View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Liverpool Repository

Noname manuscript No.
(will be inserted by the editor)

Artificial Neural Network Design for Improved Spectrum Sensing in Cognitive Radio

Dhaval K. Patel · Miguel López-Benítez · Brijesh Soni · Ángel F. García-Fernández

Received: December 21, 2019, Revised April 27, 2020 and June 04, 2020. Accepted July 03, 2020

Abstract Dynamic Spectrum Access/Cognitive Radio systems access the channel in an opportunistic, non-interfering manner with the primary network. These systems utilize spectrum sensing techniques to sense the occupancy of the primary user. In this paper, an artificial neural network based hybrid spectrum sensing technique is proposed, which considers sensing as a binary classification problem to detect whether the primary user is idle or busy. The proposed scheme utilizes energy detection and likelihood ratio test statistic as features to train the neural network. Moreover, we demonstrate the impact of hyperparameter tuning and carry out the detailed study of it, yielding a combination of best-suited hyperparameters. The performance of the proposed sensing scheme is validated on primary signals of various real world radio technologies acquired with an empirical testbed setup. We conclude that the best performing configuration results in an increase of approximately 63% in detection performance compared to classical energy detection and improved energy detection sensing schemes when averaged over all the radio technologies considered in this work.

Keywords Artificial neural network, Hyperparameter tuning, Cognitive radio, Spectrum sensing.

Dhaval K. Patel, Brijesh Soni
School of Engineering and Applied Science, Ahmedabad University, India. E-mail: {dhaval.patel, brijesh.soni}@ahduni.edu.in

Miguel López-Benítez, Ángel F. García-Fernández
Department of Electrical Engineering and Electronics, University of Liverpool, UK and ARIES Research Centre, Antonio de Nebrija University, Madrid, Spain. E-mail: {M.Lopez-Benitez,Angel.Garcia-Fernandez}@liverpool.ac.uk

1 Introduction

1.1 Background

With the expeditious advancement of wireless communication technologies and the advent of 5G massive multiple input multiple output (MIMO) systems, spectrum resources are becoming highly scarce [1]. As per the spectrum occupancy campaign in 2016, the overall usage of spectrum ranges from 7% to 34%, which demonstrates significant under-utilization of the spectrum resources [2]. Evidently, the conventional fixed spectrum allocation policy is not optimal. Hence, the spectrum allocation needs to be dynamic for efficient usage and opportunistic access of the spectrum band. Dynamic Spectrum Access/Cognitive Radio (DSA/CR) is envisaged as a promising solution to alleviate this existing conflict between increasing spectrum demand and spectrum under-utilization [3].

DSA/CR systems aims at increasing the efficiency of spectrum usage by allowing unlicensed or secondary users (SUs) to opportunistically access licensed spectrum bands temporarily unused by the licensed or primary users (PUs) in a non-interfering manner [4]. More specifically, CR exploits the parts of radio spectrum that are not occupied at some specific time instances in some specific locations and moves its operation to these parts called *spectrum holes or white spaces* for opportunistic access [5]. In order to opportunistically access a licensed spectrum band in a non-interfering manner, it is extremely important for DSA/CR systems to have the knowledge of PUs spectrum activity. Consequently, a number of different spectrum sensing schemes have been proposed in the literature [6]. Each sensing scheme provides a different trade-off between required sensing time as well as computational and memory complexities.

Spectrum sensing schemes can be broadly divided into two categories, namely parametric and non-parametric. Parametric sensing schemes require the DSA/CR system to have apriori information of the PU. Many spectrum sensing techniques, including matched filter detection, adaptive spectrum sensing and cyclostationary based sensing, have been elucidated in [7], [8]. On the contrary, non-parametric sensing schemes do not require any a priori information, which is usually the case in real-life scenarios. The channel state (busy/idle) is often uncertain and not known, hence, non-parametric schemes are preferred. One of the most simple and widely popular non-parametric sensing schemes is energy detection [9], [10], which compares the received signal energy with a predefined threshold and determines the state of the PU, either busy or idle. The implementation of energy detection is straightforward, but it is sensitive to signal and noise uncertainty [11], [12]. In other words, energy detection fails to perform in cases of highly variable signal power and/or noise power, thereby dramatically degrading the overall sensing performance. As an alternative, a new class of algorithms based on Goodness-of-Fit (GoF) tests has been proposed in the literature such as Anderson-Darling (AD) test [13], Ordered Statistics (OS) test [14], Kolmogorov-Smirnov (KS) test [15], [16], Likelihood Ratio Statistics (LRS) test [17], [18] and Improved Likelihood ratio statistic based test in our recent work [19]. Sensing schemes based on GoF test yield a reasonably better performance than other algorithms in low signal to noise ratio (SNR) conditions.

1.2 Related Works

In recent past, researchers from industry and academia have resorted to the use of machine learning and deep learning algorithm for various fields including computer vision, natural language processing, medical science and wireless networks among many others [20]. Owing to the excellent learning ability using data driven approach and with the rapid advancement in the learning based signal processing techniques [21], machine learning and deep learning algorithms have gained wide attention from industry and academia in the context of future wireless networks [20, 22–24]. The key advantage of CR network is its cognitive ability, i.e., learning by itself from the radio environment. This is analogous to the machine learning/deep learning framework. In similar lines, authors in the recent work [25] proposed a spectrum intelligent radio which hierarchically performs perception, understanding and reasoning. Thus, these frameworks have been applied to CR networks as well [26–29].

There are many works in the literature that have applied machine learning approach for spectrum sensing in CR networks. For instance, support vector machine (SVM) classifier based spectrum sensing and real-time detection was proposed in [30]. Also, the SVM - Radial Basis Function (RBF) based signal classification and spectrum occupancy was carried out in [31]. The Naive Bayesian classifier (NBC) based multi-class classification for spectrum sensing was proposed in [32], while the authors in [33] proposed the random forest classifier based approach to decrease the interference of the unlicensed user to the licensed users in CR network, thus improving the network throughput dramatically. The authors in [34] studied the unsupervised learning approach for signal classification using various statistical features as training parameters. The similar unsupervised learning approach namely, k-means clustering, was applied for cooperative spectrum sensing in [35]. Furthermore, the work in [36] proposed the four machine learning classifiers: K-nearest neighbor (KNN), SVM, NBC, and Decision Tree in classifying the state of PUs in cooperative spectrum sensing.

Moreover, many works have also reported the shallow and deep multilayer perceptron network aided spectrum sensing techniques. For instance, authors in [37] proposed a sensing scheme that uses the energy and cyclostationary features to train the neural network for spectrum sensing. In our previous work [38], we used energy and Zhang statistics as a training features for the ANN. Furthermore, the ANN based multi-slot spectrum prediction and an adaptive mode selection in full duplex CR network was proposed in [39]. The comprehensive study of machine learning based spectrum sensing in CR can be found in the recent work [40].

In addition, recently there are few works that have applied deep learning for spectrum sensing. For instance, spectrum sensing using convolutional neural network (CNN) was proposed in [41–43] and CNN for cooperative spectrum sensing in [44]. Moreover, various other deep learning architectures have also been recently implemented for spectrum sensing like stacked autoencoder based sensing in [45], variational autoencoder approach aided unsupervised deep learning for spectrum sensing in [46], spectrum prediction based on Taguchi method in deep learning with long short term memory (LSTM) was carried out in [47]. Moreover, our recent work in [48] focused on LSTM based spectrum sensing scheme, while the work in [49] focused on utilising the PU activity statistics alongwith the LSTM based sensing prediction to improve the detection performance. The summary of the related works is provided in Table 1.

Table 1: Summary of the related works

Ref.	**Brief Description**	**ML/DL framework**	**Classification /Prediction**	**Performance metric**	**Input to the ML/DL framework**	**Empirical Test-bed setup**	**Impact of hyper-parameter Tuning**
[30]	Support Vector Machine based spectrum sensing	ML (SVM)	Classification	Detection probability	Energy detection based binary value	×	×
[31]	Support Vector Machine - Radial basis function based spectrum sensing and spectrum occupancy	ML (SVM)	Classification	Classification accuracy	Power values (in dBm)	✓	×
[32]	Naive Bayesian based multi-class classification for OFDM systems	ML (Naive Bayes)	Classification	Detection probability	F_1 and F_2	×	×
[33]	Random forest based algorithm to decrease the interference of the unlicensed user in CR network	ML (Random forest)	Classification	Detection probability and accuracy	Energy statistics	×	×
[34]	Robust unsupervised sensing scheme and developed countermeasures to the class manipulation attacks.	ML (K-means)	Classification	Misclassification rate	time series autocorrelation and higher moments of the data	×	×
[35]	K-means clustering aided cooperative spectrum sensing	ML (K-means)	Classification	Detection probability	Energy statistics and no. of clusters	×	×
[36]	Analysis of ML classifiers in cooperative spectrum sensing	ML (KNN, SVM, NBC, and Decision Tree)	Classification	Throughput and arrival rate	Energy statistics	×	×
[37]	Cyclostationary feature detection aided ANN based spectrum sensing	ML (ANN)	Classification	Detection probability	Cyclostationary feature	×	×
[38]	ANN based spectrum sensing using ED and LRS as features	ML (ANN)	Classification	Detection probability	Energy and Zhang Statistic	✓	×
[39]	ANN based multi-slot spectrum prediction and an adaptive mode selection in full duplex CR network	ML (ANN)	Prediction	Prediction error probability and Throughput	Energy detection based binary value	×	×
[41]	DNN featured likelihood ratio test aided CNN based spectrum sensing	DL (CNN)	Classification	Detection probability	Covariance matrix	×	×
[42]	Covariance matrix aware CNN based spectrum sensing	DL (CNN)	Classification	Detection probability	Covariance matrix	×	×
[43]	Primary user activity pattern aware spectrum sensing	DL (CNN)	Classification	Detection probability	Covariance matrix	×	×
[44]	CNN based co-operative spectrum sensing scheme	DL (CNN)	Classification	Detection probability	Matrix containing individual SU sensing results	×	×
[45]	Stacked Autoencoder based spectrum sensing for OFDM signals	DL (SAE)	Classification	Detection probability	Raw signal data	×	×
[46]	Variational autoencoder approach aided unsupervised deep learning for spectrum sensing	DL (VAE)	Classification	Detection probability	Raw signal data	×	×
[47]	Spectrum prediction based on Taguchi method in deep learning with long short term memory	DL (LSTM)	Prediction	Classification accuracy and MMSE	Raw signal data	✓	✓
[48]	Long Short Term Memory aided spectrum sensing.	DL (LSTM)	Classification	Detection probability	Raw signal data	✓	×
[49]	Long Short Term Memory aided spectrum sensing using primary channel activity statistics	ML (ANN) and DL (LSTM)	Classification and Prediction		Raw signal data + extracted PU statistics	✓	×
Our work	Hyper-parameter tuning based ANN based spectrum sensing	ML (ANN)	Classification	Detection probability	Energy and Zhang Statistic	✓	✓

1.3 Motivation and Contributions of this work

The design of accurate and robust machine learning/ deep learning algorithms relies on the hyperparameters used. The tuning of the hyperparameters plays a vital role on the performance of machine learning and deep learning algorithms [50]. Selecting the hyperparameters without adequate tuning may result in an over optimistic performance or poor performance of the designed algorithms. Although the work in [47] has proposed the Taguchi method based hyperparameter tuning in LSTM, the performance metric evaluated was classification accuracy. On the contrary, this work considers the detection probability as a key performance metric and have demonstrated the effect of SNR for different values of false alarm probability. These performance metrics are the key players while dealing with the design and dimensioning of CR networks. In this work, we carry out the detailed analysis and impact of tuning various hyperparameters on the spectrum sensing performance, which is not yet explicitly reported in literature.

This paper focuses on the selection and tuning of hyperparameters: 1) optimization algorithms, 2) activation functions, and 3) learning rate. Firstly, optimization algorithms such as Adams [51], Nesterov [52] and Ada Boost [53] have been used to update the weights of the neural network such that the loss function of the ANN is minimized. Secondly, activation functions like sigmoid, hyperbolic tangent (tanh) and rectified linear unit (ReLU) have been considered and their performance on the captured data has been studied. Thirdly, the learning rate of the ANN is a key parameter, which is decided upon the possibility and latency of convergence of forward and backward propagation algorithms. In order to validate the results of the proposed scheme, we evaluate it on various real-world primary signals captured with an experimental test-bed setup. In addition, we compare the proposed approach with the classical energy detection (CED) commonly used in the literature and the improved energy detection (IED) algorithm proposed in [54] and also with NBC based sensing in [32]. The simulation results demonstrate that the tuning of hyperparameter significantly impacts the performance of the proposed sensing scheme.

The main contributions of this work can be outlined as follows:

1. We thoroughly analyse and study the selection of hyperparameters in an ANN for CR network to obtain a better classification accuracy of the ANN, resulting in a higher detection performance at low SNR regime.
2. A comprehensive study of the impact of lower and higher SNR regime on the performance of ANN is carried out. Experimental results demonstrate the superior performance of ANN as compared to existing CED and IED algorithms across lower and higher SNR regimes.
3. The tuned hyperparameters of ANN like activation function, optimization algorithms and learning rate are validated based on empirical data-sets of various radio technologies. Further, the impact of the number of epochs for which the neural network is trained on the loss function for different optimizers is studied.

   Experimental results prove that by finding the optimal hyperparameters, the proposed ANN framework for spectrum sensing outperforms CED and IED algorithms.

1.4 Paper Organization

The rest of this paper is organized as follows. Section 2 describes the system model and formulates the problem. Section 3 elucidates the proposed hyperparameter tuning aided ANN spectrum sensing scheme. Further, Section 4 discusses the empirical setup and data capturing process employed in this work. Section 5 describes the experimental results. Finally, the conclusions are drawn in Section 6.

2 System Model and Problem Formulation

2.1 Cognitive Radio Network and Signal Model

In this work the spectrum sensing performance is evaluated considering a link level scenario with a single channel, which is the usual approach widely adopted in the literature for the performance evaluation of spectrum sensing methods (see [55] and references therein). The multi-channel scenario is relevant from a network level point of view, however as far as spectrum sensing is concerned, this scenario can simply be seen as an extension of the single channel scenario where each channel, even though is physically different in terms of carrier frequency, is statistically equivalent. Notice that the consideration of multiple channel scenario leads to the introduction of other problems that are beyond the scope of this work (e.g., MAC scheduling of the channels to be sensed), but in general does not affect the per-channel performance of the evaluated spectrum sensing methods in this paper.

A typical architecture of neural networks with one hidden layer and single output is shown in Fig. 1. We assume that a SU detect only a single PU channel activity signal in a CR network using the subsequently proposed spectrum sensing algorithm. Here, the SU is required to sense the PU's channel periodically to determine if the channel is idle or busy. We consider a sequence of measurements of captured signals $y_1, y_2, \ldots, y_i, \ldots, y_k, \ldots, y_K$ where i is any arbitrary timestamp, k is the current timestamp, K is the last timestamp. Each of these measurements is modelled as:

$$y_i = \begin{cases} n_i, & H_{i,0} \text{ (PU is absent)}, \\ hx_i + n_i, & H_{i,1} \text{ (PU is present)}, \end{cases} \tag{1}$$

where h denotes the channel response coefficient between PU and SU, x_i denotes the transmitted signal at i^{th} timestamp, n_i denotes the noise at i^{th} timestamp modelled as additive white Gaussian noise (AWGN) with zero mean and variance σ^2. Also, hypothesis $H_{i,0}$ represents that the PU is absent and spectrum opportunity is available for SU at i^{th} timestamp whereas hypothesis $H_{i,1}$ represents that the PU is present and spectrum opportunity is unavailable for SU at i^{th} timestamp. Based on the captured observations y_1, y_2, ..., y_i,..., y_k, ..., y_K, the prime objective is to predict whether the channel will be idle or busy in the next i.e. $(k+1)^{th}$ timestamp. Mathematically, the aim is to calculate the following conditional probability:

$$Pr(H_{k+1,j} | y_1, y_2, \ldots, y_k), \tag{2}$$

where $j \in \{0,1\}$. Note that $j = 0$ indicates the null hypothesis while $j = 1$ indicates the alternate hypothesis.

There are numerous mathematical models in order to estimate or calculate this probability. However, an ANN based approach has been undertaken in this work considering the unique ability to fit complex non-linear functions. Further, we extract four features of the received signal as described in Section 3, represented as X_k, to be provided as inputs to the ANN. Therefore, the neural network is used to learn the mapping of:

$$Pr(H_{k+1,j} | X_k), j \in \{0, 1\} \tag{3}$$

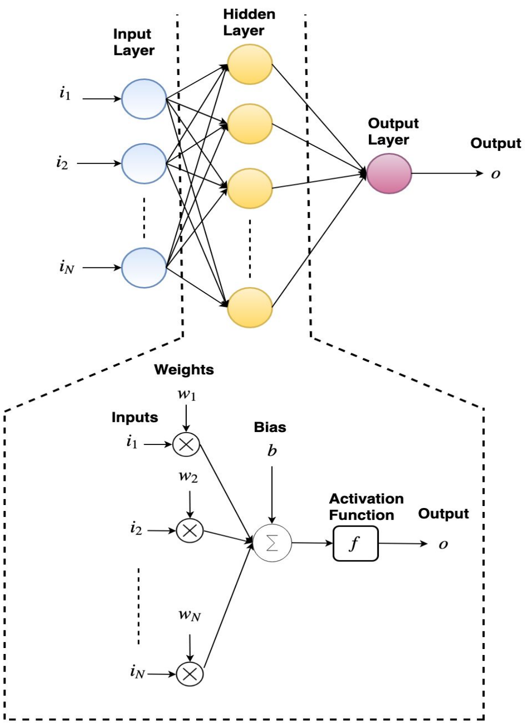


Fig. 1: Architecture of ANN with Input, Hidden and Output layers. The bottom part shows the corresponding weights and biases followed by the activation function.

2.2 Preliminaries of ANN Cost Function formulation

In order to invoke a probabilistic framework, we need to map the output of the ANN between 0 and 1. To produce an output between 0 and 1, we use certain mathematical functions that map the input features X within the desired interval. To this end, $\forall a \in \mathbb{R}$, we consider the sigmoid function, which is defined as follows [50]:

$$f(a) = \frac{1}{1+e^{-a}} \tag{4}$$

Let $\hat{y}_i$ represent the predicted received signal status (idle/busy). We consider $\hat{y}_i = 1$ to be the decision if the output of the considered ANN is greater than 0.5, otherwise $\hat{y}_i = 0$. The output of the ANN with no hidden layers is as follows:

$$\hat{y}_i = f\Big(\sum_n w_n X\Big), \tag{5}$$

where w_n is the weight associated with n^{th} neuron of total N neurons and X is its input feature vector. Therefore, the predicted sigmoid output is interpreted as:

$$\hat{y}_i = Pr(H_{i,1}|X), \quad \text{and} \tag{6}$$

$$1 - \hat{y}_i = Pr(H_{i,0}|X). \tag{7}$$

Since the target output is binary, it is reasonable to model the randomness as Bernoulli trials. For $j \in \{0, 1\}$,

$$Pr(H_{i,j}|X, w) = \hat{y}_i^{H_{i,j}} (1 - \hat{y}_i)^{1-H_{i,j}} \tag{8}$$

Solving (8) further, we take the natural logarithm on both sides of the equation to simplify the analysis and additionally, the min/max of the log likelihood is the same as the min/max of the likelihood,

$$\ln(Pr(H_{i,j}|X, w)) = H_{i,j} \ln(\hat{y}_i) + (1 - H_{i,j}) \ln(1 - \hat{y}_i) \tag{9}$$

In order to maximize this probability, we need to maximize the log likelihood. This can be easily done by converting it to a minimization problem by multiplying the log likelihood with a negative sign. Hence, we obtain the cost function $J(w)$ for a single layered neural network at i^{th} timestamp with M samples of training data, which is as follows:

$$\begin{aligned} J(w) = -\Big(\frac{1}{M}\Big) \sum_{m=1}^{M} & H_{i,j\,(m)} \ln(\hat{y}_{i\,(m)}) \\ & + (1 - H_{i,j\,(m)}) \ln(1 - \hat{y}_{i\,(m)}) \end{aligned} \tag{10}$$

Furthermore, the cost function can be simply extended to L-layered neural network as follows:

$$\begin{aligned} J(w) = -\Big(\frac{1}{M}\Big) \sum_{m=1}^{M} \sum_{l=1}^{L} & H_{i,j\,(l,m)} \ln(\hat{y}_{i\,(l,m)}) \\ & + (1 - H_{i,j\,(l,m)}) \ln(1 - \hat{y}_{i\,(l,m)}) \end{aligned} \tag{11}$$

Hence, our objective is to find the weights of the ANN in such a way that $J(w)$ is minimized, and in order to optimize these weights, various algorithms like stochastic gradient descent, adam optimizer and among others can be used as discussed in the subsequent sections.

3 Proposed hyperparameter tuning aided ANN based spectrum sensing scheme

The most important characteristic of neural networks is that they are remarkable at learning non-linear functional mapping between input and output and as a result they can adapt to the non-linear characteristics of PU's signals. The proposed sensing scheme utilizes

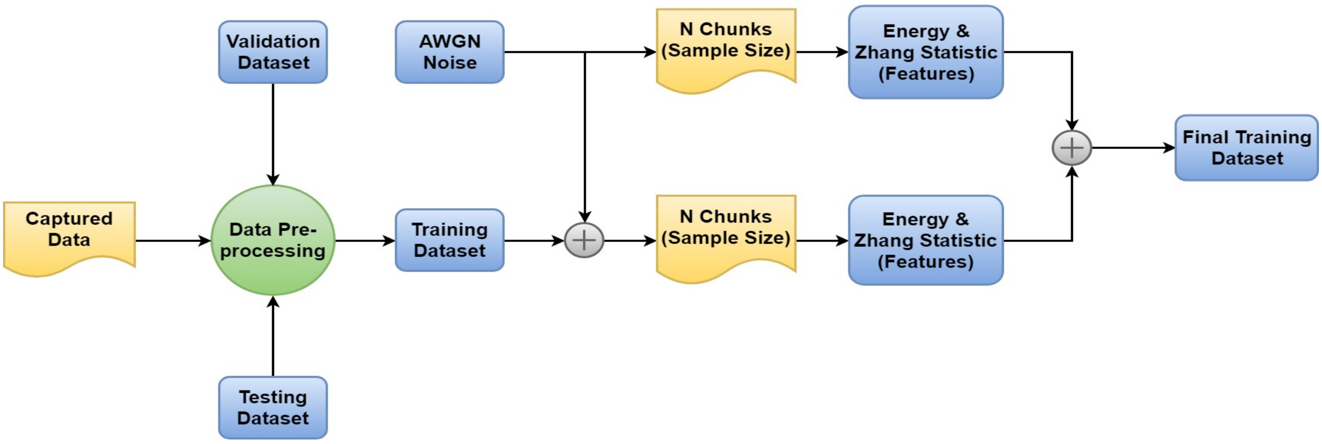


Fig. 2: The flowchart of the feature extraction process and obtaining final training dataset for the proposed scheme. The extracted features are Energy and Zhang statistics of the signal along with that of the AWGN noise.

a back propagation neural network (BPNN). The purpose of our proposed scheme is to determine/classify whether the PU channel is busy or idle. In this paper, we use a supervised learning setting, where the classifier is trained with features and corresponding labels. The ANN primarily incorporates classical energy detection and likelihood ratio statistics as its features. The labels in this classifier are 0 and 1 for the cases when the channel is idle and busy respectively. Denoting the discrete version of the received signal as y_i, the energy value E is given as:

$$E = \sum_{i=1}^{N} |y_i|^2 \tag{12}$$

Similarly, the Zhang statistic Z_c is given as:

$$Z_c = \sum_{i=1}^{N} \left[\log \left\{ \frac{F_0(y_i)^{-1} - 1}{\left(N - \frac{1}{2}\right) / \left(i - \frac{3}{4}\right) - 1} \right\} \right]^2, \tag{13}$$

where N is the sample size and $F_0(y)$ is the known cumulative distribution function (CDF) of noise.

The first step is to collect a data using an empirical setup as elucidated in Section 4. Thereafter, we extract four key features from the captured data. The first feature is the energy of the signal, denoted as $x_1(l)$, where l denotes the l^{th} training sample. The second feature is the energy value of the previous sensing event, denoted as $x_2(l)$. We consider this feature additionally in order to avoid a steep decrease in energy values caused due to small errors by empirical setup leading to miss-classification. Similarly, the third and fourth features are the Zhang statistics of the current and previous sensing events denoted by $x_3(l)$ and $x_4(l)$ respectively. Hence, the training vector, X can be denoted as: $X = [x_1(l); x_2(l); x_3(l); x_4(l)]$. Likewise, the labels for the l^{th} training sample are denoted by $Y = [A(l)]$. Here, l is the index number of the feature vector, different from k which is the raw index number. Here, we use various optimization techniques for ANN in sensing with different hyperparameters, which eventually can be used to improve the performance in terms of detection probability at low SNR conditions as well.

Once all the models with different hyperparameters are trained, the algorithm moves into the validation mode. Validation is an important step in terms of machine learning, since there may be a case where the classifier remembers all the features from the training data-set and is able to perform exceptionally on training data-set, but then fails when deployed practically and exposed to real-life systems. A patent reason for such cases is that the ANN has remembered the training data and has not learned or understood it [50]. This issue is known as overfitting of data. To avoid this problem, data-sets are usually divided into three different sets: training, validation and testing.

Validation data-set uses all the models that were trained using training data-set to evaluate the performance of the algorithm. Also, an indicator function is used to check how many times the classifier wrongly predicts the labels. Further, the model which yields the lowest value of the indicator function is chosen. The classifier that satisfies the above condition is passed to the testing phase. Upon reception of the signal, the feature set is calculated and provided as input to the neural network, which then decides the state of the PU channel (idle/busy).

Algorithm 1 The proposed ANN design scheme

```
Subroutine for feature extraction
Input: N, Label, Data
Output: Features
size = len(Data) / N
Samples $1^{st}$ N Samples
E = Energy (Samples) {Compute energy value}
Z = Zhang statistic (Samples) {Compute Zhang statistic}

for $j \leftarrow 2$ to size do
    Samples = $(j)^{th}$ N Samples {read the signal with window $N$}
    E_P = E{Energy of previous sensing event}
    Z_P = Z{Zhang statistic of previous sensing event}
    E = Energy (Samples)
    Z = Zhang statistic (Samples)
    Feature[j-1] = {E, E_P, Z, Z_P, Label}
end for

Subroutine for ANN Training
Input: N, Epoch, Batch_size, Label, $\alpha$, Data
Output: ANN Network
ANN Network = Construct Network Layers ()
Network_weights & Bias = Initialize (ANN Network)
Feature = FeatureExtraction(N,Label,Data)

for $i \leftarrow 1$ to $Epoch$ do
    T_F & label = Extract (Feature, Batch_size)
    output = Forward Propagate (T_F, ANN Network)
    Error = BackwardPropagateError (label, output)
    Weights = UpdateWeights(label,output,ANN Network,$\alpha$)
end for

Subroutine for ANN Testing
Input: ANN Model, N, No. of sensing events
Output:S
weights & bias = ANN Model

    for $i \leftarrow 1$ to No. sensing event do
        E_P = Energy of $(i-1)^{th}$ N Samples
        Z_P = Zhang statistic of $(i-1)^{th}$ N samples
        E = Energy of $(i)^{th}$ N Samples
        Z = Zhang statistic of $(i)^{th}$ N samples
        Features = {E, E_P, Z, Z_P}
    end for
output = ANN Predict (Features, weights, bias)

    if output == 1 then
        S$_i$ = $\mathcal{H}_A$

    else
        S$_i$ = $\mathcal{H}_0$
end if
```

3.1 Feature Extraction

Firstly, data pre-processing steps are applied on the captured data in which the data-set is divided into three categories, training, validation and testing as discussed earlier. AWGN is then added to the generated training data and samples of size N are extracted to calculate features like the energy and Zhang statistics. Similarly, feature samples of size N are also extracted for the case where only noise is generated. Both components are then added together and each row in the data-set then behaves as detection in one instance. The labels are added to the data-set, where signal plus noise is denoted by label 1, implying that the PU channel is busy. Similarly, only noise is denoted by label 0, implying that the PU channel is idle. Feature extraction pseudo code is formulated in Algorithm 1 and a summary of the overall feature extraction procedure is illustrated in Fig. 2.

3.2 ANN Training

The prime objective of ANN training is to minimize the cost function formulated in (11). Training of any ANN model iteratively involves two steps, namely forward propagation and backward propagation, which are discussed below:

3.2.1 Forward Propagation

The feature vector is fed as input to the ANN and each neuron is assigned random values of weight. In the forward pass, the neuron weights are multiplied with feature values and added to a bias value which is eventually passed as an input to an activation function. For an ANN with n_H hidden layers, mathematically,

$$\boldsymbol{net_j} = \sum_{i=1}^{p} x_i w_{ji} + w_{j0}, \tag{14}$$

where x_i is the i^{th} feature value, w_{ji} is the i^{th} weight of j^{th} hidden layer, w_{j0} is the bias value and net_j is the net activation of j^{th} hidden layer. The output of the j^{th} hidden layer can be obtained by:

$$\boldsymbol{y_j} = f(\boldsymbol{net_j}), \tag{15}$$

where f denotes the activation function. Similarly, we evaluate the net activation and output for the output layer of the ANN:

$$\boldsymbol{net_k} = \sum_{j=1}^{n_H} y_j w_{kj} + w_{k0}, \tag{16}$$

$$\boldsymbol{z_k} = f(\boldsymbol{net_k}) \tag{17}$$

Here, the activation function maps input $\mathbb{R} \rightarrow [0, 1]$. This is one of the hyperparameters which can be optimized to obtain better results. After evaluating various activation functions on real world signals, the best activation function is used for testing the ANN model.

3.2.2 Backpropagation

This calculates the gradient of the cost function. Therefore, the error in output to input layer is calculated using the chain rule of differentiation as follows:

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial z_k}\frac{\partial z_k}{\partial net_k}\frac{\partial net_k}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} \tag{18}$$

The weights of the neural network are updated using these gradients by moving into the direction of steepest slope, minimizing the cost function. To update the weights, there are various optimization techniques such as Adam optimizer, momentum based gradient descent, etc. A summary of the ANN training module and procedural psuedo code is given in Algorithm 1. The various hyperparameters, the best of which is used in testing the ANN model after evaluation of its performance on various real life signals is described in the section below.

3.3 ANN Hyperparameter selection

After the different ANN architectures are trained with appropriate training data-set and epochs, we identify the ANN architecture that best fits based on its performance. The validation data-set is used to identify the ANN model with different hyperparameters that performs best on validation data-set. The validation data-set is generated using the signal data and is processed in a way similar to training data. The combined feature data is then passed to all the stored models of the ANN with different hyperparameters. The ability of different classifiers to detect correctly is identified and then passed to the selection stage. Here, the classifier that yields the maximum accuracy is selected and used for testing the proposed ANN scheme.

3.3.1 Hyperparameter-I : Number of hidden layers and number of nodes

Number of hidden layers and number of neurons in each layer is an important hyperparameter that needs to be analyzed and tuned to improve the prediction accuracy of ANN. A few set of combination of values were tested as mentioned in Table 2 below and the accuracy of the model was obtained. The accuracy is calculated as the number of correctly classified labels by the built ANN. It is worth noting that for smaller number of layers, the validation accuracy remains almost constant.

Table 2: Different ANN architectures with their accuracy on cross validation data-set

No. of Hidden Layers	Nodes in each layer	Achieved Accuracy
1	2	85.45%
1	7	87.79%
1	20	87.77%
2	5	87.85%

3.3.2 Hyperparameter-II : Optimization Techniques

In this section, we discuss the optimization techniques used to obtain the improved results for each radio technology.

1. **Stochastic Gradient Descent (SGD)**: This algorithm updates the parameters w of the cost function $J(w)$ as:

$$w := w - \eta \cdot \nabla_w J(w) \tag{19}$$

   The learning rate η remains constant for all the updates. Also, the learning rate for SGD is typically smaller as compared to batch gradient descent in order to avoid overshooting the convergence point.
2. **AdaGrad**: This algorithm is a sound method for an efficient adaptation of the learning rate [53]. It maintains the learning rate per parameter as opposed to having the same learning rate for all the weights. One of the drawbacks of AdaGrad is that once it reaches closer to minima, the learning rate becomes significantly smaller, hence it cannot reach the exact minima even after numerous iterations.
3. **Adam**: This optimization algorithm can be used instead of classical SGD to update network weights iteratively based on training data. The name is derived from "Adaptive Moment Estimation" (ADAM) [51]. Unlike the classical SGD which maintains a single learning rate for all the weight updates, a learning rate is maintained for each network weight (parameter) and separately adapted as learning unfolds. Adam realizes the benefits of both AdaGrad and RMSProp (another optimization algorithm), making use of first (mean) and second (uncentered variance) moments of gradients.
4. **Nesterov Accelerated Gradient**: This optimization algorithm is a momentum based gradient descent technique. Its goal is to tune smaller updates when closer to minima and tune larger updates when away from minima [52].

ADAM optimizer is generally used for achieving faster convergence. However, the best performing optimizer is subject to the data-set on which it is trained and being tested. In the case of spectrum sensing, it is observed that Nesterov accelerated gradient optimization technique yields minimum model loss due to its ability to gather momentum and avoid convergence on other local minima. The plot of loss v/s epoch for various optimizers over different radio technology is explained in Fig. 9 of the numerical results section.

3.3.3 Hyperparameter-III : Activation Functions

Another important hyperparameter to be tuned optimally is the choice of activation function. For that, we consider the following activation functions to analyze the performance of the proposed ANN architecture:

1. Sigmoid activation function:

$$f(a) = \frac{1}{1 + e^{-a}}, \tag{20}$$

2. ReLU activation function:

$$f(a) = max(0, a), \tag{21}$$

3. Tanh activation function:

$$f(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}, \tag{22}$$

Sigmoid and Tanh usually are not used due to the vanishing gradient problem, which eventually degrades the overall accuracy. However, for the case of spectrum sensing, since the network used here is not very deep, all the activation functions yield nearly similar accuracy as discussed in Table 7 of numerical results section.

3.4 ANN Testing

After having trained and validated the proposed ANN framework, we move to the final phase to evaluate the sensing scheme. Fig. 3 illustrates the entire testing process. In order to compute the detection probability P_d, the testing data-set is added with AWGN to obtain signals with the desired SNR. Further, the data are then divided into small chunks (e.g., $N = 100$ samples) and features like the energy and Zhang test statistics are extracted. These features are fed as inputs to the trained neural network. We compute P_d for a given SNR by dividing the number of times that the neural network predicts that the channel is busy $(H_{k,1})$ by the total number of samples. Similarly, we compute the false alarm probability P_f, by using only AWGN sequences for feature extraction instead of test data. Algorithm 1 enumerates the final sensing steps where the ANN takes the decision about the existence of a primary signal. Fig. 4 shows the flowchart of the proposed scheme.

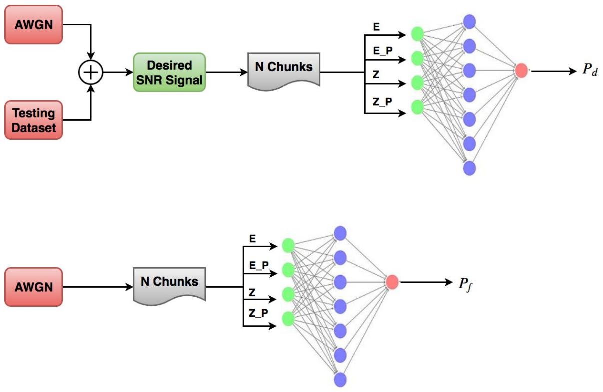


Fig. 3: ANN Testing: Computation of P_d and P_f for the proposed scheme. The number of times network correctly the PU signal to the total examples fed gives P_d. Similarly, number of times it does not predict H_0 divided by the total AWGN examples fed gives P_f .

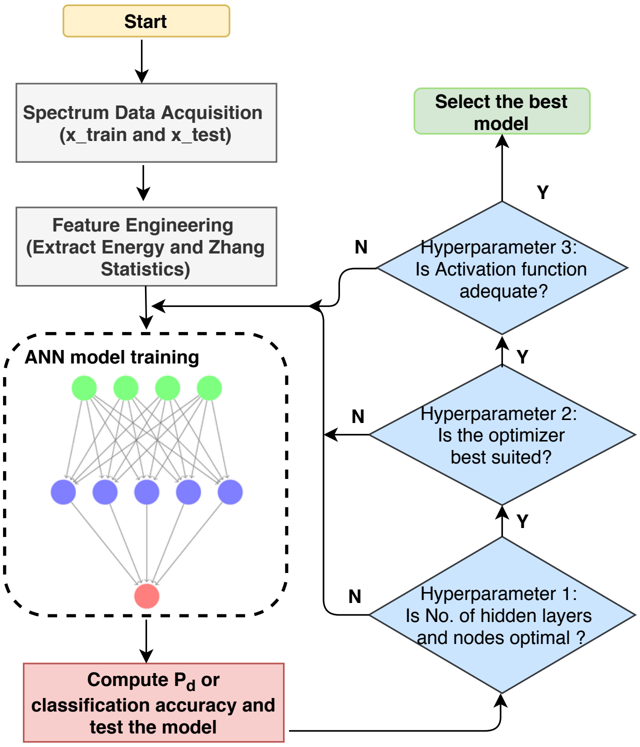

Start
Spectrum Data Acquisition (x_train and x_test)
Feature Engineering (Extract Energy and Zhang Statistics)
ANN model training
Compute P_d or classification accuracy and test the model
Select the best model
Y
N
Hyperparameter 3: Is Activation function adequate?
Y
N
Hyperparameter 2: Is the optimizer best suited?
Y
N
Hyperparameter 1: Is No. of hidden layers and nodes optimal ?

Fig. 4: Flowchart of the proposed scheme. After training the model, the detection probability is computed. The Algorithm is then iterated for various hyperparameters, until the best model is obtained.

4 Empirical Setup

4.1 Measurement Platform

A combination of both hardware and software components has been employed in this study, similar to [56]. The hardware components used for this study are the Universal Software Radio Peripheral (USRP - N210)

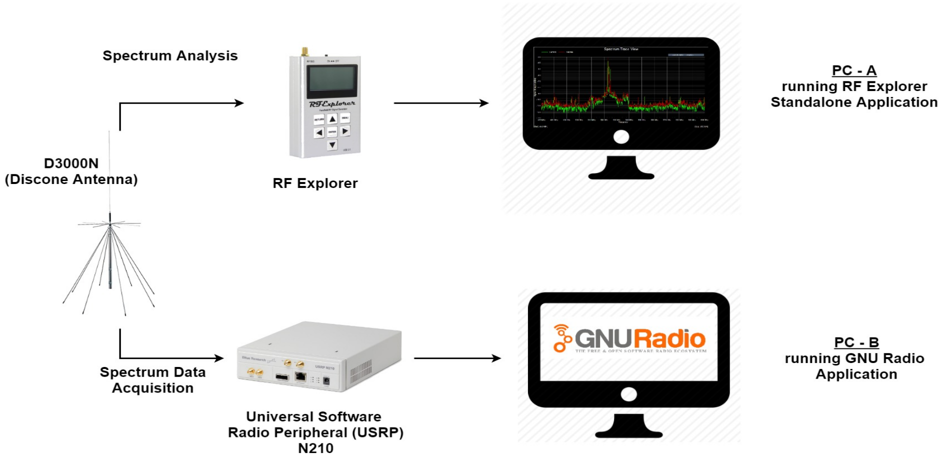


Fig. 5: Empirical test-bed measurement setup for spectrum data acquisition. The upper part of the figure shows the spectrum analysis through which the busy channel is identified while the bottom part shows the data acquisition using USRP N210 in the identified busy channel.

Table 3: Channels measured in this study

Radio Technology	Channel Number	F_{start} (MHz)	F_{center} (MHz)	F_{stop} (MHz)	Signal Bandwidth (MHz)	Gain (dB)	Decimation Rate (M)	Sampled Bandwidth (MHz)
FM Broadcasting		94.1 96.5 98.3	94.3 96.7 98.3	94.5 96.9 98.5	0.2	45	64	1
E-GSM (900) DL	27 77 120	940.2 950.2 958.8	940.4 950.4 959	940.6 950.6 959.2	0.2	45	64	1
DCS (1800) DL	527 690	1808 1839.6	1808.2 1840.8	1808.4 1841	0.2	45	64	1
UHF Television	U - 29 U - 33	534 566	538 570	542 574	8	45	8	8

[57], WBX daughter board, RF Explorer and a D3000N super discone antenna. All these components are required to capture the PU data. The software used includes GNU Radio [58] and MATLAB. The host PC runs the GNU Radio's script to collect the digital samples from USRP. Once the data are captured, off-line processing is performed in MATLAB. Further, the proposed scheme is employed on the stored data to evaluate the detection and false alarm probabilities.

4.2 Data Capturing and Pre-processing

The empirical setup was placed on the roof of the School of Engineering and Applied Science (SEAS), Ahmedabad University in order to have direct line of sight from several transmitters with minimum shadowing and reflection loss. A block diagram of the experimental setup for data collection is shown in Fig. 5. Data acquisition and spectrum measurement was conducted on Monday morning from 10:00 AM to 11:00 AM. These timings only affect the discontinuous transmitters. We used a Intel Core-i7 embedded processor Personal Computer (PC) system along-with MATLAB and GNU-Radio. Various environmental factors like temperature, humidity, moisture, dew, rain, etc., need to be considered as elaborated in [59]. However, as our measurement was of short duration we did not consider such factors into measurement account.

With the help of RF Explorer, certain channels with high SNR were identified for various radio technologies. As depicted from Table 3, radio technologies include FM broadcasting, GSM-900 DL, DCS-1800 DL, and UHF TV band. The channels with appropriate centre frequency, decimation rate and gain factor are selected. The gain factor is selected such that the highest SNR is achieved in the received signal, while the decimation rate ensures that the reception bandwidth is higher than or equal to the actual signal bandwidth. For each channel, around 10^7 samples were captured. A pre-processing step was applied to filter the signals and remove any transient peaks in the initial samples. The pre-processed data are then divided into three data-sets for all the radio technologies: training (60%), validation (20%) and testing (20%).

5 Experimental Results

The proposed ANN based spectrum sensing scheme was tested and evaluated on the empirical data. We used channels of four different radio technologies instead of using channels from only a single radio technology, because the overall performance for different channels of a single radio technology was found to be very similar to each other. Hence, as described earlier, four ANN architectures are used and are allotted a specific radio technology. Having obtained independent training and testing data for each radio technology, the detection probability P_d and probability of false alarm P_f are computed. One can use a single ANN also incorporating all the radio technologies together, but on doing so, one may miss the inherent effect of the technology dependency on P_d according to [56]. Moreover, a comparative study of the proposed scheme with existing CED and IED algorithms on captured data is presented for each radio technology and the results show that the proposed scheme outperforms the state of art schemes in all cases.

5.1 Validity of Captured Data

Even though it becomes difficult to eliminate the randomness factor and its repercussions from the empirical setup, the captured data have been used to reproduce the results of CED and IED algorithms given in [54]. In addition, in order to ensure the validity of the data-set used, the process of data capturing and sensing evaluation with the proposed scheme was repeated several times and the performance remained unchanged. Moreover, these analyses were conducted to ensure that no unexpected or incorrect trends that contradict fundamental theoretical expectations were present (e.g., increasing the SNR and/or the sample size N, P_d increases).

5.2 Analysis of ANN Training Module

5.2.1 Manipulation in training data

The proposed ANN based sensing scheme has a few critical requirements that need to be satisfied. Firstly, it requires proper training in order to accurately classify the hypotheses. Secondly, the ANN architecture requires to be as simple as possible so that it can be deployed on a real sensing framework. Keeping these requirements in mind, we manipulate the training data such that it changes the percentage of higher and lower SNR samples in training set, leaving the pure AWGN samples unaffected.

In this paper, the SNR values have been categorised such that the low SNR regime ranges from −20 dB to −6 dB, whereas the high SNR regime ranges from −4 dB to 4 dB. The manipulation of data has been done in such a way that each lower SNR value contributes equally with samples of percentage and the remaining percentage of samples will be contributed by higher SNR class. The ANN is trained by forming different combinations of training data-set. For instance, varying the percentage combinations of lower and higher SNR class samples, which are further used to train the ANN. After completion of training, the testing mechanism is applied to the trained ANN to compute P_d and P_f (see Fig. 6 and Table 4).

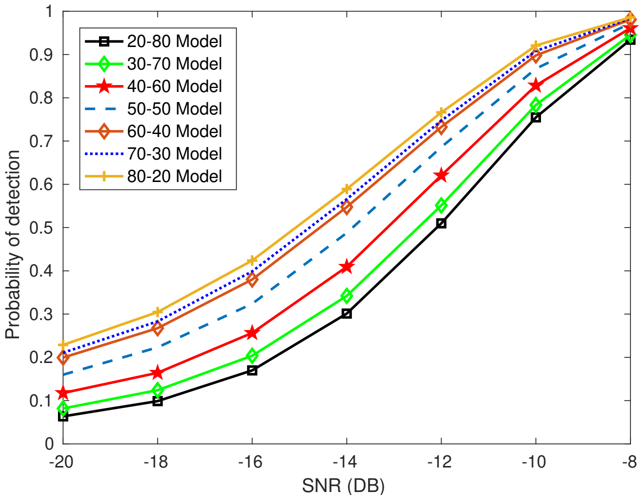


Fig. 6: P_d vs SNR for different training combinations of the model. This result indicates the trade-off between the samples from high SNR and low SNR values. We have considered the 40-60 model in all our results. (Radio technology: DCS-1800, N=100)

The radio technology DCS 1800 is used to evaluate the training and testing data-set for Fig. 6 and Table 4 (other radio technologies demonstrated similar behaviour). Here, the training models of data represent percentage distribution of higher and lower class samples in training data-set. For instance, 20-80 model represents that 50% samples which contain a primary signal in the training data-set have 20% samples from lower SNR regime and 80% samples from higher SNR regime, the remaining 50% are for AWGN samples.

As evident from Fig. 6 and Table 4, P_d and P_f values are lowest for the 20-80 model and highest for the

Table 4: P_f for different training combinations (Radio technology: DCS-1800, $N = 100$).

Training Models	Probability of False Alarm
20-80	0.0166
30-70	0.0231
40-60	0.0352
50-50	0.0570
60-40	0.0795
70-30	0.0860
80-20	0.0988

80-20 model. The fundamental reason behind this behavior is that at lower SNR the features become almost similar to the features of noise samples, so if the training data-set of primary signal contains a larger proportion of low SNR samples compared to high SNR samples, the neural network is likely to indicate the presence of a primary signal when it identifies even a slight increase in the energy value and Zhang statistic (80-20 Model). On the other hand, if the training data-set of primary signal has a larger proportion of samples of high SNR compared to low SNR, the neural network becomes more robust to identify the actual presence of a primary signal (20-80 model). Hence, every outcome depends primarily on how the neural network is being trained. In our study, we consider the 40-60 model which consists of 60% samples from higher SNR values and 40% samples from lower SNR values for the ANN to avoid overfitting on the either of lower SNR and higher SNR values and generalize better.

5.2.2 Training outcome analysis using different features

We analyze the performance of the proposed architecture considering different sets of features as follows: 1) only the energy of current samples, 2) only the Zhang statistic of current samples, 3) both energy and Zhang statistic of current samples and 4) both energy and Zhang statistic of current and previous samples. More specifically, we evaluate the performance of how trained ANN fits the cross-validation data-set.

5.2.3 Impact of variations in Data-set on Accuracy

Based on the experimental results, we observed a significant decrease in the accuracy of the proposed ANN for the cases when the data-set has signal samples with lowest SNR value of −10 dB and when the data-set has signal samples with lowest SNR value of −20 dB. This phenomenon is due to the fact that at very low SNR conditions, the feature vector seems to overlap with signal and noise and it becomes difficult for the neural network to learn the difference between signal and noise. For instance, we considered two data-sets with signal samples of lowest SNR value of −20 dB and 0 dB. Fig. 7 demonstrates that the scatter plot of signal and noise samples obtained using previous energy test-statistic as the feature forms overlapping clusters for lower SNR value of −20 dB yielding an indecisive boundary line and forms a decisive boundary line for higher SNR value of 0 dB making it easier for the ANN to identify the type of current sample. Similar trend can be observed in Fig. 8 where the scatter plot of signal and noise samples is shown using Zhang statistic value. An important observation is that the Zhang statistic performs better clustering as compared to the energy samples using energy detection which is also supported by results shown in Table 6 for all the radio technologies.

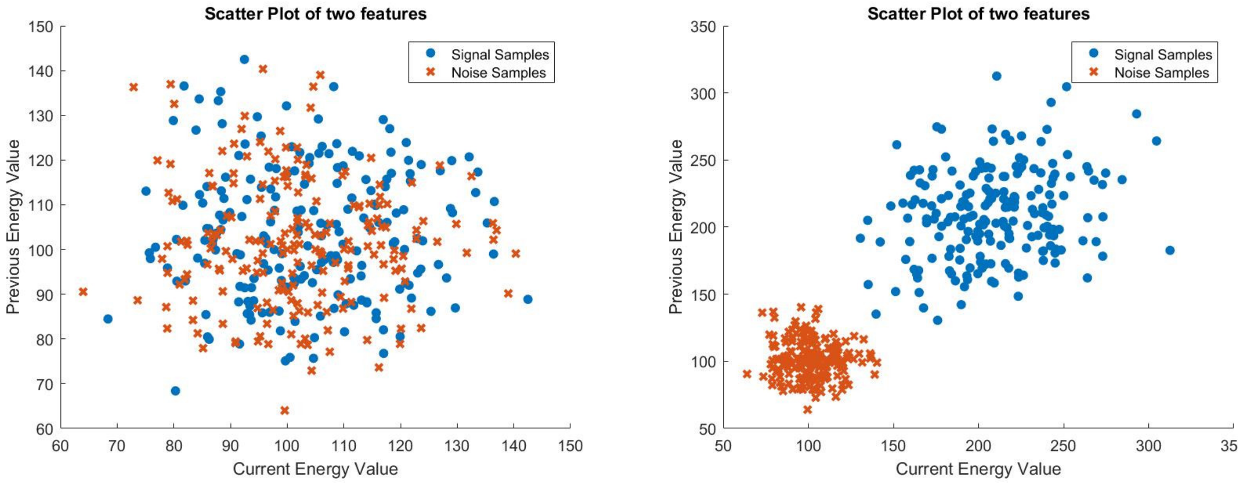
Based on the observations of different combinations of activation functions discussed previously, we conclude that the performance of the proposed ANN architecture is almost similar and yields nearly identical results. The results are summarized in Table 7. Therefore, we can conclude that an activation function with lowest computation complexity must be used for the proposed ANN, since they do not bring any major changes in its performance.

Table 5 shows the observed values of accuracy for different sets of radio technologies for a data-set consisting of 100 samples with lowest SNR value of −10 dB. As it can be appreciated, the accuracy is not significantly affected by the considered ratio technology, showing that the neural network can provide equally accurate results for different radio technologies. Similarly, a comparison of fitting accuracy values of training dataset for different sample sizes ($N = 100$ and $N = 500$) is shown in Table 5 with lowest SNR value of −20 dB in the dataset. We can notice that the results obtained for $N = 500$ are much better than that obtained for $N = 100$.

As evident from Table 5, considering only the energy of the current sensing event as a feature for training the ANN yields the worst performance compared to other

Table 5: Fitting accuracy of training data-set for different radio technologies observed using proposed ANN for different SNR values and sample size N

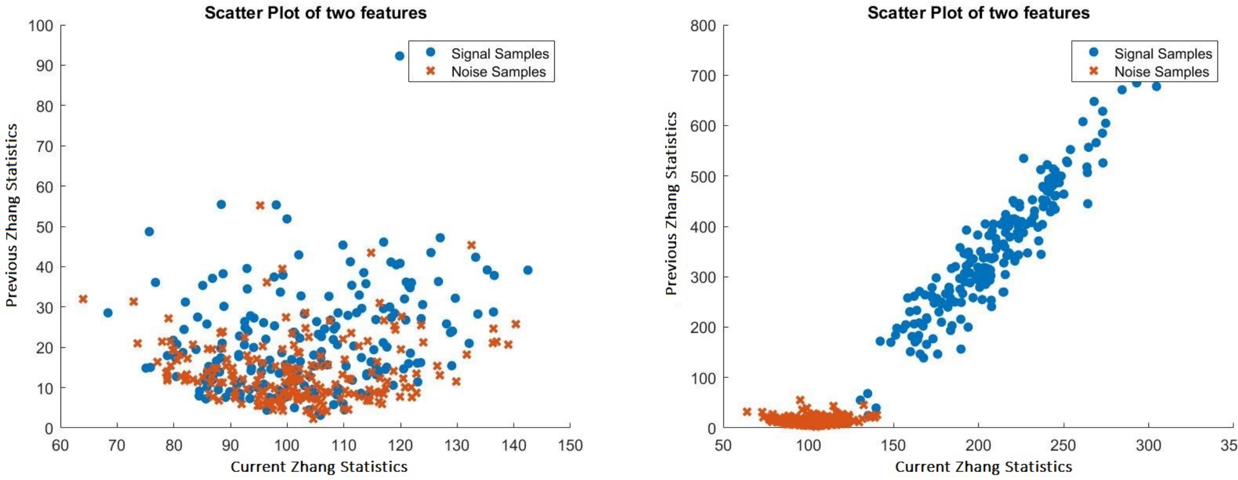
Radio Technology	Training Accuracy		
	SNR = -20 dB		SNR = -10 dB
	$N = 100$	$N = 500$	$N = 100$
FM Broadcast	87%	94.90%	96.50%
GSM (900) DL	86.46%	93.67%	94.32%
DCS (1800) DL	89.20%	95.86%	97.15%
UHF TV	88.58%	95.37%	96.96%



(a)SNR = -20 dB (low)

(b)SNR = 0 dB (high)

Fig. 7: Comparison of noise and signal samples using previous energy test-statistic and current energy value for low and high SNR. The scatter plot suggests that the energy of the signal and noise samples are close to each other at low SNR as compared to high SNR.



(a)SNR = -20 dB (low)

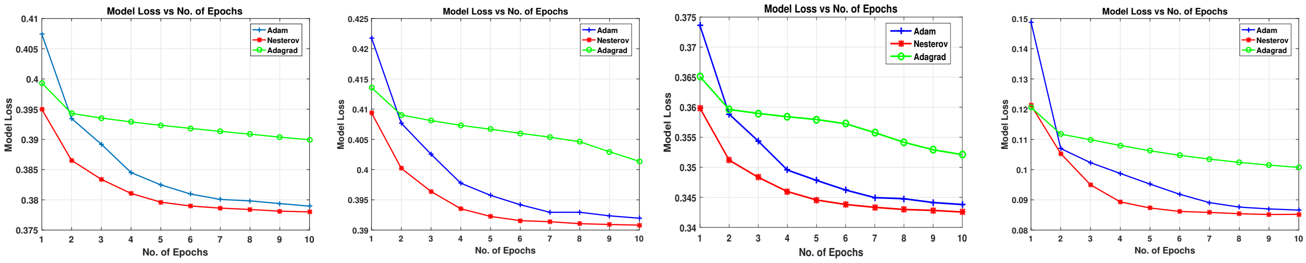
(b)SNR = 0 dB (high)

Fig. 8: Comparison of noise and signal samples using previous and current Zhang statistic for low and high SNR values. The scatter plot suggests that the Zhang statistic of the signal and noise samples are less closer as compared to the energy statistic. This is one of the motivation to use both the features in our proposed scheme.

Table 6: Accuracy measurement of ANN using different radio technologies with different set of features (N = 100)

Radio Technology	Only Energy	Only Zhang Statistic	Current Energy & Zhang Statistic	All Four Features
FM Broadcast	76.66%	83.82%	84.15%	86.82%
GSM (900) DL	76.91%	83.93%	84.27%	86.88%
DCS (1800) DL	78.19%	86.49%	86.51%	89.04%
UHF TV	78%	85.97%	86.02%	88.55%

combinations of features for all the radio technologies. It is interesting to note that when the proposed ANN is trained using both energy and Zhang statistic of the current sensing event, the accuracy on cross-validation data-set remains almost similar to when only the Zhang statistic is used as a feature, proving the effectiveness of the Zhang statistic as a feature for spectrum occupancy decision. More specifically, Zhang statistic performs better in the low SNR regime and hence acts as a mitigator for cases where energy detection fails to perform better. Owing to this strategy supported by the experimental results from Table 5, the combination of all four features provides the best accuracy for all cases.



(a) FM broadcast (b) GSM-900 DL (c) DCS-1800 DL (d) UHF TV Band

Fig. 9: Plot of loss versus epoch for various optimizers validated over different radio technologies. We notice that the Nesterov classifier outperforms the Adam and Adagrad optimizer for all radio technologies.

Table 7: Fitting accuracy of training data set for different technologies using different activation functions (the lowest SNR in data set is −10 dB).

Activation function	**FM Broadcast**		**GSM (900) DL**		**DCS (1800) DL**		**UHF Television**	
	First Iteration Accuracy	Training Accuracy	First Iteration Accuracy	Training Accuracy	First Iteration Accuracy	Training Accuracy	First Iteration Accuracy	Training Accuracy
Both layers: Sigmoid	94.33%	96.95%	92.32%	94.32%	92.50%	97.15%	95.31%	95.15%
1st layer: ReLU 2nd layer: Sigmoid	94.29%	96.93%	91.54%	94.34%	91.50%	97.22%	94.30%	96.93%
1st layer: Tanh 2nd layer: Sigmoid	95.49%	96.98%	92.50%	94.38%	90.50%	97.21%	95.49%	96.96%
Both layers: Tanh	92.09%	96.07%	89.24%	93.80%	91.50%	96.50%	92.49%	96.02%

5.3 Performance Analysis of various hyperparameters

Fig. 9 demonstrates the plots of model loss versus number of epochs. Nesterov accelerated gradient optimizer yields the best performance for all the radio technologies giving minimum loss with increase in number of epochs. This happens because if the convergence algorithm slows down at any local minima then the momentum helps it regain its speed and convergence happens quickly. Also, the novelty of Nesterov accelerated gradient is to make a big jump based on the previous momentum and then calculate the gradient followed by a correction which results in a parameter update. This makes the algorithm more responsive to changes in the gradient updates.

Table 8 shows the training accuracy for different learning rates. As shown, a learning rate of 10 gives the worst performance of proposed ANN, whereas a learning rate of 0.0001 results into the best accuracy for all the radio technologies. This happens because a large value of learning rate results in an unstable training process, where abrupt changes in the loss function may eventually skip the global minima. The learning rate is a hyperparameter which if not tuned properly, may lead the ANN to yield lower accuracy. This is due to the fact that when learning rate is very large it causes the optimizer to move around the minima and never converge on the point of minima. leading to results of lower accuracy. Hence, it becomes necessary to tune the learning rate parameter such that it yields superior performance.

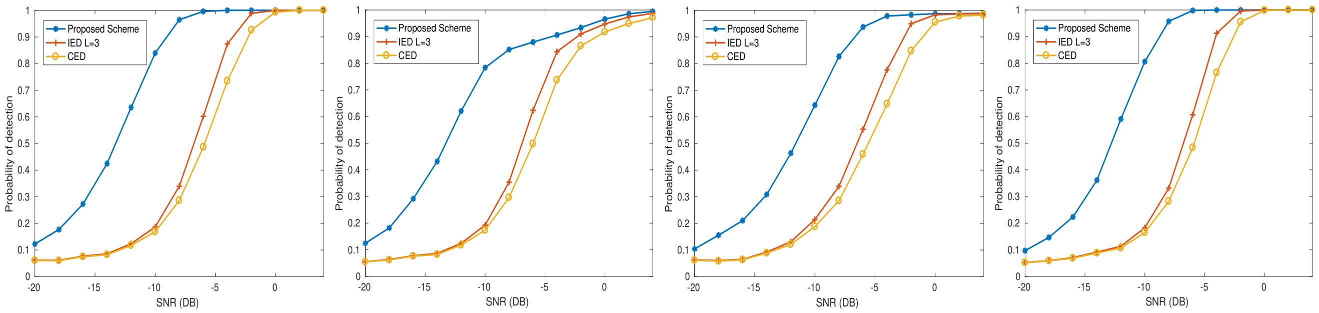
Table 8: Fitting accuracy of training dataset for different technology for different learning rate (the lowest SNR in data-set is −10 dB).

Learning Rate	**Training Accuracy**			
	FM Broadcast	**GSM (900) DL**	**DCS (1800) DL**	**UHF TV**
0.0001	95.96%	93.11%	96.39%	92.40%
0.2	92.44%	92.16%	95.01%	91.02%
1	93.77%	92.74%	50%	50%
10	50%	50%	50%	50%

5.4 Overall Performance Analysis of the Proposed ANN scheme

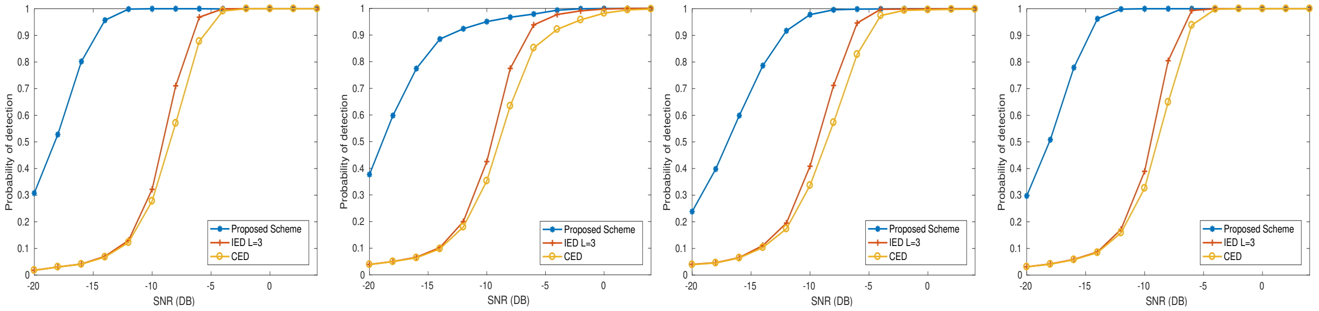
The proposed ANN based sensing scheme has been evaluated for all the radio technologies mentioned in Table 3 with different combinations of sample sizes and false alarm rates as evident from Fig. 10 ($N = 100$) and Fig. 11 ($N = 500$) where the detection performance of the proposed scheme is compared with CED and IED sensing schemes.

The plot in Fig. 12 shows the P_d v/s SNR comparison for the proposed scheme with NBC in [32] and BPNN. The spectrum data in our case are acquired through the empirical test-bed setup. However, for a fair comparison, we have generated the data through the simulation parameters as provided in [32]. More-



(a) DCS-1800 DL (b) FM Broadcast (c) GSM-900 DL (d) UHF TV Band

Fig. 10: Detection performance of proposed optimal hyperparameter tuned ANN based spectrum sensing method (P_d vs SNR) for sample size, $N = 100$. The proposed scheme is compared with IED (L=3) and CED schemes for all radio technologies.



(a) DCS-1800 DL (b) FM broadcast (c) GSM-900 DL (d) UHF TV Band

Fig. 11: Detection performance of proposed optimal hyperparameter tuned ANN based spectrum sensing method (P_d vs SNR) for sample size, $N = 500$. The proposed scheme is compared with IED (L=3) and CED schemes for all radio technologies.

over, we have trained our proposed model with the features F_1 and F_2 as mentioned in the article. The number of training samples per class in Fig. 12 (a) is 100 while in Fig. 12 (b) it is 250. We can notice that, as compared to the BPNN scheme without hyperparameter tuning, our proposed scheme performs quite close with the NBC at low SNR regime. This is due to the fact that the proposed algorithm learns better when the hyperparameters are tuned as compared to the trained model without hyperparameter tuning.

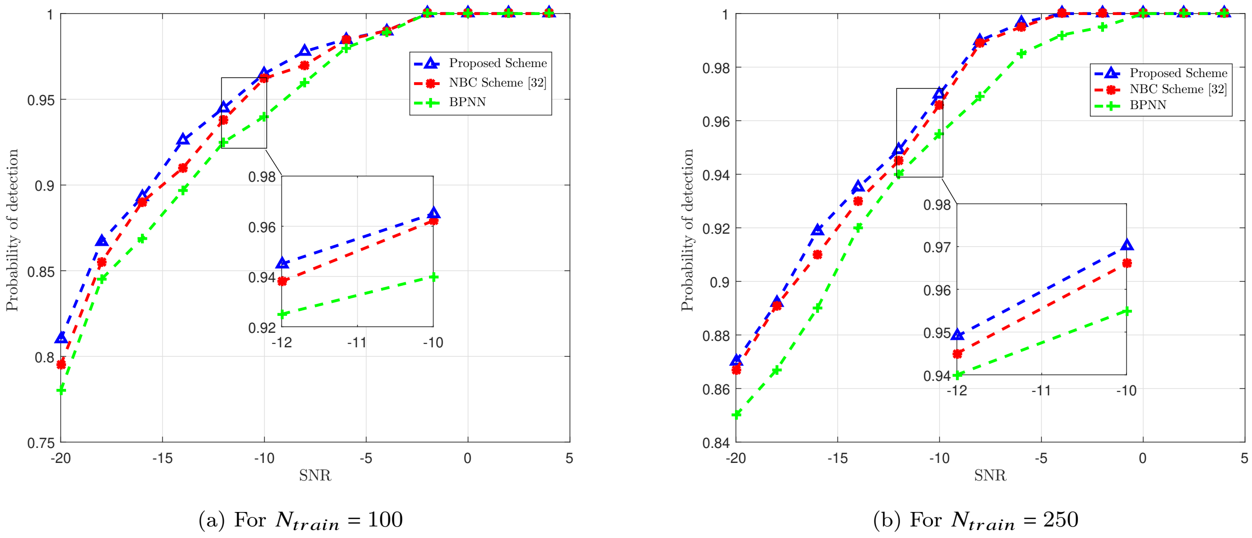
Table 9 shows the computational complexities of the algorithms considered in this work. In the table, N denotes sample size, N_{train} denotes the number of training samples, E denotes the number of epochs, m denotes the number of features, C indicates the number of classes while i, j denote the nodes in the first and second layer, respectively. We can notice that although CED and IED are computationally simple, ML based scheme outperforms them in terms of detection probability. Moreover, the computational complexity of NBC is a function of C (i.e., twice the number of SNR values considered). This makes the computational complexity of NBC very high as compared to the proposed model.

The above observation is also substantiated by the average execution time as shown in Table 10. We also notice that the computational complexity of BPNN and our proposed scheme will be the same. Furthermore, the proposed scheme is also compared with NBC and BPNN in terms of average execution time, Area Under the Curve (AUC) and F-Score as shown in Table 10. F-Score is often used to quantify the accuracy and it can be interpreted as a weighted average of the precision and recall. We can observe from Table 10 and Fig. 13 that the average execution time for NBC scheme is approximately 0.911 ms while that for the proposed scheme is approximately 0.633 ms i.e., ≈ 44% less. The comparable accuracy alongwith the low computational complexity and lower execution time are the key advantages of the proposed scheme, which is an important parameter for the real time systems.

In order to realize the difference in the proposed novel sensing scheme in comparison to the existing schemes, a bar graph is shown in Fig. 14 where the detection

Table 9: Performance comparison with other algorithms

Parameter \Algorithms	Proposed Scheme	NBC [32]	BPNN
Average execution time (ms)	0.637	0.911	0.614
Area under the curve (AUC)	0.971	0.952	0.946
F-Score	0.86	0.81	0.79



(a) For N_{train} = 100

(b) For N_{train} = 250

Fig. 12: P_d vs SNR comparison of the proposed scheme with NBC [32] and BPNN. Training examples per class in (a) is 100 while in (b) is 250.

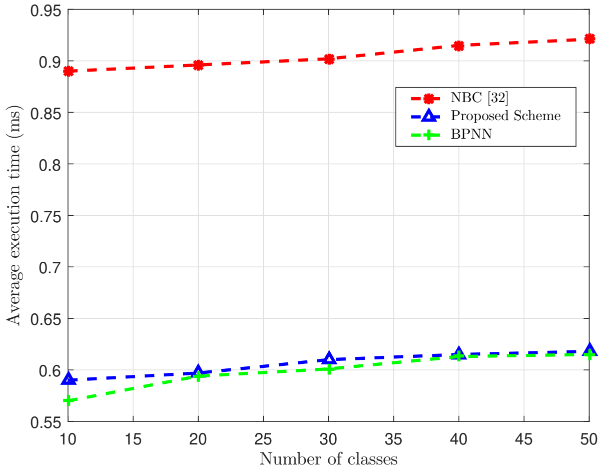


Fig. 13: Comparison of average execution time v/s number of classes for the proposed scheme, NBC [32] and BPNN.

probability values are plotted for the SNR value of -8 dB for a sample size N = 100. These values have been considered particularly because a sample size of 100 is more efficient for a system as compared to a sample size of 500. Similarly, the SNR value of -8 dB lies in the low SNR regime and hence is a true test for the proposed scheme to be justify its significance. As evident from Fig. 14, an analysis of performance improvement in terms of percentage using the proposed scheme against the IED and CED schemes for all the four radio technologies for number of samples, N = 100 has been demonstrated. Clearly, the proposed scheme outperforms the standard spectrum sensing schemes. For DCS (900) DL radio technology, the proposed ANN based scheme performs 67% better than the IED scheme

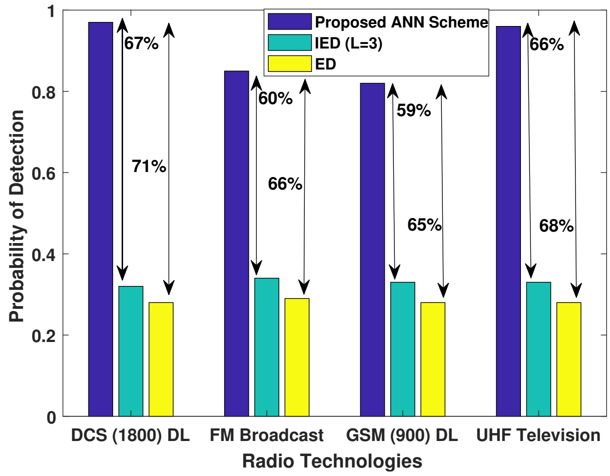


Fig. 14: Analysis of improvement in detection probability using proposed scheme as compared to CED and IED schemes (N = 100) for all radio technologies. On an average, the proposed scheme performs ≈ 63% better than the IED scheme and ≈ 67.5% better than the CED scheme.

Table 10: Computational complexities of the various schemes

Algorithms	Computational Complexity
CED	$O(N \cdot K)$
IED	$O(N \cdot K)$
NBC	$O(N_{train} \cdot m \cdot C)$
BPNN	$O(N_{train} \cdot E \cdot i \cdot j)$
Proposed Scheme	$O(N_{train} \cdot E \cdot i \cdot j)$

Table 11: False alarm rate for the ANN model trained on different radio technologies with different sample sizes (N).

Radio Technology	P_{fa} **at** $N = 100$	P_{fa} **at** $N = 500$
FM Broadcast	0.0440	0.0220
GSM (900) DL	0.0434	0.0200
DCS (1800) DL	0.0352	0.0137
UHF Television	0.0414	0.0191

and 71% better than the ED scheme. Similarly, the percentage improvement for all other schemes has been shown in Fig. 14.

Further, as the ANN learns about the spectrum sensing decision from the features dynamically, the associated false alarm rate remains almost static for a particular sample size for different radio technologies as evident from Table 11. Moreover, it is worth noting that as the sample size increases from 100 to 500, there is a significant reduction in the false alarm rate. This is due to the fact that as the number of samples increases, the ANN learns better as compared to a lower count of samples. However, one cannot overshoot the number of samples, since it may lead to issues like overfitting. Similar reasoning can be applied to the behaviour of detection probability P_d as well.

6 Conclusion

The designed artificial neural network for spectrum sensing have an outstanding ability to learn the non-linear behaviour of the dataset, yielding higher classification accuracy than IED and CED. In this work, we provide energy and Zhang statistic of current and previous sensing events as an input features for improving spectrum sensing performance. Furthermore, we determine the optimal set of hyperparameters such as optimization techniques, learning rate and activation functions. Moreover, the performance of proposed scheme is evaluated and validated on the spectrum data of various radio technologies captured using an empirical testbed setup.

The optimal set of hyperparameters yielding the best performance of ANN are obtained by using the $40 - 60$ model of SNR values (40% of the samples are from lower SNR values and 60% of the samples are from higher SNR values), all the four features (energy and Zhang statistic of current and previous samples) for training, Nesterov Accelerated Gradient optimizer as the optimization algorithm and learning rate of 0.0001 and sample size $N = 500$. However, the sample size can be varied based on the system requirement of the particular application. Setting these parameters, we obtain performance improvement of approximately 63% on averaging the performance gain of all four RF technologies, which demonstrates a significant superiority of the proposed scheme. Notice that this work considered only a single PU and a single SU. However, the study of multiple PU and multiple SU [60] which includes the PU channel selection among multiple PUs is an interesting topic of further research.

Acknowledgments

Results in parts were published in proceedings of IEEE PIMRC conference, held in September 2017 at Montreal, Canada [38]. The authors would like to thank financial support received from UKIERI under the DST Thematic Partnership 2016-17 (ref. DST/INT/UK/P-150/2016). The authors also thank Ahmedabad University and University of Liverpool for infrastructure support.

References

1. J. Lundén, V. Koivunen, and H. V. Poor, “Spectrum Exploration and Exploitation for Cognitive Radio: Recent Advances,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 123–140, 2015.
2. M. Cardenas-Juarez, M. A. Diaz-Ibarra, U. Pineda-Rico, A. Arce, and E. Stevens-Navarro, “On spectrum occupancy measurements at 2.4 GHz ISM band for cognitive radio applications,” in *Proc. of CONIELECOMP*, 2016, pp. 25–31.
3. J. Eze, S. Zhang, E. Liu, and E. Eze, “Cognitive radio technology assisted vehicular ad-hoc networks (vanets): Current status, challenges, and research trends,” in *2017 23rd International Conference on Automation and Computing (ICAC)*, Sep. 2017, pp. 1–6.
4. Y. C. Liang, K. C. Chen, G. Y. Li, and P. Mähönen, “Cognitive radio networking and communications: An overview,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3386–3407, 2011.
5. Y. Chen and H. S. Oh, “A survey of measurement-based spectrum occupancy modeling for cognitive radios,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 848–859, 2016.
6. Isha, A.Malik, and A. Bakshi, “Spectrum sensing techniques in cognitive radio based sensor networks: A survey,” *Int. J. Appl. Eng. Res.*, vol. 10, no. 8, pp. 19 063–19 076, 2015.
7. M. Bkassiny, Y. Li, and S. K. Jayaweera, “A survey on machine-learning techniques in cognitive radios,” *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, Third 2013.
8. N. Abbas, Y. Nasser, and K. El Ahmad, “Recent advances on artificial intelligence and learning techniques in cognitive radio networks,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2015, no. 1, p. 174, 2015.
9. 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.
10. H. Urkowitz, “Energy detection of unknown deterministic signals,” *Proc. IEEE*, vol. 55, no. 4, pp. 523–531, 1967.
11. B. Gajera, D. K. Patel, B. Soni, and M. López-Benítez, “Performance evaluation of improved energy detection under signal and noise uncertainties in cognitive radio networks,” in *Proc. of IEEE ICSigSys*, 2019, pp. 131–137.

12. B. Gaiera, D. K. Patel, B. Soni, and M. López-Benítez, "Experimental performance evaluation of improved energy detection under noise uncertainty in low SNR regime," in *Proc. of IEEE WCNCW*, 2019.
13. H. Wang, E. H. Yang, Z. Zhao, and W. Zhang, "Spectrum sensing in cognitive radio using goodness of fit testing," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5427–5430, Nov. 2009.
14. S. Rostami, K. Arshad, and K. Moessner, "Order-statistic based spectrum sensing for cognitive radio," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 592–595, 2012.
15. G. Zhang, X. Wang, Y.-C. Liang, and J. Liu, "Fast and robust spectrum sensing via Kolmogorov-Smirnov test," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3410–3416, 2010.
16. K. Arshad and K. Moessner, "Robust spectrum sensing based on statistical tests," *IET Commun.*, vol. 7, no. 9, pp. 808–817, 2013.
17. D. Teguig, V. L. Nir, and B. Scheers, "Spectrum sensing method based on likelihood ratio goodness-of-fit test," *IET Electronics Letters*, vol. 51, no. 3, pp. 253–255, 2015.
18. D. K. Patel and Y. N. Trivedi, "LRS-$\boldsymbol{G}^2$ based non-parametric spectrum sensing for cognitive radio," in *Proc. CROWNCOM*, 2016, pp. 330–341.
19. D. K. Patel, B. Soni, and M. Lopez-Benitez, "Improved likelihood ratio statistic based cooperative spectrum sensing for cognitive radio," *IET Commun.*, Dec. 2019.
20. C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
21. J. Tian, Y. Pei, Y. Huang, and Y. 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.
22. T. 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, Dec 2017.
23. S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb 2018.
24. J. Wang, C. Jiang, H. Zhang, Y. Ren, K. Chen, and L. Hanzo, "Thirty years of machine learning: The road to Pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, pp. 1–1, 2020.
25. P. Cheng, Z. Chen, M. Ding, Y. Li, B. Vucetic, and D. Niyato, "Spectrum intelligent radio: Technology, development, and future trends," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 12–18, 2020.
26. A. Agarwal, S. Dubey, M. A. Khan, R. Gangopadhyay, and S. Debnath, "Learning based primary user activity prediction in cognitive radio networks for efficient dynamic spectrum access," in *Proc.of SPCOM*, June 2016, pp. 1–5.
27. C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of machine learning to cognitive radio networks," *IEEE Wireless Commun.*, vol. 14, no. 4, pp. 47–52, Aug. 2007.
28. F. Azmat, Y. Chen, and N. Stocks, "Analysis of spectrum occupancy using machine learning algorithms," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 6853–6860, Sept. 2016.
29. S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Communications*, vol. 17, no. 2, pp. 138–148, 2020.
30. D. Zhang and X. Zhai, "SVM-based spectrum sensing in cognitive radio," in *Int. Conf. Wirel. Commun. Netw. Mob. Comput.*, 2011, pp. 1–4.
31. M. Panchal, D. Patel, and S. Chaudhary, "Spectrum occupancy classification using svm-radial basis function," in *Proc. of CROWNCOM.* Springer, 2017, pp. 112–127.
32. 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. 11 374–11 378, 2019.
33. D. Wang and Z. Yang, "An novel spectrum sensing scheme combined with machine learning," in *Proc. of CISP-BMEI*, 2016, pp. 1293–1297.
34. T. C. Clancy, A. Khawar, and T. R. Newman, "Robust signal classification using unsupervised learning," *IEEE Trans. Wireless Commun.*, vol. 10, no. 4, pp. 1289–1299, 2011.
35. V. Kumar, D. C. Kandpal, M. Jain, R. Gangopadhyay, and S. Debnath, "K-mean clustering based cooperative spectrum sensing in generalized $\boldsymbol{\kappa}$-$\boldsymbol{\mu}$ fading channels," in *Proc. of IEEE NCC*, 2016, pp. 1–5.
36. A. M. Mikaeil, B. Guo, and Z. Wang, "Machine learning to data fusion approach for cooperative spectrum sensing," in *Proc. of Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2014, pp. 429–434.
37. Y.-J. Tang, Q.-Y. Zhang, and W. Lin, "Artificial neural network based spectrum sensing method for cognitive radio," in *Proc. of IEEE WiCOM*, 2010, pp. 1–4.
38. M. R. Vyas, D. K. Patel, and M. López-Benítez, "Artificial neural network based hybrid spectrum sensing scheme for cognitive radio," in *Proc. of PIMRC*, Oct 2017, pp. 1–7.
39. Y. Zhang, J. Hou, V. Towhidlou, and M. R. Shikh-Bahaei, "A neural network prediction-based adaptive mode selection scheme in full-duplex cognitive networks," *IEEE Trans. Cognitive Commun. and Netw.*, vol. 5, no. 3, pp. 540–553, 2019.
40. Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors*, vol. 19, no. 1, p. 126, 2019.
41. C. Liu, J. Wang, X. Liua, and Y. Liang, "Deep CM-CNN for spectrum sensing in cognitive radio," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2019.
42. C. Liu, X. Liu, and Y. Liang, "Deep CNN for spectrum sensing in cognitive radio," in *Proc. of IEEE ICC*, May 2019, pp. 1–6.
43. J. Xie, C. Liu, Y. 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, June 2019.
44. 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.
45. Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Sensing OFDM signal: A deep learning approach," *IEEE Trans. Commun.*, pp. 1–1, 2019.
46. J. Xie, J. Fang, C. Liu, and L. Yang, "Unsupervised deep spectrum sensing: A variational auto-encoder based approach," *IEEE Trans. Veh. Technol.*, pp. 1–1, 2020.
47. L. Yu, J. Chen, G. Ding, Y. Tu, J. Yang, and J. Sun, "Spectrum prediction based on Taguchi method in deep learning with long short-term memory," *IEEE Access*, vol. 6, pp. 45 923–45 933, 2018.
48. N. Balwani, D. K. Patel, B. Soni, and M. López-Benítez, "Long short-term memory based spectrum sensing scheme for cognitive radio," in *Proc. of IEEE PIMRC*, Sept. 2019, pp. 1–6.
49. 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. 97 437–97 451, 2020.

50. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.
51. R. Shindjalova, K. Prodanova, and V. Svechtarov, "Modeling data for tilted implants in grafted with bio-oss maxillary sinuses using logistic regression," in *AIP Conf. Proc.*, vol. 1631, no. 1. American Institute of Physics, 2014, pp. 58–62.
52. I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.
53. R. E. Schapire, "A brief introduction to boosting," in *Int. Joi. Conf. Art. Inte.*, vol. 99, 1999, pp. 1401–1406.
54. M. López-Benítez and F. Casadevall, "Improved energy detection spectrum sensing for cognitive radio," *IET Commun.*, vol. 6, no. 8, pp. 785–796, 2012.
55. D. K. Patel, B. Soni, and M. López-Benítez, "On the estimation of primary user activity statistics for long and short time scale models in cognitive radio," *Wireless Networks*, vol. 25, no. 8, pp. 5099–5111, Aug. 2019.
56. M. López-Benítez, F. Casadevall, and C. Martella, "Performance of spectrum sensing for cognitive radio based on field measurements of various radio technologies," in *European Wireless Conference*, 2010.
57. E. R. LLC, "Universal Software Radio Peripheral (USRP)," Available: http://www.ettus.com.
58. E. Blossom, "GNU radio: tools for exploring the radio frequency spectrum," *Linux journal*, vol. 2004, no. 122, p. 4, 2004.
59. M. Wellens and P. Mähönen, "Lessons learned from an extensive spectrum occupancy measurement campaign and a stochastic duty cycle model," *Mob. networks Appl.*, vol. 15, no. 3, pp. 461–474, 2010.
60. T. Febrianto, J. Hou, and M. Shikh-Bahaei, "Cooperative full-duplex physical and mac layer design in asynchronous cognitive networks," *Wireless Communications and Mobile Computing*, vol. 2017, 2017.