722.2018.9001622.
[43] Zhang L, Xing M-d, Qiu C-W, Bao Z. Two-dimensional spectrum matched filter banks for high-speed spinning-target three-dimensional ISAR imaging. IEEE Geosci Remote Sensing Lett 2009;6(3):368–72. https://doi.org/10.1109/LGRS.2009.2013487.
[44] Eduardo AF, González Caballero RG. Experimental evaluation of performance for spectrum sensing: Matched filter vs energy detector. In: IEEE Colombian Conference on Communication and Computing (IEEE COLCOM 2015), Popayan, Colombia, 2015, pp. 1-6, doi: 10.1109/ColComCon.2015.7152105.

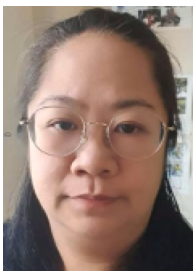
Arun Kumar received his Ph.D. in electronics and communication engineering from JECRC University, Jaipur, India. He is an Associate Professor in Electronics and Communication Engineering at New Horizon College of Engineering in Bengaluru, India. Dr. Kumar has a total of 10 years of teaching experience and has published more than 82 research articles in SCI-E and Scopus Index journals, 2 books and 3 International patents (granted). His research interests are advanced waveforms for 5G mobile communication systems and 5G-based smart hospitals, PAPR reduction techniques in the multicarrier waveform, and spectrum sensing techniques. Dr. Kumar has successfully implemented different reduction techniques for multi-carrier waveforms such as NOMA, FBMC, UFMC, and so on, and has also implemented and compared different waveform techniques for the 5G system. Currently, he is working on the requirements of a 5G-based smart hospital system. He is a member of the IEEE and a reviewer for many refereed, indexed journals. He was the Lead Guest Editor of the Special Issue Journal titled "Advanced 5G Communication System for Transforming Healthcare" in the Journal CMC-Computers, Materials and Continua (Q2, SCI, Scopus, Impact Factor: 3.7). He has served as Co-Convenor, Technical Programme Committee (TPC) Member and Reviewer in various International Conferences such as ICPCCAI-2019, ICPCCAI-2020, IEEE ISMAC (2018), ESG2018 and so on.

Nishant Gaur working as an Associate Professor of Physics at JECRC University. He has over 23 years of experience in teaching and research. Nishant is currently based in Jaipur, India. He has published several journals and conference papers. his research interest area is 5G, Spectrum Sensing, Optical communication and so on.

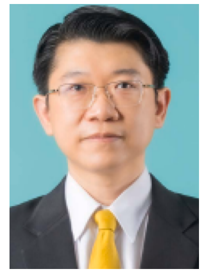
Sumit Chakravarty currently works as an Associate Professor of Electrical Engineering with Kennesaw State University. He also serves as the MSECE program coordinator. He completed his doctoral studies from University of Maryland, Baltimore County and his Master of Science from Texas A&M University, Kingsville, both in Electrical Engineering. His PhD dissertation is on analysis of Hyperspectral Signatures and Data. He has multiple peer-reviewed journal publications, conference publications, a book chapter and three granted patents. His area of expertise is Wireless Communications, AI and Signal Processing.



Mohammed H. Alsharif received the B.Eng. degree in electrical engineering (wireless communication and networking) from the Islamic University of Gaza, Palestine, in 2008, and the M.Sc.Eng. and Ph.D. degrees in electrical engineering (wireless communication and networking) from the National University of Malaysia, Malaysia, in 2012 and 2015, respectively. He joined Sejong University, South Korea, in 2016, where he is currently an Assistant Professor with the Department of Electrical Engineering. His current research interests include wireless communications and networks, including wireless communications, network information theory, the Internet of Things (IoT), green communication, energy-efficient wireless transmission techniques, wireless power transfer, and wireless energy harvesting.



Monthippa Uthansakul (Member, IEEE) received the B.E. degree in telecommunication engineering from the Suranaree University of Technology, Thailand, in 1997, and the M.E. degree in electrical engineering from Chulalongkorn University, Thailand, in 1999, and the Ph.D. degree in information technology and electrical engineering from The University of Queensland, Australia, in 2007. She is currently an Associate Professor with the Telecommunication School, Suranaree University of Technology. Her research interests include wideband/narrowband smart antennas, automatic switch beam antenna, DOA finder, microwave components, application of smart antenna, and advance wireless communications. She received the Second Prize of Young Scientist Award from 16th International Conference on Microwaves, Radar, and Wireless Communications, Poland, in 2006.



Peerapong Uthansakul (Member, IEEE) received the B.E. and M.E. degrees in electrical engineering from Chulalongkorn University, Bangkok, Thailand, in 1996 and 1998, respectively, and the Ph.D. degree in information technology and electrical engineering from The University of Queensland, Brisbane, QLD, Australia, in 2007. From 1998 to 2001, he was employed as a telecommunication engineer at one of the leading telecommunication companies in Thailand. He is currently working as an Associate Professor and the Dean of the Research Department, Suranaree University of Technology, Nakhon Ratchasima, Thailand. He has more than 100 research publications and the author/coauthor of various books related to MIMO technologies. His research interests include green communications, wave propagation modeling, MIMO, massive MIMO, brain wave engineering, OFDM and advanced wireless communications, wireless sensor networks, embedded systems, the Internet of Things, and network security. He has won various national awards from the Government of Thailand for his contributions and motivation in the field of science and technology. Furthermore, he is the Editor of Suranaree Journal of Science and Technology and other leading Thailand journals related to science and technology.