IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 August 2022, accepted 12 September 2022, date of publication 15 September 2022,
date of current version 23 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206814

RESEARCH ARTICLE

PU-DetNet: Deep Unfolding Aided Smart Sensing Framework for Cognitive Radio

BRIJESH SONI[1], DHAVAL K. PATEL[2], (Member, IEEE), SANKET B. SHAH[3], MIGUEL LÓPEZ-BENÍTEZ[4,5], (Senior Member, IEEE), AND SIDDHARTAN GOVINDASAMY[1], (Member, IEEE)

[1]Department of Engineering, Boston College, Boston, MA 02467, USA
[2]School of Engineering and Applied Science, Ahmedabad University, Ahmedabad 380009, India
[3]Department of Computer Science, San Francisco State University, San Francisco, CA 94132, USA
[4]Department of Electrical Engineering and Electronics, University of Liverpool, L69 3BX Liverpool, U.K.
[5]ARIES Research Centre, Antonio de Nebrija University, 8015 Madrid, Spain

Corresponding author: Brijesh Soni (sonib@bc.edu)

The work of Dhaval K. Patel was supported in part by the Department of Science and Technology-Gujarat Council of Science and Technology (DST-GUJCOST) under Grant GUJCOST/STI/2021-22/3916.

ABSTRACT Spectrum sensing in cognitive radio (CR) paradigm can be broadly categorized as analytical-based and data-driven approaches. The former is sensitive to model inaccuracies in evolving network environment, while the latter (machine learning (ML)/deep learning (DL) based approach) suffers from high computational cost. For devices with low computational abilities, such approaches could be rendered less useful. In this context, we propose a deep unfolding architecture namely the Primary User-Detection Network (PU-DetNet) that harvests the strength of both: analytical and data-driven approaches. In particular, a technique is described that reduces computation in terms of inference time and the number of floating point operations (FLOPs). It involves binding the loss function such that each layer of the proposed architecture possesses its own loss function whose aggregate is optimized during training. Compared to the state-of-the-art, experimental results demonstrate that at SNR $= -10$ dB, the probability of detection is significantly improved as compared to the long short term memory (LSTM) scheme (between 39% and 56%), convolutional neural network (CNN) scheme (between 45% and 84%), and artificial neural network (ANN) scheme (between 53% and 128%) over empirical, 5G new radio, DeepSig, satellite communications, and radar datasets. The accuracy of proposed scheme also outperforms other existing schemes in terms of the F1-score. Additionally, inference time reduces by 91.69%, 90.90%, and 93.15%, while FLOPs reduces by 62.50%, 56.25%, 64.70% w.r.t. LSTM, CNN and ANN schemes, respectively. Moreover, the proposed scheme also shows improvement in throughput by 56.39%, 51.23%, and 69.52% as compared to LSTM, CNN and ANN schemes respectively, at SNR $= -6$ dB.

INDEX TERMS Dynamic spectrum access, machine learning, deep learning, computational cost.

I. INTRODUCTION

Detection and sensing is envisaged to have numerous applications for fifth generation (5G) and beyond wireless communications, including but not limited to radar detection, integrated sensing and communications, and dynamic spectrum access [1]. With the rapid advancement of wireless technologies and services, the number of connected devices have increased massively in the fixed/statically allocated radio spectrum [2].

Dynamic spectrum access/cognitive radio (CR) communications have emerged as a potential solution to trade-off between spectrum availability and its demanding growth [3]. CR is defined as "*an intelligent wireless communication system that is aware of its environment and uses the methodology of understanding-by-building to learn from the environment*" [4]. The underlying principle of CR is to allow the unlicensed users to access the temporarily unoccupied

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

licensed bands in an opportunistic and non-interfering manner. This calls for highly reliable and efficient spectrum sensing schemes [4].

A. STATE-OF-THE-ART AND MOTIVATION

State-of-the-art spectrum sensing approaches can be broadly classified into two categories: analytical-based and data-driven. Analytical-based approaches involves the use of established mathematical techniques and analytical tools for evaluating the spectrum sensing performance. Many such works can be found in [5], [6], [7], [8], and [9], and references therein. Although analytical approaches help to analyze the system well, they require precise modeling of a large number of environmental characteristics. It becomes challenging in practical scenarios when the wireless environment is rapidly evolving, especially in the 5G and beyond networks [10]. Contrary to analytical-based approaches, data-driven approaches do not incorporate any knowledge obtained by analysis through mathematical techniques, and rely mostly on the distribution and/or type of data.

With the rapid advancement in learning-based signal processing techniques, data-driven approaches have gained wide attention from industry and academia in the context of future wireless networks [11], [12], [13], [14], including spectrum sensing in CR networks [15]. Owing to the excellent learning ability of data-driven approaches, many works in literature have leveraged the machine learning (ML) / deep learning (DL) techniques considering spectrum sensing as a binary classification problem. For instance, artificial neural network (ANN) based spectrum sensing was carried out in [16]. A novel ANN-based hybrid sensing scheme which used energy values and the Zhang statistics as the training features was proposed in [17], while its improved technique was presented in [18]. The sensing of orthogonal frequency division multiplexing (OFDM) signals at a low signal-to-noise ratio (SNR) regime using a Naive-Bayes classifier was proposed in [19]. Recently, ML enabled cooperative spectrum sensing for non-orthogonal multiple access was carried out in [20]. To overcome the efforts required in feature engineering for ML approaches, few works have also applied the DL approach to spectrum sensing. For instance, a convolutional neural network (CNN) based spectrum sensing was proposed in [21], [22], and [23], CNN based cooperative sensing was implemented in [24], while stacked auto-encoder based spectrum sensing of OFDM signals was proposed in [25]. Moreover, long short term memory (LSTM) based spectrum sensing schemes were proposed in [26] and [27]. Few other DL architectures for spectrum sensing schemes were reported in [28], [29], [30], and [31].

Although the data-driven approach aided ML/DL spectrum sensing techniques have shown performance efficiency and unprecedented empirical success, such techniques usually suffer from the requirement of exhaustive training, computationally large training data, explainability of trained ML/DL model, numerous trainable parameters, and high computational cost. Furthermore, such ML/DL aided frameworks are generally trained, tested and deployed on an environment powered by a computationally equipped graphics processing unit (GPU). The time taken to produce outputs completely depends upon the specifications and robustness of the hardware. This implies that devices with lesser efficiency and capacity are bound to face difficulties in deploying ML/DL frameworks effectively. Especially for network architectures where it is envisaged that there would be massive devices with low computational abilities, for instance machine-type communications, low cost sensors, and edge devices (also included in the 3rd generation partnership project (3GPP) release-14 and to its subsequent versions [32]), the implementation and usage of a conventional ML/DL framework could be rendered less useful [33].

To overcome the aforementioned shortcomings of both: the analytical and data-driven approaches, the concept of deep unfolding was introduced in [34] which aims at simultaneously harvesting the strength of both the approaches. Given an analytical-based approach, one unfolds the iterations of a derived algorithm into a layer-wise structure analogous to an ML/DL architecture such that each iteration is considered a layer and an algorithm is called a network [35]. This approach combines the expressive power of a conventional deep network with the explainability of an analytical-based approach [36], [37]. There have been few recent works in the literature which have utilized deep unfolding techniques. For instance, the work in [38] proposed to compute the bit error rate for MIMO systems using alternating direction method of multipliers (ADMM) unfolding, while [33] attempted via iterative algorithm induced deep unfolding. Moreover, MIMO detection using deep unfolding technique was studied in [39]. The work in [40] focused on the deep unfolding for compressive sensing. A comprehensive survey of deep unfolding for wireless communications can be found in [41].

As far as spectrum sensing is concerned, the detection performance needs to be accurate as well as computationally efficient. Inspired by the fact that the future wireless networks is envisaged to have massive machine-type communications, as included in 3GPP release-14 and its subsequent versions; the aforementioned shortcomings of the data-driven approach would further add challenges in sensing for devices with low computational abilities. In this context, the key objective of this work is to design a deep unfolding aided smart sensing framework, which to the best of the authors' knowledge is yet to be reported in the literature. A deep unfolding scheme alongside the architectural innovations presented performs spectrum sensing in a computationally inexpensive manner while also yielding promising results, especially in the low SNR regime.

B. CONTRIBUTIONS OF THIS WORK

The key contributions of this work can be summarized as:

- Firstly, we propose a novel architecture namely the primary user detection network (PU-DetNet), where the approach is to iteratively reduce the error between estimation (classification) and ground truth by unfolding

the iterations of the algorithm. When such k iterations are unfolded, a DL architecture with k layers analogous to conventional deep network is formed along with the explainability of an analytical-based approach.

- Secondly, a technique is described that involves binding the loss function such that each layer of the proposed architecture possesses its own loss function whose aggregate is optimized. This technique leads to a state where after training, values of the loss functions from shallow to deep layers become nearly equal. The implication of this property is that the shallow layers exhibit optimal performance than the deep layers, hence forming the computationally efficient model as compared to data-driven approaches.
- Thirdly, the proposed PU-DetNet scheme is experimentally validated with spectrum data from five different datasets corresponding to realistic scenarios which includes: an empirical test-bed measurement setup, a 5G new radio (NR) dataset, the DeepSig dataset, a satellite communications dataset, and a radar dataset. The obtained results show that at SNR $= -10$ dB, the probability of detection is improved by a significant amount compared to the LSTM approach (between 39% to 56%), CNN approach (45% to 84%) and the ANN approach (between 53% and 128%). Also, the accuracy of proposed scheme outperforms other existing schemes in terms of the F1-score. Moreover, compared to the baselines: LSTM, CNN, and ANN based sensing schemes, inference time reduces by 91.69%, 90.90%, and 93.15%, while the number of floating point operations (FLOPs) reduces by 62.50%, 56.25%, 64.70% respectively. Furthermore, the proposed scheme shows an improvement in throughput by 56.39%, 51.23%, and 69.52% as compared to LSTM, CNN and ANN scheme respectively, at SNR = -6 dB.

The remainder of this paper is organized as follows. Section II describes the system model and preliminaries of the deep unfolding framework. The architecture of the proposed PU-DetNet scheme is presented in Section III. Section IV provides the description of the considered datasets. Section V comprehensively describes the experimental results. Finally, Section VI concludes this work.

Notations: Boldface uppercase letters represent matrices. Boldface lowercase represent vectors. Raw outputs (pre-activation) are denoted as $(\cdot)'$. Transpose is denoted as $(\cdot)^T$. The i^{th} data sample is denoted as $(\cdot)_i$ and the truth label corresponding to the same is denoted as $(\cdot)_i^t$. Variables corresponding to the k^{th} layer of the considered architecture are referred to as $(\cdot)_k$.

II. SYSTEM MODEL AND PRELIMINARIES OF DEEP UNFOLDING

A. CONSIDERED SYSTEM MODEL

The problem of PU detection can be formulated as a binary classification problem. The secondary user (SU) detects the presence of the PU from the signal it receives, defined as $y(t)$.

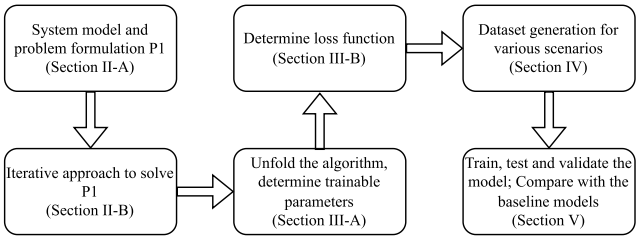


FIGURE 1. Generic deep unfolding framework and the paper flow.

TABLE 1. Variables and their description used in this paper.

Variable	Description
$\mathbf{X}$	Signal energies dataset
$\mathbf{y}_k$	Estimated outputs of the k^{th} layer
$\mathbf{y}^t$	Ground truth in vector form
$\mathbf{e}_k$	Error value between $\mathbf{y}_k$ and $\mathbf{y}^t$
$\mathbf{x}_i$	Signal energies of the i^{th} data sample
θ_k	Trainable parameters of k^{th} layer
n	Number of samples in the dataset
y_i^t	True label/ground truth of the i^{th} data sample
δ_k	Tunable scaling parameter of k^{th} layer

The binary hypothesis can be written as:

$$\begin{aligned} \mathscr{H}_0 &: y(t) = w(t) \\ \mathscr{H}_1 &: y(t) = h(t)x(t) + w(t), \end{aligned} \tag{1}$$

where $y(t)$ is the received signal by the SU at a time t, $x(t)$ is the transmitted signal, $w(t)$ is additive white Gaussian noise with zero mean and variance σ_w^2, and $h(t)$ is the flat-fading channel coefficient. $\mathscr{H}_0$, the null hypothesis, describes the scenario when only noise is present during the fixed time sensing event, i.e., indicating the absence of PU, while $\mathscr{H}_1$, the alternate hypothesis, describes the presence of PU.

Fig. 1 shows the generic deep unfolding framework and the paper flow. Given an analytical-based approach, one unfolds the iterations of a derived algorithm into a layer-wise structure analogous to an ML/DL architecture such that each iteration of an algorithm is computed by a different layer in the unfolded architecture [35]. To form such a model, an iterative algorithm is derived and trainable parameters are identified which form the basis of the learning architecture. Instead of optimizing a generic neural network, we untie the trainable parameters of the model across layers to create a more flexible network. The resulting architecture can be trained discriminatively to obtain accurate inference within a fixed network size. This approach combines the expressive power of a conventional DL method with the explainability of an analytical-based approach.

B. ITERATIVE APPROACH TO PU DETECTION

The variables used in this paper are listed in Table 1. The objective is to derive an expression that aims at iteratively

reducing the error of estimation.[1] The error is estimated and reduced in an iterative manner. Mathematically, it can be expressed as:

$$\mathbf{y}_k = \Pi\left[\mathbf{y}_{k-1} - \delta_k \mathbf{e}_k\right], \tag{2}$$

where $\mathbf{y}_k$ is the estimation in the k^{th} iteration, $\mathbf{y}_{k-1}$ is the estimation in the $(k-1)^{th}$ iteration with respect to the whole signal dataset, δ_k is a tunable scaling parameter, $\mathbf{e}_k$ is the error observed in the k^{th} iteration, and $\Pi[\cdot]$ is a non linear projection operator. To define the error $\mathbf{e}_k$ in (2), we first define the error for a single data sample:

$$e_{ik} = y_i^t - \boldsymbol{\theta}_k^T \mathbf{x}_i, \tag{3}$$

where e_{ik} is the error of k^{th} iteration on i^{th} data sample, y_i^t is the i^{th} true label, $\mathbf{x}_i$ is i^{th} the data sample and $\boldsymbol{\theta}_k^T$ represents the trainable parameters during the k^{th} iteration. For the entire dataset, the vectorized form can be expressed as:

$$\mathbf{e}_k' = \mathbf{y}^t - \boldsymbol{\theta}_k^T \mathbf{X}, \tag{4}$$

where $\mathbf{e}_k'$, $\mathbf{y}^t$ are the vector forms of their respective variables and $\mathbf{X}$ represents the entire dataset. However, the estimated error $\mathbf{e}_k'$ is imperfect as its value depends upon the size of the dataset. Hence, the error is further normalized as:

$$\mathbf{e}_k = \left|\left| \frac{\mathbf{y}^t - \boldsymbol{\theta}_k^T \mathbf{X}}{n} \right|\right|, \tag{5}$$

where n is the number of total samples in the dataset. On substituting (5) in (2), we obtain:

$$\mathbf{y}_k = \Pi\left[\mathbf{y}_{k-1} - \delta_k \left|\left| \frac{\mathbf{y}^t - \boldsymbol{\theta}_k^T \mathbf{X}}{n} \right|\right| \right]. \tag{6}$$

The above expression represents an iterative algorithm to estimate $\mathbf{y}_k$. In the deep unfolding approach we unfold (6) into layers of a neural network and describe the resulting architecture in the next section.

III. PROPOSED PU-DetNet ARCHITECTURE

A. PROPOSED ARCHITECTURE FOR PU-DETNET SCHEME

The proposed architecture contains various components which are listed in Table 2. From (6) it is clear that the architecture follows a relationship such that a layer receives the previous layer's output, reduce the error and make it closer to the ground truth. As mentioned, we utilize (6), implement and unfold it to formulate the design of a single layer of the proposed PU-DetNet. In case of a generic neural network, the output of any given layer is dependent on just the previous layer's output (or just the input data, in case of the first hidden layer). However, from (6) $\mathbf{y}_k$ is dependant on $\mathbf{y}_{k-1}$ and $\mathbf{x}$. Thus, the output of k^{th} hidden layer can be given as:

[1]Although PU detection is a classification problem, it is analogous to estimating a function $f : \mathbb{R}^n \rightarrow \{\mathscr{H}_0, \mathscr{H}_1\}$ and assigning a categorical label to the input data [42].

TABLE 2. Components of the architecture used in this paper.

Component	Description
$\mathbf{W}_{1k}, \mathbf{b}_{1k}$, $\mathbf{W}_{2k}, \mathbf{b}_{2k}$, $\mathbf{W}_{3k}, \mathbf{b}_{3k}$	Trainable weights and biases of the k^{th} layer
$\mathbf{z}_k$	Output of the k^{th} hidden layer
$\mathbf{v}_k$	Propagation vector output of the k^{th} layer
y_k	Output of the k^{th} layer
α	Output residual constant
h_s	Size of hidden layer
x_s	Number of columns in dataset / size of a single example
n_l	Number of hidden layers
y_s	Size of the output
v_s	Size of the propagation vector
ReLU	Rectified linear unit function

$$\mathbf{z}_k = \text{ReLU}\left(\mathbf{W}_{1k} \begin{bmatrix} \mathbf{x} \\ \mathbf{v}_{k-1} \\ \mathbf{y}_{k-1} \end{bmatrix} + \mathbf{b}_{1k} \right) \tag{7}$$

$$\mathbf{v} = \mathbf{W}_{2k}\mathbf{z}_k + \mathbf{b}_{2k} \tag{8}$$

$$\mathbf{y} = \mathbf{W}_{3k}\mathbf{z}_k + \mathbf{b}_{3k} \tag{9}$$

$$\mathbf{v}_k = \alpha \cdot \mathbf{v} + (1-\alpha)\mathbf{v}_{k-1} \tag{10}$$

$$\mathbf{y}_k = \alpha \cdot \text{ReLU}(\mathbf{y}) + (1-\alpha)\mathbf{y}_{k-1} \tag{11}$$

Fig. 2 and expressions (7)-(11) describe a layer of the proposed PU-DetNet architecture, where the dashed box represents a single layer of the architecture with the trainable parameters $\theta_k = \{\mathbf{W}_{1k}, \mathbf{b}_{1k}, \mathbf{W}_{2k}, \mathbf{b}_{2k}, \mathbf{W}_{3k}, \mathbf{b}_{3k}, \delta_k\}_{k=1}^{K}$. When such k iterations are unfolded, a DL architecture with k layers analogous to conventional deep network is formed, along with the explainability of an analytical based approach. Moreover, every layer has a unique set of trainable weights and biases. Fig. 3 describes the structural relationship between the input nodes, the hidden nodes, and the output nodes of a single layer of PU-DetNet. In particular, Fig. 3(a) describes the process of obtaining the propagation vector $\mathbf{v}_k$ of a given layer, while Fig. 3(b) describes how a layer computes $\mathbf{y}_k$. As seen in both figures, the input sequence of nodes receives as its input the spectrum dataset ($\mathbf{x}$), the predicted output provided by the previous layer ($\mathbf{y}_{k-1}$), and the propagation vector of the previous layer ($\mathbf{v}_{k-1}$). The architecture uses a sequence of hidden nodes $\mathbf{z}_k$ with parameters $\mathbf{W}_{1k}, \mathbf{b}_{1k}$ to expand upon the knowledge provided by the input nodes. Furthermore, the propagation vector $\mathbf{v}_k$ is obtained that passes on the inference obtained from layer k to layer $k+1$. After obtaining $\mathbf{v}_k$, the architecture obtains the layer's estimated output $\mathbf{y}_k$. From Fig. 2 and Fig. 3, it can be noted that the intermediate outputs of each layer $\mathbf{v}_k$ and $\mathbf{y}_k$ are obtained via different sets of weights and biases: $(\mathbf{W}_{2k}, \mathbf{b}_{2k})$ and $(\mathbf{W}_{3k}, \mathbf{b}_{3k})$, respectively. The output $\mathbf{y}_k$ of the k^{th} layer is also used to calculate the loss value of layer k which further contributes to the aggregate loss function as described in the following subsection.

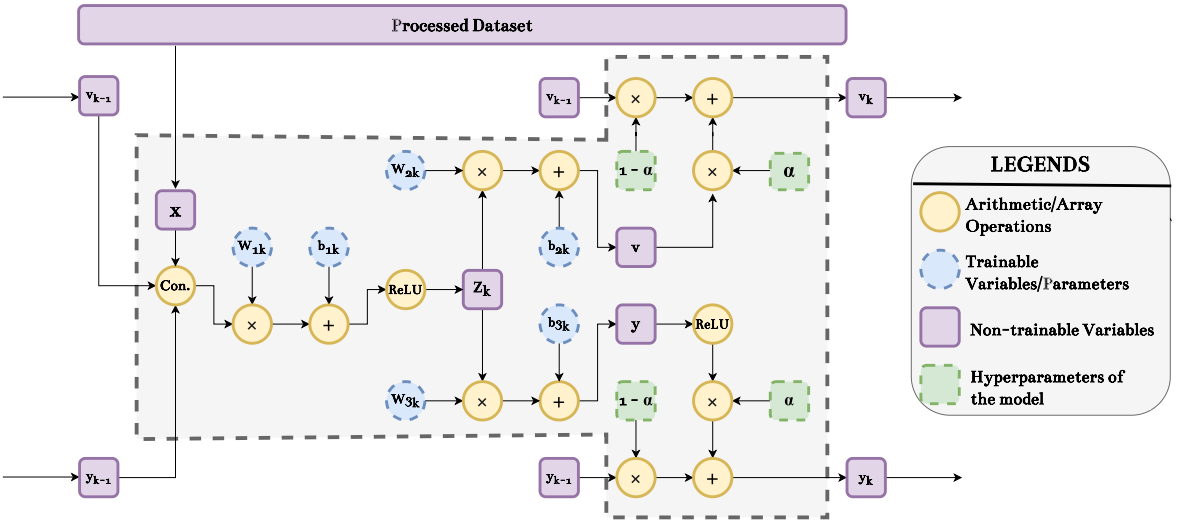
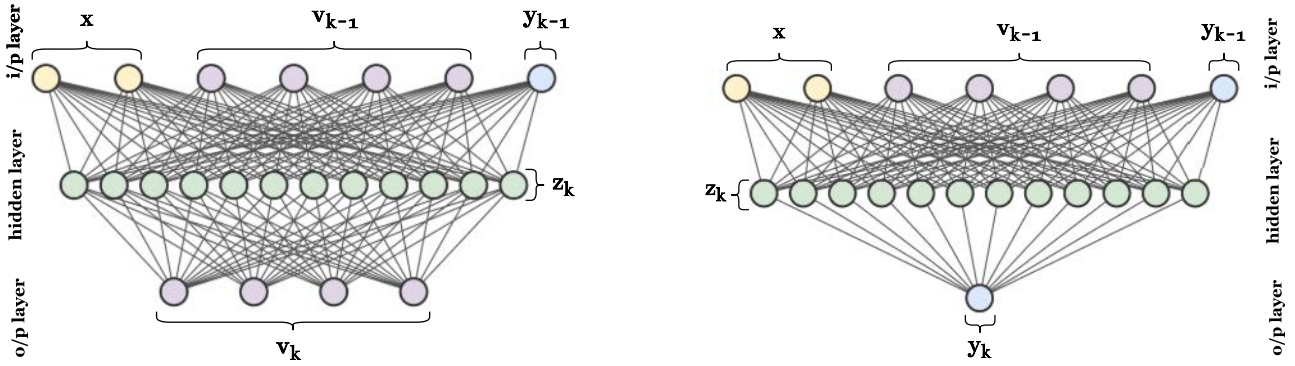


FIGURE 2. Architecture of a single layer of PU-DetNet.



(a) Obtaining propagation vector $\mathbf{v}_k$

(b) Obtaining output $\mathbf{y}_k$

FIGURE 3. Nodal structure of a single layer PU-DetNet.

B. DESCRIPTION OF THE LOSS FUNCTION

A loss function is typically used with gradient descent to update trainable parameters of the model, such that the value of the loss function reduces in successive epochs of the training process. Since the problem under consideration is binary classification (i.e., whether PU is present or not), the intuitive choice of the loss function is binary cross-entropy (BCE). The BCE between the ground truth and the architecture output for one data sample is described as [42]:

$$l_{nk} = y_n^t \cdot \log(P(y_n)_k) + (1 - y_n^t) \cdot \log(1 - P(y_n)_k), \quad (12)$$

where y_n^t is the ground truth of n^{th} data sample and $P(y_n)_k$ is the probability of y_n being true, i.e., the estimated output of the k^{th} layer of architecture.

During the training part, learning architectures calculate the loss function of a specific portion of the dataset at once. In case of batch gradient descent, the architecture calculates the loss value for the entire dataset as follows:

$$l_k = \frac{1}{N} \cdot \sum_{n=1}^{N} l_{nk} = \frac{1}{N} \cdot \sum_{n=1}^{N} [y_n^t \cdot \log(P(y_n)_k) + (1 - y_n^t) \cdot \log(1 - P(y_n)_k)]. \quad (13)$$

Unlike neural networks where the loss function is calculated using the output of just the final layer, PU-DetNet is designed such that it accounts for the loss of every single layer during the training process. Therefore, we use the following loss function which combines the losses of all layers as follows:

$$l = \frac{1}{K} \sum_{k=1}^{K} l_k = \frac{1}{K} \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} [y_n^t \cdot \log(P(y_n)_k) + (1 - y_n^t) \cdot \log(1 - P(y_n)_k)]. \quad (14)$$

The proposed PU-DetNet architecture aims to learn the underlying structure of the PU detection problem in

Algorithm 1 Pre-Processing the Data

Input: X_{Signal}, n
Output: $X_{Energies}$, labels
data_size $\leftarrow \frac{\text{length}(X_{Signal})}{n}$
for SNR $\leftarrow -20$dB to 4dB **do**
  $X_{Noisy} \leftarrow X_{Signal}$ + AWGN % ($\mathscr{H}_1$)
  for i $\leftarrow$ 1 to data_size **do**
    $X_{Energies} \leftarrow$ Append(Energy(X_{Noisy}, i, data_size))
    labels $\leftarrow$ Append(1)
    $X_{Energies} \leftarrow$ Append(Energy(AWGN, i, data_size))
    labels $\leftarrow$ Append(0)
  end for
end for
return $X_{Energies}$, labels

a computationally efficient manner, without compromising the accuracy. Hence, the architecture is designed to be deep in layers, such that it can capture the complicated patterns of a highly noisy dataset, i.e., even at low SNR regimes. One common problem of deep architectures is increased computational cost. To overcome the same, we incorporate the concept of loss binding where layers of the architecture are bound in a way such that their outputs and hence their losses are tied together and thus behave similarly in terms of estimation. In implementation, we add the losses of all layers to obtain an aggregate loss of the whole architecture. This aggregate loss is further used by gradient descent like optimization methods.

This idea was inspired by the concept of auxiliary classifiers presented in GoogleNet [43] that aimed to discriminate the lower layers in the network such that intermediate layers could be used directly as classifiers. Due to this, a non conventional design of the loss function is described in (14). The effect of binding the loss function of each layer is that the every layer is constrained to be optimized and be equally close to the ground truth. This effect further leads to infer that even though the training is performed on a deep architecture, one can use outputs from even the shallower layers of the model. In implementing such idea where only shallower layers are required while testing, we observe the need for a high number of layers during training. Theoretically, the higher number of training layers can help build a more descriptive network as the number of trainable parameters increases. Due to the loss binding, the layers succeeding any given layer form a backward feedback system, as the optimization of the deeper layers can affect the optimization of preceding layers as well. This is analyzed and comprehensively described in the numerical results section.

C. PROCESSING THE ARCHITECTURE

In this subsection, we first describe the pre-processing of the data in order to make it suitable for training and testing of the proposed PU-DetNet architecture. As described in **Algorithm 1**, AWGN noise with power $\sigma_w^2 = \sigma_{X_{Signal}}^2/\text{SNR}$ is generated and added to the signal at each SNR, which forms the data for hypothesis ($\mathscr{H}_1$). Similarly, for hypothesis ($\mathscr{H}_0$), only the AWGN noise is considered. Energy is computed for both hypotheses and labelled accordingly. These processed datasets are further used to compare the performance of the proposed PU-DetNet scheme with the other state-of-the-art schemes.

Algorithm 2 Training the Proposed PU-DetNet Scheme

Input: X_{train}, y_{train}, epochs
Output: Parameters
Parameters $\leftarrow$ RandomInitialization()
for i $\leftarrow$ 1 to epochs **do**
  $y_{est} \leftarrow$ EstimateOutput(X_{train}, Parameters)
  loss $\leftarrow$ CalculateBindedLoss(y_{est},y_{train})
  Parameters $\leftarrow$ UpdateParameters(loss, Parameters)
end for
return Parameters

1) TRAINING PHASE

We divide the processed data into train (X_{train}, y_{train}) and test (X_{test}, y_{test}) datasets. The parameters of the architecture are randomly initialized and updated iteratively. The detailed process of training the PU-DetNet architecture is described in **Algorithm 2**. The function EstimateOutput() is based on equations (7) - (11), while the function CalculateBindedLoss() is based on (14). The function UpdateParameters() is an optimization function and in the case of PU-DetNet, it is chosen to be Adam Optimizer [44] due to its lower computational cost and lower dependence on hyperparameter tuning.

2) TESTING PHASE

The optimal parameters obtained after the process of training the model collectively represent the final architecture which is ready to be tested. While testing, the number of correct and incorrect estimations (classifications) are recorded and the performance metrics probability of detection (P_d) and probability of false alarm (P_f) are calculated. The detailed process of testing the PU-DetNet architecture is described in **Algorithm 3**.

To evaluate the ability of the proposed PU-DetNet architecture to detect the PU signal correctly, we compute the performance metrics P_d and P_f as:

$$P_d = \frac{\text{No. of } \mathscr{H}_1 \text{ samples correctly classified as } \mathscr{H}_1}{\text{Total no. of } \mathscr{H}_1 \text{ samples fed}} \quad (15)$$

$$P_f = \frac{\text{No. of } \mathscr{H}_0 \text{ samples incorrectly classified as } \mathscr{H}_1}{\text{Total no. of } \mathscr{H}_0 \text{ samples fed}} \quad (16)$$

Furthermore, in DSA/CR systems, spectrum sensing has direct impact on throughput [45]. The relation between sensing time (T_s) and the throughput for SU can be expressed as:

$$\text{Throughput} = \frac{(T - T_s)}{T} \times \text{B} \cdot \log_2(1 + \text{SNR}), \quad (17)$$

TABLE 3. Channels measured in empirical setup and USRP configuration [26].

Radio Technology	Channel Number	f_{start} (MHz)	f_{center} (MHz)	f_{stop} (MHz)	Signal Bandwidth (MHz)	Decimation Rate	Sampled Bandwidth (MHz)
FM broadcasting	–	96.500	96.700	96.900	0.2	64	1
E-GSM 900 DL	77	950.2	950.4	950.6	0.2	64	1
DCS 1800 DL	690	1839.6	1840.8	1841	0.2	64	1
UHF television (Band IV)	U-33	566	570	574	8	8	8

TABLE 4. Parameters of the configurations of the 5G-NR test models for data simulation.

	Config. A	Config. B	Config. C	Config. D
Frequency Range	450 MHz - 6 GHz	450 MHz - 6 GHz	24.25 GHz - 52.6 GHz	24.25 GHz - 52.6 GHz
Modulation	64QAM	QPSK	QPSK	64QAM
Channel Bandwidth	10 MHz	10 MHz	50 MHz	50 MHz
Subcarrier Spacing	60 kHz			
Duplex Mode	Frequency Division Duplex (FDD)			

where T is the frame duration in DSA/CR networks, and B is the bandwidth. These metrics are further used in Section V to demonstrate experimental results.

Algorithm 3 Testing the Proposed PU-DetNet Scheme

Input: X_{test}, y_{test}, Parameters*
Output: P_d, P_f

```
N ← length(X_test)
Out_incorr ← 0
Out_corr ← 0
for i ← 1 to N do
    y_est ← EstimateOutput(X_test(i),Parameters)
    if y_est = 1 and y_test(i) = 1 then
        Out_corr ← Out_corr + 1
    else if y_est = 1 and y_test(i) = 0 then
        Out_incorr ← Out_incorr + 1
    end if
end for
P_d ← Out_corr / N
P_f ← Out_incorr / N
return P_d, P_f
```

IV. CONSIDERED DATASETS

To validate the proposed scheme, we have considered data from five different datasets corresponding to signal formats whose detection performance is relevant in several realistic spectrum sharing and coexistence scenarios.

A. EMPIRICAL TEST-BED SETUP

We deployed an empirical test-bed setup on the roof-top of the School of Engineering and Applied Science, Ahmedabad University for spectrum data acquisition, the details of which are reported in [17], [26], and [27], omitted here for the sake of brevity. The empirical measurement setup is shown in Fig. 4. The aforementioned setup was used to capture raw signal data of four bands: FM, GSM, DCS, and UHF. The specifications of signal data captured are described in Table 3. This dataset contains more than 1,000,000 samples per band.

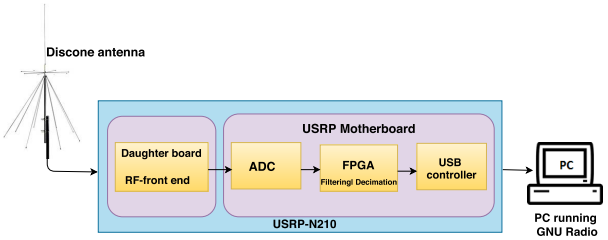


FIGURE 4. Empirical measurement test-bed setup [26].

B. 5G NEW RADIO SIMULATED DATASET

The detection of 5G signals is relevant in spectrum sharing scenarios such as those enabled by the 5G NR - unlicensed technology, where the presence of 5G NR waveforms in unlicensed bands needs to be detected. We used MATLAB 5G toolbox in generating 5G waveforms which are compliant with 3GPP Release 15 [46]. Test models from the waveform generator were used to obtain signal data from four different configurations, each having a different set of parameters as shown in Table 4. This dataset contains 153,600 samples for each configuration.

C. DEEPSIG DATASET

The publicly available DeepSig dataset [47] (RADIOML 2016.10A) contains signal data consisting of 11 modulations (8 digital, 3 analog). This dataset was first released at the 6th annual GNU radio conference and is useful in the context of this work to assess the performance of the proposed PU-DetNet scheme with commonly used signal modulations. While typically used for modulation classification, we utilize

TABLE 5. Parameters of the DVB-S2 simulation.

Parameter	Value
Samples per symbol	2
Rolloff Factor	0.35
Channel Bandwidth (Hz)	3.6×10^7
Carrier Frequency Offset	3×10^6
Sampling Clock Offset	5 ppm

this dataset for PU-detection after processing it as described in Algorithm 1. It should be noted that unlike other mentioned datasets, no noise was added to the DeepSig dataset since it is considered that the AWGN noise is already present in the signal (i.e., for $\mathscr{H}_1$) at different SNR. However for $\mathscr{H}_0$, the AWGN noise with power $\sigma_w^2 = \sigma^2_{\mathrm{X_{Signal}}}/\mathrm{SNR}$ at each respective SNR were generated. Out of the 11 modulations, signal data of the modulations BPSK, QPSK, 64-QAM and GFSK were considered and processed. 128,000 samples per modulation scheme were considered.

D. SATELLITE COMMUNICATIONS DATASET

CR has been proposed to enable spectrum sharing not only in terrestrial but also in satellite communication bands and therefore the performance of spectrum sensing methods with satellite communication signals is relevant as well. We utilized MATLAB's satellite communications toolbox [48] that provides standards-based tools for designing, simulating, and verifying satellite communications systems and links. To obtain the dataset, an end-to-end DVB-S2 simulation with RF impairments and corrections was used. The parameters of the configuration are described in Table 5. This dataset contains 800,000 samples.

E. RADAR DATASET

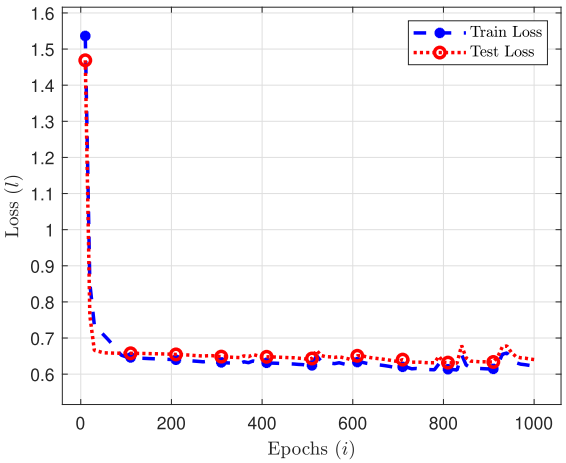
The software tool [49] provides a radio frequency (RF) dataset generator for incumbent signals in the 3.5 GHz citizens broadband radio service (CBRS) band, which is another practical scenario where detection and sensing is relevant. The pulse modulation types for the radar signals and their parameters are selected based on national telecommunications and information administration (NTIA) testing procedures for environmental sensing capability (ECS) certification. Using the generator provided, two simulated radar waveforms with varying parameters were obtained to validate the PU-DetNet scheme. The parameters of the configurations set up to obtain these waveforms are described in Table 6. This dataset contains 800,000 samples per SNR for each configuration.

V. EXPERIMENTAL RESULTS

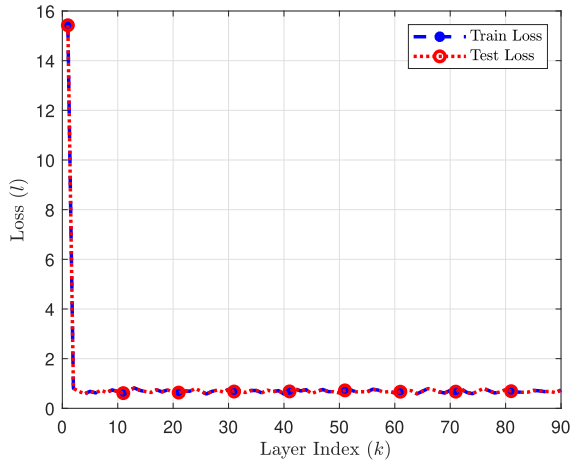
In this section, we describe the experiments comparing the proposed PU-DetNet scheme and other state-of-the-art ML based sensing schemes. The training and testing of the proposed PU-DetNet architecture were performed with the aid of the Tensorflow library [50]. For processing the proposed

TABLE 6. Parameters of the simulated radar waveform.

Parameter	Config. A	Config. B
Pulses per second	2130	2340
Pulse width	5×10^{-6} m	
Pulses per burst	20	14
Chirp Width	8×10^7 m	6×10^7 m
Chirp Direction	Down	Up
Sampling Frequency	10^7 Hz	



(a) Aggregate loss with respect to epochs



(b) Layer-wise variation of loss function value

FIGURE 5. Behaviour of loss function for FM band with SNR = -5 dB (similar trend was observed for other datasets).

architecture and benchmark architectures, we used a 12GB NVIDIA Tesla K80 GPU offered by Google Colab. For training and testing, signal data with SNR values from -20 dB to $+4$ dB were considered, unless otherwise mentioned. Moreover, to ensure that the model is not biased, equal number of data points in both hypotheses were generated. Training dataset size was kept approximately 70%, while the remaining 30% was used for testing. Moreover, all the results of P_d v/s SNR are plotted by training the model such that $P_f \approx 0.05$.

TABLE 7. Variation in P_d with respect to number of layers used for training (columns) and number of layers used for testing (rows) at SNR = -4 dB, Empirical testbed setup.

Training / Testing	2	5	15	25	50	100
2	**0.6484**	0.6993	0.7293	0.7620	0.8729	0.9256
5	-	**0.7564**	0.7864	0.8252	**0.9443**	**0.9755**
15	-	-	**0.8379**	**0.8704**	0.8570	0.8961
25	-	-	-	0.8406	0.8532	0.8530
50	-	-	-	-	0.7958	0.8183
100	-	-	-	-	-	0.8002

A. MODEL EVALUATION

In learning architectures, performance is often evaluated by the value of a loss function which describes the behaviour of the model with respect to the ground truth. The loss function of the PU-DetNet architecture is described in (14). This loss is calculated and then plotted with respect to epochs, i.e., the number of times a dataset is passed through the architecture during the training phase.

It can be observed in Fig. 5 (a) that the train and test loss decrease with the number of epochs. An elbow-shaped curve is obtained with a cut-off at around 100 epochs. This states that the loss reduction slows down after 100 epochs. The reduction in loss over epochs suggests that with increasing training, the model performs better on data. It can also be seen that the loss over test data remains slightly higher than the loss over train data which is quite intuitive given the fact that test data were unseen by the model when this result was obtained. Fig. 5 (b) demonstrates the PU-DetNet's property of loss binding as described in Section III. It can be observed that the loss value quickly drops and almost becomes constant after layer 2. Thus, we can infer that after the architecture is completely trained on the dataset, the losses across the layers are bound to have similar values. However, we would like to highlight that all the layers are necessary for training, as confirmed through Table 7.

Table 7 shows the value of P_d obtained from a model where for a given value in the table, the column index represents the number of layers present in the architecture while training, and the row index represents the number of layers used for testing. The value indicated in boldface represents the optimum value of P_d for a given number of training layers. We can notice that for a model with 100 training layers, the optimal output ($P_d = 0.9755$) is obtained at shallower layer (layer 5 in this case). The need for higher number of layers while training is thus clearly observed in Table 7. When 50 or 100 layers are used, the optimum output is provided by the 5th layer, however, this does not mean that the network could achieve the same level of performance with a lower number of layers for training, since in the case of a network with only 5 layers; the accuracy ($P_d = 0.7564$) is significantly lower than in the case of 100 layers ($P_d = 0.9755$). This is because feedback across all layers helps optimize the performance of shallower layers. However, output from the deep layers tends to get overfitted (due to more testing layers) and hence high generalization error (or lower performance) at the deep

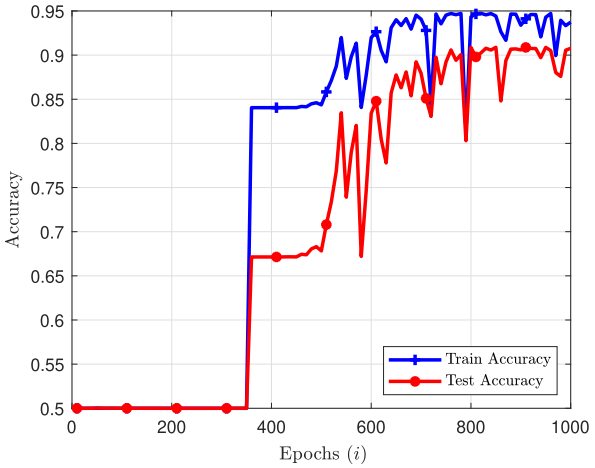


FIGURE 6. Comparison of train and test accuracies with respect to epochs for (Empirical dataset, FM broadcasting band, SNR = -5 dB).

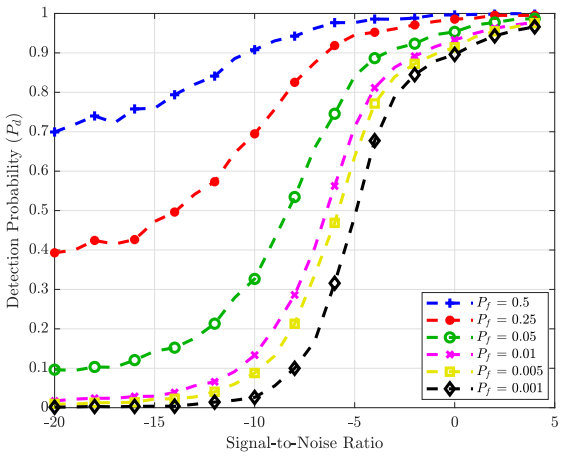
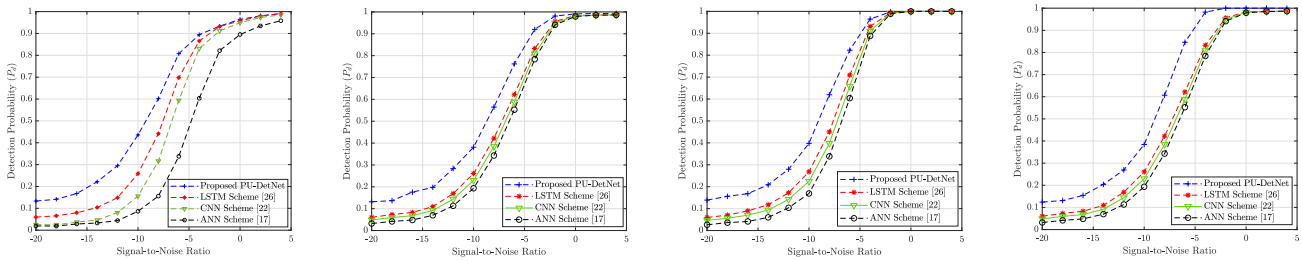


FIGURE 7. P_d v/s SNR for the proposed PU-DetNet scheme at various values of P_f (Empirical testbed dataset, FM band).

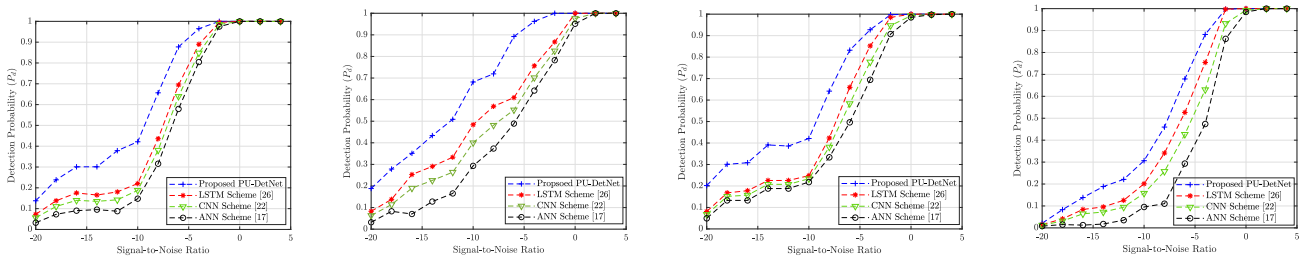
layers i.e., in the last layers. As the number of training layers decrease, the overfitting decreases and the optimum shifts towards higher number of testing layers. The training layers that form a backward feedback system with the aid of loss binding are not observed to be necessary in making a decision regarding the presence of PU. Thus, the layers succeeding the layer that was used while testing were discarded and a computationally efficient system was formed.

Fig. 6 shows the plot of accuracy v/s epochs for the PU-DetNet architecture in classifying correctly over the train and test data. An observation from this figure can be made that the accuracies remain at 50% till around 350 epochs. It can be inferred that the model does not learn much about the underlying structure of the data and hence displays a random behaviour on these binary labelled data. However, after around 350 epochs the accuracies spike and the model starts learning the underlying structure of the dataset as the training accuracy immediately rises up to 85% and the test accuracy rises up to 65%. With increasing epochs, the gap



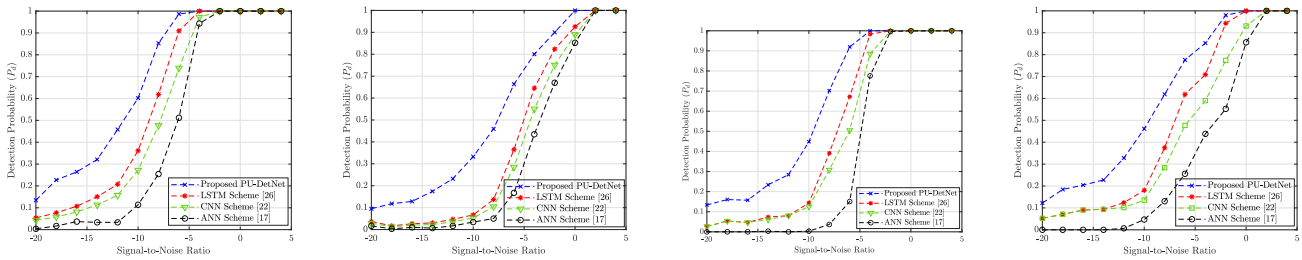
(a) FM Band (b) GSM Band (c) DCS Band (d) UHF Band

FIGURE 8. Comparison and validation of detection probability on the empirical test-bed data set for the considered spectrum sensing schemes: proposed PU-DetNet, LSTM-based sensing [26], CNN-based sensing [22], and ANN-based sensing [17] ($P_f \approx 0.05$).



(a) 5G NR configuration A (b) 5G NR configuration B (c) 5G NR configuration C (d) 5G NR configuration D

FIGURE 9. Comparison and validation of detection probability on 5G simulated Dataset for the considered spectrum sensing schemes: proposed PU-DetNet, LSTM-based sensing [26], CNN-based sensing [22], and ANN-based sensing [17] ($P_f \approx 0.05$).



(a) BPSK Modulation (b) QPSK Modulation (c) GFSK Modulation (d) 64-QAM Modulation

FIGURE 10. Comparison and validation of detection probability on DeepSig dataset for the considered spectrum sensing schemes: proposed PU-DetNet, LSTM-based sensing [26], CNN-based sensing [22], and ANN-based sensing [17] ($P_f \approx 0.05$).

between train and test accuracy slowly decreases as they converge. This indicates that with increasing epochs, the model's ability to generalize improves on unseen data. Furthermore, we observe the probability of the model to detect the presence of PU in spectrum with respect to the SNR value for varying values of P_f in Fig. 7.

B. PERFORMANCE COMPARISON WITH THE BASELINE MODELS

The performance of PU-DetNet is compared with the state-of-the-art baseline models: The LSTM [26], CNN [22] and the ANN [17]. For LSTM and ANN schemes, the models mentioned in the base papers were implemented. For CNN based scheme, we adopted the 1D-CNN model due to the one dimensional structure of data. The model consist of two convolutional layers (CL) with ReLu activation function followed by the fully connected (FC) layer. The number of Kernels (n_{ker}) considered was 3 with shape (s_{ker}) of each kernel as 4 (can also be viewed as 4×1 due to 1D CNN). For fair evaluation and comparison, all the schemes were trained, tested on same datasets (for various radio technologies as discussed in Section-IV). Hyperparameters were tuned to ensure optimum results as summarized in Table 8.

Fig. 8 shows the comparison of P_d on the empirical testbed dataset. An average gain of 42.04%, 57.42% and 78.03% is observed at -10dB with respect to LSTM, CNN and ANN schemes respectively on this dataset. Fig. 9 shows the comparison with respect to the 5G simulated dataset and an average gain of 47.43%, 63.74% and 86.11% is observed at -10 dB with respect to LSTM, CNN and ANN schemes respectively. Fig. 10 validates the PU-DetNet scheme on the DeepSig dataset and an average gain of 56.03%, 84.66% and

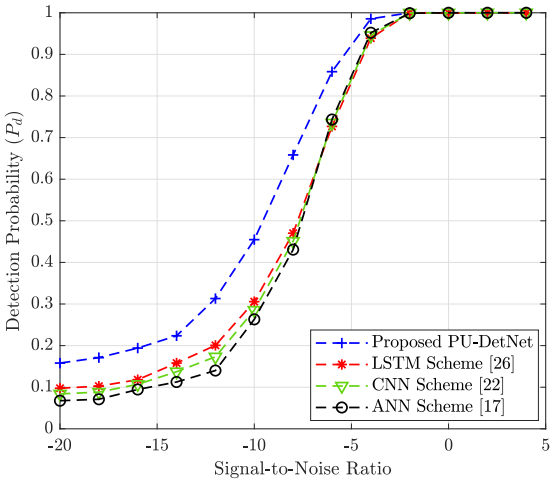
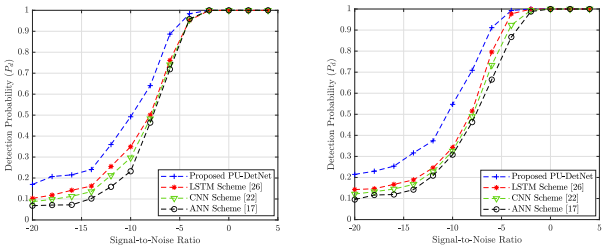


FIGURE 11. Comparison and validation of detection probability on satellite communications dataset for the considered spectrum sensing schemes: proposed PU-DetNet, LSTM-based sensing [26], CNN-based sensing [22], and ANN-based sensing [17] ($P_f \approx 0.05$).

TABLE 8. Hyperparameters of the considered schemes.

Hyperparameter	PU-DetNet	LSTM [26]	CNN [22]	ANN [17]
Epochs Trained	1000	20	50	40
Layers Trained	100	2	2	4
Tested on Layer No. (n_l)	5	2	2	4
Nodes per Hidden Layer (h_s)	7	3	2 CL, 1 FC	8
No. of Kernels (n_{ker})	-	-	3	-
Kernel Shape (s_{ker})	-	-	4×1	-
Loss function	Binded BCE	BCE		
Optimizer	Adam Optimizer			
Activation function	ReLU			



(a) Simulated radar config. A (b) Simulated radar config. B

FIGURE 12. Comparison and validation of detection probability on radar dataset for the considered spectrum sensing schemes: proposed PU-DetNet, LSTM-based sensing [26], CNN-based sensing [22], and ANN-based sensing [17] ($P_f \approx 0.05$).

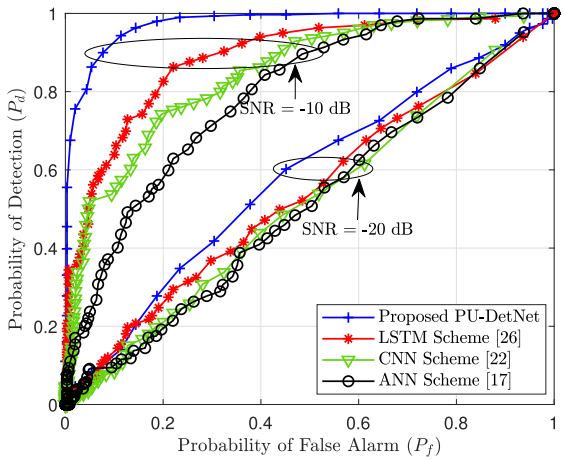


FIGURE 13. ROC comparison of proposed scheme with LSTM-based sensing [26], CNN-based sensing [22], and ANN-based sensing [17] at SNR = −10 dB and SNR = − 20 dB (DeepSig Dataset, BPSK Modulation).

128.32% is observed at -10 dB with respect to LSTM, CNN and ANN schemes respectively. Fig. 11 shows the comparison with respect to the satellite communications dataset and an average gain of 39.21%, 45.38% and 53.37% is observed at −-10 dB with respect to LSTM, CNN and ANN schemes respectively. Furthermore, Fig. 12 provides the comparison with respect to the radar dataset and an average gain of 40.06%, 58.50% and 63.86% is observed at -10 dB with respect to LSTM, CNN and ANN schemes respectively. We can notice that the proposed scheme consistently outperforms the benchmarks scheme in terms of P_d. Although varying with different datasets, it can be said that the proposed PU-DetNet scheme can yield an acceptable value of ($P_d = 0.9$) at 2 dB to 6 dB of SNR lesser than the state-of-the-art schemes.

Fig. 13 shows the comparison of the receiver operating characteristic (ROC) curves. It can be observed that for a given P_f, the PU-DetNet scheme yields a higher P_d than the other baseline schemes. It is also observed that for lower values of SNR, all schemes yield a lower P_d for a given P_f but the proposed method still outperforms the state-of-the-art LSTM, CNN and ANN schemes.

Table 9 shows the comparison of precision, recall and F1 score [51] observed for the proposed scheme and the benchmark schemes for data comprising of samples with SNR $= -5$ dB. Values marked in boldface represent the optimal value for each set of comparison. In terms of precision and recall, the proposed PU-DetNet scheme provides the best accuracy in most cases; in those cases where it does not, the achieved accuracy is very similar to the highest attained value. F1 score is commonly used to evaluate the accuracy of an algorithm and can be interpreted as a weighted average of the precision and recall. It can be appreciated from the table that the proposed PU-DetNet scheme outperforms the other state-of-the-art sensing schemes in terms of F1 score, thus highlighting its ability to provide more accurate results.

C. COMPUTATIONAL ANALYSIS

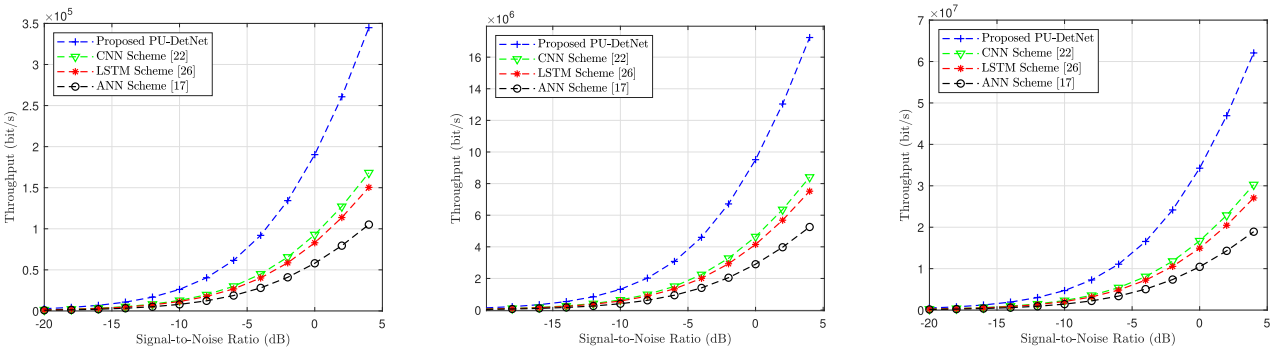
In addition to the detection performance, we perform the computational analysis of the proposed PU-DetNet scheme and the baseline models. The number of FLOPs is one of the measure of computation that describes the total number of instructions a processor has to execute to perform the specific operation. FLOPs calculation can be done by analyzing the structure of a model and the final value depends on the hyperparameters of the model. Table 10 analyzes the

TABLE 9. Precision, Recall and F1 score comparison of the proposed PU-DetNet scheme with the baseline models on considered datasets for SNR = −5 dB.

Dataset		Precision				Recall				F1 Score			
		ANN	CNN	LSTM	PU-DetNet	ANN	CNN	LSTM	PU-DetNet	ANN	CNN	LSTM	PU-DetNet
Empirical	FM	0.769	0.792	0.856	**0.924**	0.718	0.778	0.841	**0.892**	0.743	0.785	0.848	**0.908**
	E-GSM	0.791	0.814	0.873	**0.917**	0.744	0.794	0.864	**0.913**	0.767	0.804	0.868	**0.915**
	DCS	0.782	0.779	**0.889**	0.884	0.732	0.843	0.837	**0.897**	0.756	0.810	0.862	**0.89**
	UHF	0.723	0.836	0.844	**0.941**	0.773	0.771	**0.879**	0.865	0.747	0.802	0.861	**0.901**
5G New radio	Config. A	0.764	0.823	0.861	**0.935**	0.711	0.789	0.840	**0.928**	0.737	0.806	0.850	**0.931**
	Config. B	0.734	0.791	**0.886**	0.872	0.776	0.769	0.845	**0.893**	0.754	0.780	0.865	**0.882**
	Config. C	0.744	0.829	0.839	**0.906**	0.724	0.841	0.838	**0.880**	0.734	0.835	0.838	**0.893**
	Config. D	0.783	0.819	0.879	**0.921**	0.765	0.844	0.839	**0.909**	0.774	0.831	0.859	**0.915**
DeepSig	BPSK	0.785	0.782	0.859	**0.925**	0.747	0.781	**0.883**	0.872	0.766	0.781	0.871	**0.898**
	QPSK	0.777	0.848	0.843	**0.936**	0.724	0.794	0.873	**0.921**	0.75	0.820	0.858	**0.928**
	GFSK	0.731	0.810	**0.884**	0.879	0.759	0.824	0.841	**0.874**	0.745	0.817	0.862	**0.876**
	QAM-64	0.769	0.844	0.840	**0.912**	0.744	0.783	**0.887**	0.885	0.756	0.812	0.863	**0.898**
Radar	Config. A	0.780	0.771	0.878	**0.899**	0.724	0.818	0.848	**0.893**	0.751	0.794	0.863	**0.896**
	Config. B	0.719	0.819	0.881	**0.906**	0.778	0.768	0.850	**0.904**	0.747	0.793	0.865	**0.905**
Satellite Comm.		0.788	0.822	0.849	**0.883**	0.747	0.791	0.844	**0.891**	0.767	0.806	0.846	**0.887**

TABLE 10. Computational analysis.

Scheme	FLOPs Calculation	No. of FLOPs	Train Time (Whole Dataset)	Inference Time
PU-DetNet	$2 \cdot n_l((x_s + s_s + v_s) \cdot (h_s - 1))$	168	937.38 s	4.86 μs
LSTM	$8 \cdot 2 \cdot (x_s(h_s - 1) + n_l h_s(h_s - 1))$	448	1,397.32 s	58.51 μs
CNN	#CL $\cdot$ ($n_{\text{ker}} \cdot s_{\text{ker}} \cdot$ ker. o/p shape) + #FC $\cdot$ (2 $\cdot$ i/p size $\cdot$ o/p size)	384	511.2 s	53.60 μs
ANN	$2 \cdot (x_s(h_s - 1) + n_l h_s(h_s - 1))$	476	311.9 s	71 μs



(a) Empirical FM Dataset (b) 5G NR test model (Config. A) (c) Satellite communication dataset

FIGURE 14. Throughput performance of the considered spectrum sensing methods: proposed PU-DetNet scheme, CNN [22], LSTM [26], and ANN-based sensing schemes [17] over various datasets.

FLOPs calculation to compute the total number of FLOPs for each scheme. In addition to the number of FLOPs, the time each scheme consumed for training and testing were also observed. As mentioned before, all the schemes were processed on a 12GB NVIDIA Tesla K80 GPU offered by Google Colab. Considering hyperparameters from Table 8, it is intuitive that due to its deep nature the PU-DetNet scheme takes more time than ANN and CNN in training. However, PU-DetNet still manages to train completely in less time than LSTM as observed in Table 10. It is worth mentioning that training is often performed once before the network is deployed in real system for operation. Therefore, a high training time is not necessarily inconvenient, in particular if it allows a better performance/accuracy and even shorter execution time as demonstrated by the inference time. Moreover, it can be observed from Table 10 that the number of FLOPs for proposed PU-DetNet scheme reduces by 62.50%, 56.25%, 64.70% w.r.t. LSTM, CNN and ANN schemes, respectively. Although the FLOPs account for the number of arithmetic operations undergone to perform a task, the actual time consumed by a scheme may vary depending upon biases, non linear activation functions and the complexity of the type of arithmetic/matrix operation. We can also notice that the inference time (per single sample) reduces by 91.69%, 90.9% and 93.15% over the LSTM, CNN and ANN schemes respectively. Hence PU-DetNet significantly outperforms the baseline schemes not only in terms of detection performance but also in terms of computation. The advantage is inevitably due to the combined analytical-based and data-driven approach in the unfolded architecture. We would

like to highlight that although the simple ML models would comparatively have low computational cost, however such approach would require extensive feature engineering. Furthermore, the state-of-the-art schemes already outperforms such simple ML based schemes in terms of detection performance, already reported in previous work [27]. In addition, the analytical-based sensing approaches are also reported to be outperformed by the data-driven approaches in [26] and [27], and hence not shown in this work for the sake of brevity.

D. THROUGHPUT ANALYSIS

In this subsection, we demonstrate the application of the proposed scheme. It is intuitive to note that the sensing time (time employed to sense PU) is analogous to the inference time of the ML architecture, as enlisted in the last column of Table 10. Fig. 14 shows the plot of throughput v/s SNR validated over various datasets for the proposed PU-DetNet, LSTM and ANN based sensing schemes (with $T = 0.1$ms, and B as per the considered dataset). We can notice that the proposed PU-DetNet has an average gain of 56.39%, 51.23%, and 69.52% as compared to LSTM, CNN and ANN scheme respectively, at SNR $= -6$ dB. The gain in throughput is due to the fact that the inference time for proposed PU-DetNet scheme is much shorter (and hence quicker detection) as compared to other schemes in Table 10.

VI. CONCLUSION

In this work, a deep unfolding approach is introduced for spectrum sensing problem that harvests the strength of both: analytical-based and data-driven approaches. The Primary User-Detection Network (PU-DetNet) is proposed to overcome the shortcomings of ML/DL frameworks like high computational cost. A unique technique is described which involves binding the loss function across all layers that helps in reducing the computational cost significantly. The proposed scheme is thoroughly evaluated on five different datasets. The proposed scheme outperforms state-of-the-art spectrum sensing schemes in all cases. Furthermore, it was observed that at SNR $= -10$ dB, probability of detection is improved by a significant amount compared to the LSTM approach (between 39% to 56%), CNN approach (45% to 84%) and the ANN approach (between 53% and 128%) using empirical, 5G new radio, DeepSig, satellite communications and radar datasets. The accuracy of proposed scheme outperforms other existing schemes in terms of the F1-score. Additionally, inference time reduces by 91.69%, 90.90%, and 93.15%, while FLOPs reduces by 62.50%, 56.25%, 64.70% w.r.t. LSTM, CNN and ANN schemes, respectively. Moreover, the proposed scheme also shows improvement in throughput by 56.39%, 51.23%, and 69.52% as compared to LSTM, CNN and ANN schemes respectively, at SNR $= -6$ dB. This work provides a comprehensive study and offers a computationally efficient detection framework suitable for devices with low computational abilities, as envisaged in the next generation of wireless networks.

ACKNOWLEDGMENT

The authors thank the Department of Engineering—Boston College, Machine Intelligence, Computing and xG Networks (MICxN) Research Laboratory—Ahmedabad University, and the University of Liverpool for providing the infrastructural support. Finally, they also thank the Associate Editor for handling the review of this paper, and anonymous reviewers for their valuable suggestions in improving the quality of this manuscript.

REFERENCES

[1] H. Wymeersch, D. Shrestha, C. Morais de Lima, V. Yajnanarayana, B. Richerzhagen, M. Furkan Keskin, K. Schindhelm, A. Ramirez, A. Wolfgang, M. Francis de Guzman, K. Haneda, T. Svensson, R. Baldemair, and S. Parkvall, "Integration of communication and sensing in 6G: A joint industrial and academic perspective," 2021, *arXiv:2106.13023*.

[2] J. Lunden, V. Koivunen, and H. V. Poor, "Spectrum exploration and exploitation for cognitive radio: Recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 123–140, May 2015.

[3] X. Liu, Q. Sun, W. Lu, C. Wu, and H. Ding, "Big-data-based intelligent spectrum sensing for heterogeneous spectrum communications in 5G," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 67–73, Oct. 2020.

[4] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[5] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proc. IEEE*, vol. 55, no. 4, pp. 523–531, Apr. 1967.

[6] F. F. Digham, M.-S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels," *IEEE Trans. Commun.*, vol. 55, no. 1, pp. 21–24, Jan. 2007.

[7] M. Lopez-Benitez and F. Casadevall, "Improved energy detection spectrum sensing for cognitive radio," *IET Commun.*, vol. 6, no. 8, pp. 785–796, May 2012.

[8] D. K. Patel, B. Soni, and M. Lopez-Benitez, "Improved likelihood ratio statistic-based cooperative spectrum sensing for cognitive radio," *IET Commun.*, vol. 14, no. 11, pp. 1675–1686, Jul. 2020.

[9] B. Soni, D. K. Patel, Z. Ding, Y. L. Guan, and S. Sun, "On sensing performance of multi-antenna mobile cognitive radio conditioned on primary user activity statistics," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3381–3394, May 2022.

[10] Z. Xu and J. Sun, "Model-driven deep-learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 22–24, Jan. 2017.

[11] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[12] T. J. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Oct. 2017.

[13] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[14] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to Pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, 3rd Quart., 2020.

[15] J. Tian, Y. Pei, Y. D. Huang, and Y.-C. Liang, "Modulation-constrained clustering approach to blind modulation classification for MIMO systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 894–907, Dec. 2018.

[16] Y.-J. Tang, Q.-Y. Zhang, and W. Lin, "Artificial neural network based spectrum sensing method for cognitive radio," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Sep. 2010, pp. 1–4.

[17] M. R. Vyas, D. K. Patel, and M. Lopez-Benitez, "Artificial neural network based hybrid spectrum sensing scheme for cognitive radio," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–7.

[18] D. K. Patel, M. López-Benítez, B. Soni, and Á. F. García-Fernández, "Artificial neural network design for improved spectrum sensing in cognitive radio," *Wireless Netw.*, vol. 26, no. 8, pp. 6155–6174, Nov. 2020.

[19] J. Tian, P. Cheng, Z. Chen, M. Li, H. Hu, Y. Li, and B. Vucetic, "A machine learning-enabled spectrum sensing method for OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11374–11378, Nov. 2019.

[20] Z. Shi, W. Gao, S. Zhang, J. Liu, and N. Kato, "Machine learning-enabled cooperative spectrum sensing for non-orthogonal multiple access," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5692–5702, Sep. 2020.
[21] 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.
[22] C. Liu, X. Liu, and Y. Liang, "Deep CNN for spectrum sensing in cognitive radio," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[23] J. Xie, C. Liu, Y.-C. Liang, and J. Fang, "Activity pattern aware spectrum sensing: A CNN-based deep learning approach," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1025–1028, Jun. 2019.
[24] 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.
[25] Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Sensing OFDM signal: A deep learning approach," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7785–7798, Nov. 2019.
[26] 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.
[27] N. Balwani, D. K. Patel, B. Soni, and M. Lopez-Benitez, "Long short-term memory based spectrum sensing scheme for cognitive radio," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 1–6.
[28] J. Xie, J. Fang, C. Liu, and L. Yang, "Unsupervised deep spectrum sensing: A variational auto-encoder based approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5307–5319, May 2020.
[29] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Commun.*, vol. 17, no. 2, pp. 138–148, Feb. 2020.
[30] 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.
[31] M. Xu, Z. Yin, M. Wu, Z. Wu, Y. Zhao, and Z. Gao, "Spectrum sensing based on parallel CNN-LSTM network," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.
[32] A. Hoglund, X. Lin, O. Liberg, A. Behravan, E. A. Yavuz, M. Van Der Zee, Y. Sui, T. Tirronen, A. Ratilainen, and D. Eriksson, "Overview of 3GPP release 14 enhanced NB-IoT," *IEEE Netw.*, vol. 31, no. 6, pp. 16–22, Dec. 2017.
[33] Q. Hu, Y. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, "Iterative algorithm induced deep-unfolding neural networks: Precoding design for multiuser MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1394–1410, Feb. 2021.
[34] J. R. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," 2014, *arXiv:1409.2574*.
[35] A. Jagannath, J. Jagannath, and T. Melodia, "Redefining wireless communication for 6G: Signal processing meets deep learning with deep unfolding," 2020, *arXiv:2004.10715*.
[36] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019.
[37] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.
[38] M.-W. Un, M. Shao, W.-K. Ma, and P. C. Ching, "Deep MIMO detection using ADMM unfolding," in *Proc. IEEE Data Sci. Workshop (DSW)*, Jun. 2019, pp. 333–337.
[39] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.
[40] J. Zhang, Y. Li, Z. L. Yu, Z. Gu, Y. Cheng, and H. Gong, "Deep unfolding with weighted ℓ_2 minimization for compressive sensing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 3027–3041, Feb. 2021.
[41] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2019, pp. 266–271.
[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[45] Y.-C. Liang, Y. Zeng, E. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1326–1337, Apr. 2018.
[46] *5G Toolbox*. Accessed: Sep. 2021. [Online]. Available: https://www.mathworks.com/products/5g.html
[47] T. J. O'shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. GNU Radio Conf.*, 2016, vol. 1, no. 1.
[48] *Satellite Communications Toolbox*. Accessed: Sep. 2021. [Online]. Available: https://in.mathworks.com/products/satellite-communications.html
[49] M. Caromi and M. R. Souryal. *Simulated Radar Waveform and RF Dataset Generator for Incumbent Signals in the 3.5 GHz CBRS Band*. Accessed: Sep. 2021. [Online]. Available: https://data.nist.gov/od/id/mds2-2229
[50] M. Abadi. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/
[51] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.



BRIJESH SONI received the B.E. degree in electronics engineering and the M.E. degree in communication systems engineering from Gujarat Technological University, in 2014 and 2017, respectively, and the Ph.D. degree from the School of Engineering and Applied Science, Ahmedabad University, in July 2021. He is currently a Postdoctoral Researcher with the Boston College, MA, USA. Prior to that, he was an Adjunct Faculty at Ahmedabad University. He was also associated as a Junior Research Fellow at Ahmedabad University for the Department of Science and Technology (DST)—U.K. India Education and Research Initiative (UKIERI) Research Project jointly funded by DST and British Council, U.K. His research interests include cognitive radio networks, applied machine learning/deep learning for xG wireless networks, NOMA, and physical layer security.



DHAVAL K. PATEL (Member, IEEE) received the B.E. and M.E. degrees (Hons.) in communication systems engineering from Gujarat University, Ahmedabad, India, in 2003 and 2010, respectively, and the Ph.D. degree in electronics and communications from the Institute of Technology, Nirma University, Ahmedabad, in 2014. He was a Visiting Faculty at the Franklin W. Olin College of Engineering-Massachusetts, Needham, MA, USA. From 2011 to 2014, he was a Junior Research Fellow at the Post Graduate Laboratory for Communication Systems, Nirma University. Since 2014, he has been with the School of Engineering and Applied Science, Ahmedabad University, India, where he is currently an Associate Professor. His research interests include vehicular cyber physical systems, 5G wireless networks, non-parametric statistics, and physical layer security. He is the Principal Investigator of Research Projects funded by the Department of Science and Technology, U.K. India Education and Research Initiative (UKIERI), Association of Southeast Asian Nations (ASEAN)—India Collaborative Research and Development Project, and the Gujarat Council on Science and Technology. He is a Reviewer of various conferences, including IEEE ICASSP, IEEE VTC, and IEEE PIMRC.



SANKET B. SHAH received the degree from the School of Engineering and Applied Sciences, Ahmedabad University, India. He is currently pursuing the degree with San Francisco State University, USA. As an Undergraduate Research Fellow and Research Intern at Ahmedabad University, he was worked on applications of machine learning and artificial intelligence that involve modules in cognitive radio, adversarial media defense, and image classification. His research interests include computer vision, architecture search, and automation in model selection.



MIGUEL LÓPEZ-BENÍTEZ (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees (Hons.) in telecommunication engineering from Miguel Hernández University, Elche, Spain, in 2003 and 2006, respectively, and the Ph.D. degree *summa cum laude* in telecommunication engineering from the Technical University of Catalonia, Barcelona, Spain, in 2011.

From 2011 to 2013, he was a Research Fellow at the Centre for Communication Systems Research, University of Surrey, U.K. In 2013, he became a Lecturer (Assistant Professor) at the Department of Electrical Engineering and Electronics, University of Liverpool, U.K., and was promoted to a Senior Lecturer (Associate Professor), since 2018. His research interests include wireless communications and networking, with special emphasis on mobile communications and dynamic spectrum access in cognitive radio systems. He is/has been the Principal Investigator or a Co-Investigator of research projects funded by the EPSRC, British Council and Royal Society, and has been involved in the European-funded projects AROMA, NEWCOM++, FARAMIR, QoSMOS, and CoRaSat. He has been a member of the Organising Committee for the IEEE WCNC International Workshop on Smart Spectrum (IWSS 2015–2020). He is an Associate Editor of IEEE Access, *IET Communications*, and *Wireless Communications and Mobile Computing*. Please visit http://www.lopezbenitez.es for details.



SIDDHARTAN GOVINDASAMY (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), in 1999, 2000, and 2008, respectively. At MIT, he worked on a master's thesis on speech enhancement algorithms in partnership with Qualcomm Inc.; and the Ph.D. thesis on *ad-hoc* wireless networks with multi-antenna devices. From 2000 to 2003, he was a DSP Engineer and later a Senior DSP Engineer at Aware Inc., where he worked on developing broadband modem technology. From 2008 to 2020, he was an Assistant Professor and later an Associate Professor of electrical and computer engineering at the Olin College of Engineering, Needham, MA, USA, where he conducted research on large multiple-input multiple-output (MIMO) systems and optical wireless communications. He joined the Department of Engineering, Boston College, as a Founding Faculty Member, in Fall 2020, where he is a Professor of engineering. He is the coauthor of the textbook *Adaptive Wireless Communications: MIMO Channels and Networks* (Cambridge University Press, 2013). His research interests include in large MIMO systems and signal detection and processing.