sensors

MDPI

Article

Cooperative multiband spectrum sensing using radio environment maps and neural networks

Yanqueleth Molina-Tenorio [1], Alfonso Prieto-Guerrero [2], Rafael Aguilar-Gonzalez [3,4,*] and Miguel Lopez-Benitez [5,6]

[1] Information Science and Technology Ph. D.; Metropolitan Autonomous University; Mexico City, Mexico; yanqueleth@xanum.uam.mx
[2] Electrical Engineering Department; Metropolitan Autonomous University; Mexico City, Mexico; apg@xanum.uam.mx
[3] Faculty of Science; Autonomous University of San Luis Potosi; San Luis Potosi, Mexico.
[4] Engineering Department, Arkansas State University Campus Queretaro, Queretaro, Mexico.
[5] Department of Electrical Engineering and Electronics; University of Liverpool; Liverpool, United Kingdom; mlopben@liverpool.ac.uk
[6] ARIES Research Centre; Antonio de Nebrija University; Madrid, Spain.
* Correspondence: rafael.aguilar@uaslp.mx

Abstract: Cognitive radio networks require high capacity and accuracy to detect the presence of licensed users (or primary users, PUs) in the sensed spectrum. Also, they must correctly locate the spectral opportunities (or holes), so that they are taken advantage by non-licensed users (or secondary users, SUs). In this research work, a centralized network of cognitive radios is proposed for monitoring a multiband spectrum in real-time and implemented in a real wireless communication environment through generic communication devices (SDRs, software-defined radios). Each SU uses locally a monitoring technique based on sample entropy to determine spectrum occupancy. Determined features (power, bandwidth, central frequency) of possible detected PUs, are uploaded to a database. The uploaded data are then processed by a central entity. The objective of this work is to determine the number of PUs, their carrier frequency, bandwidth, and the spectral gaps in the sensed spectrum in a specific area through the construction of radio-electric environment maps. For this, a comparison of results, considering classical digital signal processing methods and neural networks performed by the central entity, is carried out. Results show that both proposed cognitive networks (that one working with a central entity using typical signal processing and that one performing with neural networks) accurately locate PUs, give opportunely information to SUs to transmit, avoiding the hidden terminal problem. However, the best performing cognitive radio network is that working with neural networks for the accuracy of detecting PUs, on both carrier frequency and bandwidth.

Keywords: multiband spectrum sensing; cognitive radios; radio environment maps; neural networks; cooperative sensor networks, real-time implementation

Citation: To be added by editorial staff during production.

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

Copyright: © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Cognitive radio (CR) is a concept involving a communication device that is capable to know the spectral behavior in its environment adapting to this. Taking advantage of spectral gaps (or holes) that are not being utilized by the licensed users, also known as primary users (PUs), the CR technology allows to non-licensed users, called as secondary users (SUs), detect these available parts of the spectrum [1]. Specifically, the operation of a CR involves four stages or functions: spectrum sensing, spectrum sharing, spectrum decision, and spectrum mobility. Spectrum sensing (SS) is a fundamental task, where the presence of one or more PUs is detected, i.e., this stage dictates whether the sensed spectrum is occupied or empty [2]. Usually, this task is done in single bands, however, currently, there exists a paradigm in which multiple bands, not necessarily contiguous, are included, called multiband spectrum sensing (MBSS) [3].

In the context, many MBSS techniques issued from digital signal processing and/or machine learning (ML), such as wavelets, compressed sensing, energy detectors, blind or semi-blind methods, have been proposed; essentially in simulated scenarios [4]–[10]. Some other works have been implemented in a real wireless communications environment [11]–[16], where software defined radio (SDR) and universal software radio peripheral (USRP) technologies have been employed. Recently, SDR devices such as the HackRF One, the LimeSDR Mini and the RTL-SDR have become extremely popular because of their affordable cost and good performances [17], [18]. These items, which are generic communication devices, offer to radio equipment the flexibility of a programmable system. In other words, the behavior of a communication system can be modified simply by changing its software, permitting to anyone, including hobbyists on a budget, have access to real radio spectrum.

In addition to knowing the PUs behavior in the frequency domain, and to avoid the hidden terminal issue (but not the only reason), a knowledge of the behavior of this radio spectrum in their specific influence geographical area, is necessary. For this reason, the idea to include a geographic tool constructed with the radio electric information provided by the SUs issued from a real environment, is pertinent. Under this perspective, in recent years, the ability to build a radio environment map (REM) has become very important. A REM is "*a tool that combines information collected from the radio environment, such as received signal intensity, interference measurements, propagation conditions, etc., for specific locations and frequencies, with the aim of building a map that provides an overview of coverage of the network*" [19]. In this way, REMs permitting to characterize position, directivity, power, and modulation type of the PUs, have become a challenging task in CR networks (CRN) design [20]. Indeed, in [21] REMs are used to locate relevant PUs in a geographic region of interest, characterizing their positions, directivities, powers, and modulation types. Likewise in [22], REMs have been considered to sense the spectrum based on an adaptive compressed spectrum sensing algorithm, contributing with spatial information to the network, which has the quality of adapting to the radio environment. REMs are a very flexible tool as is in [23], where they are used in combination with ML, in order to determine the effective coverage area that is perceived by a cognitive sensor network, being it correctly estimated around 92 %.

Another important tool, currently back thanks to computing power and the amount of available data, are neural networks (NN), widely used from pattern recognition and image classification to financial market behavior prediction and autonomous vehicle driving [24]. In the context of spectrum sensing, for example in [25], a NN is implemented to obtain the local information of the spectrum detection for a single node (spatial features and temporal features). The information (extracted features) of multiple nodes feeds another NN permitting a result of cooperation in the CRN. Based on these both paradigms, REMs and NNs, in this work is proposed a new methodology in the context of a CRN, considering the implementation of a MBSS method involving a network integrated by low-cost SDR devices, in a controlled realistic wireless communications environment.

In this work a CRN is presented for the monitoring of a multiband spectrum, in which it is proposed to locate the PUs with their characteristics (bandwidth, carrier frequency and power) in a specific area. In addition, it locates the spectral opportunities where the SUs can be located. This idea is capable of being implemented in real time, unlike some works mentioned above.

This work is organized as follows. First, in Section 2 the theoretical basis about REMs and NNs are briefly developed. Section 3 presents the previous work implemented before by the authors for the MBSS, and used as a base for this new proposal. Section 4 explains in detail the new proposed methodology. In Section 5, the implemented real scenario is detailed. Finally, Section 6 shows the results, conclusions, and a discussion.

2. Theoretical basis of REMs and NNs

In this section, REMs and the NNs are briefly and concisely explained. Both techniques are introduced in the methodology explained in Section 4.

2.1. Construction of radio enviroment maps

REMs is a technique used in telecommunications research to represent the power distribution at a given specific area, emitted by different radio sources (one or more). A REM is constructed by collecting power measurements at different points in this specific area and interpolating them, to produce a graphical representation (map) of the power distribution (or covering) of the emitted signals, as it is shown in Figure 1. These maps are generally used to evaluate and predict the behavior of radio signals in a given environment, which is important for capacity planning, communication system design, and troubleshooting related to interference. In addition, REMs are also used in CR applications, where devices can use the information provided by the REM to select the most suitable frequency to transmit data and avoid interference with other devices in the same location [26].

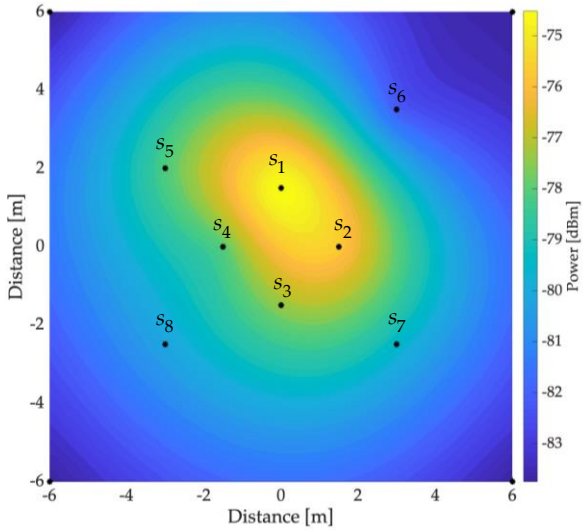

Figure 1. Example of a REM. Black dots indicate locations where the power measurements are taken based on 8 sensors. In this case, only a transmission source with -60 dBm is considered.

Collection and interpolation of power measurements are the two processes playing an essential role in the construction of a *correct* map. In this case, interpolation methods [27] are critical to *correctly* estimate an unknown value between two or more known points (measurement points). In other words, a REM is simply a smoothed estimate of the power distribution based on few known values of measured power. Two interpolation methods are widely used to build REMs, the Inverse Distance Weighting (IDW) and the Kriging. Both methods are explained below.

2.1.1. IDW method

The inverse distance weighting interpolation method [28], [29] is a simple and easy-to-implement technique, widely used in applications such as precipitation estimation, topographic data interpolation, and air pollution estimation. In fact, IDW is a classical method of interpolation in spatial analysis and one of the most commonly used in geostatistical and mathematical interpolation [30].

IDW estimates the value of a variable at a specific point based on known values at nearby points. This method is based on the idea that nearby points have a greater impact on the estimate than more distant points. For this, IDW uses a formula that assigns a weight to each known point based on the distance between these ones and the unknown point. The weights are determined by considering the inverse of the distance, in this way the closest points have higher weights and, the farther ones have lower weights. In the next IDW method is explained.

Consider data transposed vector $\mathbf{z} = \left(z\left(s_1\right),...,z\left(s_n\right)\right)^T$, being observations from a random process. In our case, $\mathbf{z}$ contains the average power at each point of the considered sampled region (see Figure 1):

$$\left\{z\left(s\right): s \in D\right\}; D \subset \mathbb{R}^2, \tag{1}$$

at known geographic locations $s_1,...,s_n$. The final estimate is obtained by means of the weighted sum of the known values in the nearby points, given by [29]:

$$Z(s) = \frac{\sum\left(\frac{s_i}{d\left(s_i,s\right)^p}\right)\left(w_i\left(s_i\right)\right)}{\sum\frac{1}{d\left(s_i,s\right)^p}}, \tag{2}$$

where $Z\left(x\right)$ is the estimated value of the variable at the unknown point s, z_i is the known value of the variable at the *i*-th point s_i, $d\left(s_i,s\right)$ is the Euclidean distance between the unknown point s and the point s_i, $p \in \mathbb{R}$ is a smoothing parameter controlling the influence of the known points over the estimated values and, $w_i\left(s_i\right)$ is the weight assigned to each known point s_i, in order to take into account the uncertainty in the known data. In practice, a value of $p = 2$ or $p = 3$ is usually used. The estimated values of $Z\left(s\right)$ will result in a mesh of unknown points. However, this method has some limitations, such as a tendency to smooth out data variability and produce an inaccurate estimate in areas where there are few known points. In this specific work, $Z\left(s\right)$ represents the power spectral density (PSD) in the different points around the sensed area.

2.1.2. Kriging method

The Kriging method [31] is a interpolation technique deriving from regionalized variable theory. It depends on expressing spatial variation of the property in terms of the variogram, and it minimizes the prediction errors which are themselves estimated [32]. The goal of Kriging is to find the estimate that is the most accurate and has the least uncertainty. To do this, the Kriging method considers not only the value of the variable at the known points, but also the distribution of the variable in space and its correlation with the known points. Kriging method can be established as follows below [31].

For this, the points and the region mentioned in $\left(1\right)$, assume μ is known and that

$$z\left(s\right) = \mu + \delta\left(s\right); s \in D, \tag{3}$$

are considered with known covariance function:

$$C\left(s,u\right) \equiv \mathrm{cov}\left(z\left(s\right),z\left(u\right)\right); s,u \in D, \tag{4}$$

where $\delta(\cdot)$ is a zero-mean stochastic process. Thus, the best linear unbiased predictor of $z(s_0)$ is obtained by minimizing [33]:

$$E\left(z(s_0)-\sum_{h=1}^{n}\lambda_h z(s_h)\right)^2, \quad (5)$$

with $\lambda_1,...,\lambda_n$, subject to

$$\sum_{h=1}^{n}\lambda_h = 1. \quad (6)$$

By using the method of Lagrange multipliers, the optimal values are computed. For this, the mean-squared prediction error (Eq. (5)) is obtained by:

$$\begin{aligned} &E\left(z(s_0)-\hat{z}(s_0)\right)^2 = \\ &= C(s_0,s_0)-c'C^{-1}c+\left(1-c'C^{-1}1\right)^2\left(1'C^{-1}1\right)^{-1}, \end{aligned} \quad (7)$$

where

$$c \equiv \left(C(s_0,s_1),...,C(s_0,s_n)\right)', \quad (8)$$

$$C \equiv \left(C(s_h,s_j)\right), \quad (9)$$

besides,

$$\hat{z}(s_0) = \gamma'\Gamma^{-1}Z+\left(1-\gamma'\Gamma^{-1}1\right)\left(1'\Gamma^{-1}1\right)^{-1}\left(1'\Gamma^{-1}Z\right), \quad (10)$$

$$\begin{aligned} &E\left(z(s_0)-\hat{z}(s_0)\right)^2 = \\ &= \gamma\Gamma^{-1}\gamma-\left(1-\gamma\Gamma^{-1}1\right)^2\left(1'\Gamma^{-1}1\right)^{-1}, \end{aligned} \quad (11)$$

where $\gamma \equiv \left(\gamma(s_0,s_1),...,\gamma(s_0,s_n)\right)'$, Γ is an $n\times n$ matrix whose $(i,j)^{th}$ element is $\gamma(s_i,s_j)$ and

$$2\gamma(s_i,s_j) \equiv C(s_i,s_i)+C(s_j,s_j)-2C(s_i,s_j), \quad (12)$$

is called the variogram [$\gamma(s_i,s_j)$ is named the semivariogram [33]. The variogram, a very useful tool in the modeling of spatial variables, provides a description of how the data are related (correlated) with distance. In this way, Kriging method allows to accomplish the spatial interpolation, using the sampled data and the variogram information to estimate the variance of the values of the variable at the unsampled points [34], [35]. In summary, Kriging is a more precise method than other interpolation methods such as IDW or cubic interpolation [30], suitable for applications in which the data have a certain spatial dependence.

2.2. Neural Networks

Neural networks are a type of artificial intelligence model that is inspired by the structure and functioning of the human brain. These networks are made up of many nodes, also known as "neurons", which are connected to each other through "synapses". Each neuron receives inputs from other neurons, processes these inputs through an activation function, and sends its output to other neurons, as it is shown in Figure 2. The combination of inputs and connections between neurons allows a neural network *to learn* complex tasks from the selected training data. In general, neural networks are a very powerful tool for machine learning because they can model complex relationships between inputs and outputs making accurate inferences from real data. However, for these tasks require a large amount of training data, being computationally intensive (and sometimes impossible) to train and use in real-time implementations *[36]*.

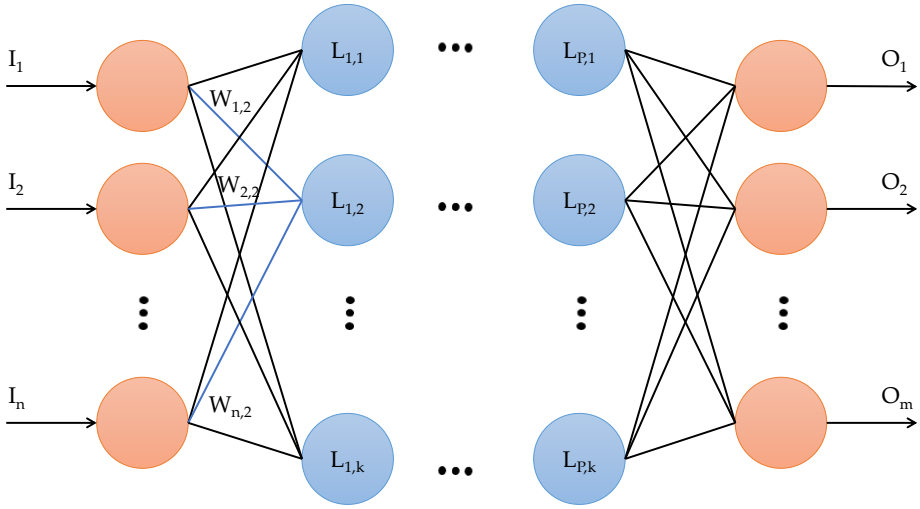

Figure 2. Neural network concept: the multilayer perceptron.

In this work, the multilayer perceptron (MLP), as it is shwon in Figure 2, is used. This type of artificial neural network consists of an input layer, one or more hidden layers, and one output layer. In the MLP, neurons in each layer are connected to neurons in the next layer and they use an activation function to determine the neuron's output. Information flows only in one direction, from the input layer to the output layer, passing through the hidden layers, which are the intermediate layers between the input and the output [37]. The general formula for calculating the output of a multilayer perceptron is given by [38]:

$$O = f\left(\sum\left(W_{i,j}\, I_i\right) + b\right), \tag{13}$$

where O is the output of the multilayer perceptron, f is the activation function, which can be a step function, sigmoid, ReLU (rectified linear unit), etc., $W_{i,j}$ are the synaptic weights connecting input I_i to the current neuron, I_i is the input of the multilayer perceptron and b is the bias, where $i = 1,...,n$ and $j = 1,...,k$, with n being the number of input features to the NN (corresponding to the number of neurons in the input layer), k the number of neurons in a given hidden layer, and m the number of provided outputs (i.e., number of neurons in the output layer of the NN).

The formula is applied to every neuron in every layer of the network, including the input layer, the hidden layers, and the output layer. Each neuron in a hidden layer takes as its input the output of all the neurons in the previous layer (i.e., assuming a fully connected network), and the output of one neuron in the output layer is the final output of the network. During the training phase, the weights and bias are adjusted to minimize the error between the network output and the desired output. This is done by a learning algorithm, such as the backpropagation algorithm [39], which updates the weights and bias in the direction of the downward gradient [40].

3. Preliminary work

This section includes a brief description of the preliminary work, developed, and published by the authors, constituting the basis of this new proposal. In this previous work, a novel MBSS technique based on the Sample Entropy (SampEn) was developed by the authors, and published in [41]. However, the MBSS technique presented in [41] is the result of some other previous works, also developed by the authors, where the MBSS technique was refined and complemented with modules, implemented in real-time, for diminishing noise introduced by SDR devices, among others. Hence, several comparative studies have been carried out in order to measure the performance of the proposed MBSS technique considering other classical methods like energy detectors, which the efficiency to locate the PUs by this MBSS technique stands out in SNR values close to 0 dB [11]. In order to highlight the contribution of this work, Table 1 is added indicating principal contributions of previous works.

Table 1. Main contributions of previous works.

Previous work	Contributions
A Novel Multiband Spectrum Sensing Method Based on Wavelets and the Higuchi Fractal Dimension [9].	-Two MBSS techniques: wavelets and multiresolution analysis (MRA). Decision rule based on the Higuchi fractal dimension (HFD). -95% effective in detecting PUs for SNR value higher than 0 dB. -Controlled simulated environment. -Blind technique.
Machine Learning Techniques Applied to Multiband Spectrum Sensing in Cognitive Radios [42].	-3 machine learning methods are applied to the MBSS -98% effective in detecting PUs in the spectrum for SNR values greater than 0 dB. -Controlled simulated environment. -Blind technique.
Real-Time Implementation of Multiband Spectrum Sensing Using SDR Technology [11].	-MBSS technique based on MRA, ML and HFD implemented in a real environment of wireless communications. -Real time operation (update every 100 ms). -Implemented with SDR devices. -A module is proposed for the elimination through the detail coefficients obtained with the MRA. -95% efficiency for SNR values greater than 0 dB. -Blind technique.

Previous work	Contributions
Multiband Spectrum Sensing Based on the Sample Entropy [41].	-MBSS technique based on SampEn. -Technique implemented in a real environment of wireless communications. -Technique with real-time operation (updated every 100 ms). -Probability of 0.99 to correctly detect the PU for SNR values greater than 0 dB. -Cooperation of the different SUs to sense a wide range of frequency. -Blind technique.

The last MBSS technique from Table 1 was implemented in a real environment using low-cost SDR devices, where each connected SDR device was considered as a SU, independent from each other processing its information locally in a determined single band, achieving in this way a wideband spectrum sensing as an ensemble. This MBSS technique used in this proposal, however it combines now different SDR devices (namely the RTL-SDR [43], the HackRF One [44] and the LimeSDR mini [45]) to conform each considered SU, forming in this way a CR infrastructure, permitting to have a MBSS in a specific location. Figure 3 shows how this MBSS method works, and its blocks are next described.

- **Sliding window of 100 ms.** Received complex signal from each SDR in the time domain. In this block, the complex signal $x_{Ii,l}(n)+jx_{Qi,l}(n)$ is received and updated every 100 ms, from the radio environment of the *i*-th SU integrated by *z* different SDR devices.
- **Power Spectrum Density (PSD) estimation.** In this module, the Welch method [46] is applied to each signal $x_{Ii,l}(n)+jx_{Qi,l}(n)$ in order to obtain, on a linear scale, the wideband PSD $R_{i,l}(k)$ from the SUs ensemble.
- **Impulsive noise reduction.** In this block, impulsive noise, high frequency noise and abrupt changes (many of them generated by the SDR devices themselves) in the signal $R_{i,l}(k)$ are eliminated or diminished, through discrete wavelets via the multiresolution analysis [47], resulting in the signal $R'_{i,l}(k)$.
- **Estimation of frequency bands and detection of primary users.** In this module, the MBSS technique, based on the SampEn, the K-means algorithm [48] (permitting to optimize certain detection parameters), and discrete wavelets is applied to each SU's analyzed wideband spectrum in order to obtain the spectrum occupation which is given by $Occupation_{T_t}\left\{R'_{i,l}(k)\right\}$. This result contains 3 vectors $\left[b_{i,1}, b_{i,2}, ..., b_{i,N-1}\right]_{T_t}$ which contains binary values indicating occupied ("1") or empty ("0") band, the second vector is $\left[L_{i,1}, L_{i,2}, ..., L_{i,N}\right]_{T_t}$ that contains the corresponding computed boundaries for each detected band and $\left[P_{i,1}, P_{i,2}, ..., P_{i,N-1}\right]_{T_t}$ that contains the power for each detected band.

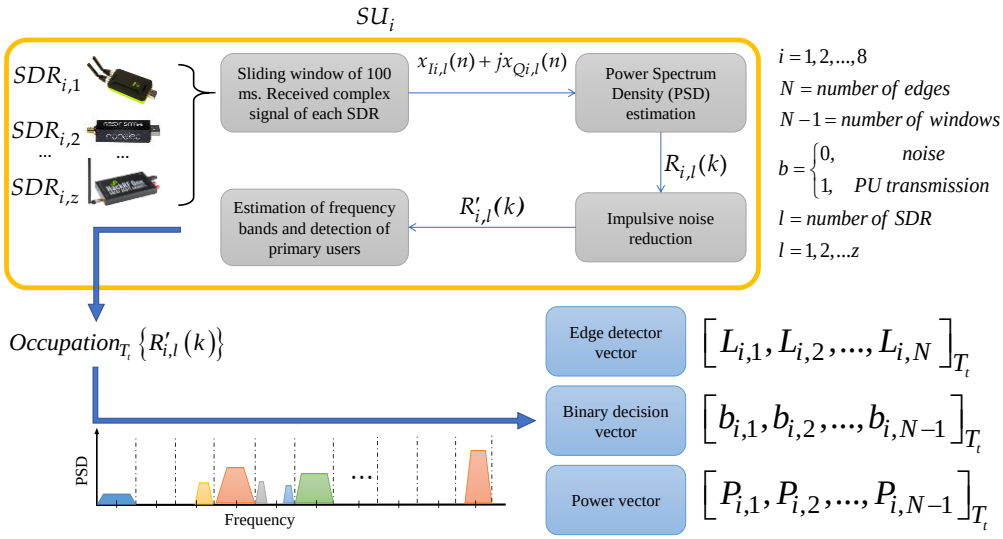

Figure 3. General scheme of implemented MBSS technique [41].

One of the main motivations of this technique is to propose a MBSS method being: (i) adequate to correctly detect a primary user and its operating parameters such as carrier frequency, bandwidth, and power, and (ii) having a computational complexity permitting a real-time implementation. Both conditions are fulfilled. Based on this fact and extending the idea to create now a CRN, as compact as possible, being able to share their spectral information to a central entity, in order to estimate the occupied area by different detected PUs in the studied zone, a wireless communications environment is proposed considering those SDR devices.

4. New proposal and methodology

In this section the main idea of this new research proposal, in addition with implemented methodology, is exposed.

4.1. Proposal

In the previous work, the raised original idea was to consider a computer integrating different SDR devices detecting each PUs in a single band, to sense a wide frequency range as it is shown in Figure 4 a). This can be seen as an only entity containing different SUs. Now, in this work, it is proposed to replicate this entity, containing different connected SDR devices, to create a cooperative CR network sensing a wide frequency range in a broader geographical region. Each entity will be considered as an SU containing different technologies to sense a broad spectrum, as it is shown in Figure 4 b). Besides to sense the radio spectrum and to know the PUs behavior in a certain geographical region, this cooperative CR network has the task of avoiding the problem of hidden terminal. One of the main strengths of this new proposal (see Figure 4 b) over previous work is the cooperation of the different SUs to geographically locate the PUs through the REMs. Furthermore, this processing can be done in real time.

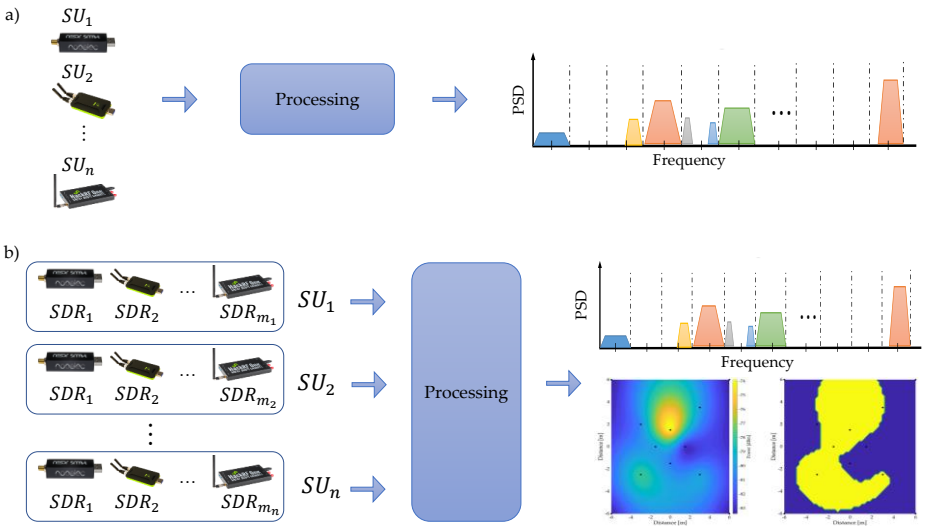

Figure 4. a) Previous work. b) New proposal.

4.2. Methodology

In general, the proposed methodology can be outlined in three big blocks (see Figure 5):

- The collection of information obtained by each SU. For this, each SU processes locally the sensed data to send to a central entity, the occupancy of the observed spectrum in its geographic location including frequency bands edges and estimated powers vectors.
- The database, which stores the information obtained by each SU in a specific time.
- The central entity, which oversees carrying out the processing to determine the geographic area occupied by the detected PUs in the radio spectrum. Each block is described in detail below.

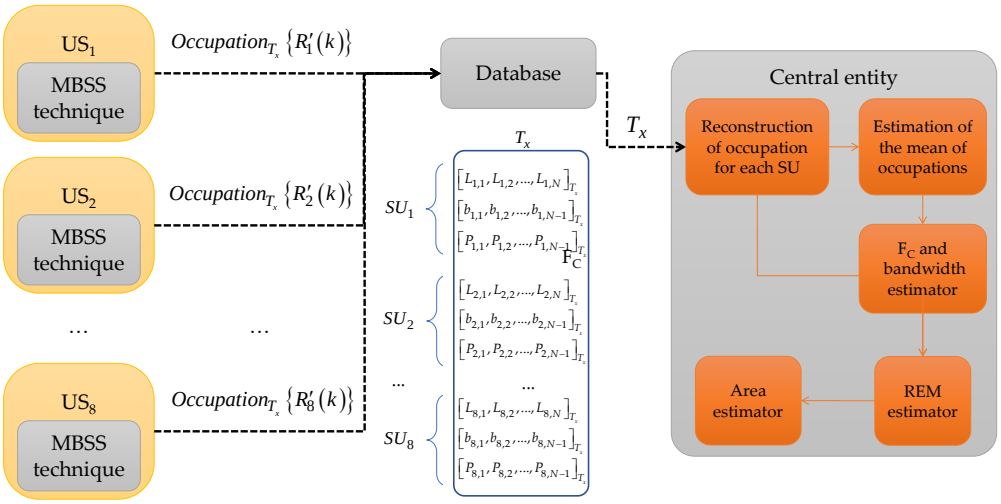

Figure 5. General scheme of the methodology.

4.2.1. Collection of information locally by the SUs

The collection of information is done every 100 ms for each SU, resulting in the $Occupation_{T_t}\left\{R'_{i,l}(k)\right\}$ of the analyzed spectral signal, as it was mentioned before in Section 3 and displayed in Figure 3. This information is conveyed by three vectors: (i) the edges detector vector $\left[L_{i,1}, L_{i,2}, ..., L_{i,N}\right]_{T_t}$, which stores the frequency edges where it is possible to find the presence of PUs, (ii) the binary decision vector $\left[b_{i,1}, b_{i,2}, ..., b_{i,N-1}\right]_{T_t}$, which stores a binary decision for corresponding delimited bands where noise or a possible PU transmission is detected, and (iii) the power vector $\left[P_{i,1}, P_{i,2}, ..., P_{i,N-1}\right]_{T_t}$, which stores the received average power corresponding to each classified window (i.e., each binary decision). In this way, vectors in (ii) and (iii) have the same size.

4.2.2. Database

The edge detector, binary decision and power vectors are stored in the database. It is important to mention that the information stored in the database stems from each SU, is not necessarily similar, i.e., the length of the vectors will be different for each SU. Indeed, each SU observes a different behavior of the radio electric spectrum because they are placed randomly in different geographic coordinates. To alleviate this situation, all vectors are labeled with the exact sensing time T_x (see Table 2), i.e., they are synchronized. All this information is uploaded to a server. In this way, saving only enough relevant information (three different vectors for each SU) in the database, which permits the storage and extraction of information to be as fast as possible; taking advantage, of the hardware of each SU, which performs locally the MBSS technique. Indeed, the three uploaded vectors for each SU, take up very little memory space. In the best of cases (when there is not even a PU in the sensed spectrum) the edge detection vector will have a length of two, while the binary decision vector and the power vector will have a length of one. In the worst case, when the SNR has a low value (close to 0 dB) there will appear multiple windowing due to noise, thus increasing vectors lengths. However, with these 3 vectors, no matters how large they may be, the information will be compact, facilitating the reconstruction of the occupation of the spectrum signal that includes the average power for each reconstructed window.

Table 2. Information that each SU shares in the database at each sensing time T_x.

	T_1	T_2	...	T_x
SU_1	$\left[L_{1,1}, L_{1,2}, ..., L_{1,N}\right]_{T_1}$ $\left[b_{1,1}, b_{1,2}, ..., b_{1,N-1}\right]_{T_1}$ $\left[P_{1,1}, P_{1,2}, ..., P_{1,N-1}\right]_{T_1}$	$\left[L_{1,1}, L_{1,2}, ..., L_{1,N}\right]_{T_2}$ $\left[b_{1,1}, b_{1,2}, ..., b_{1,N-1}\right]_{T_2}$ $\left[P_{1,1}, P_{1,2}, ..., P_{1,N-1}\right]_{T_2}$	...	$\left[L_{1,1}, L_{1,2}, ..., L_{1,N}\right]_{T_x}$ $\left[b_{1,1}, b_{1,2}, ..., b_{1,N-1}\right]_{T_x}$ $\left[P_{1,1}, P_{1,2}, ..., P_{1,N-1}\right]_{T_x}$
SU_2	$\left[L_{2,1}, L_{2,2}, ..., L_{2,N}\right]_{T_1}$ $\left[b_{2,1}, b_{2,2}, ..., b_{2,N-1}\right]_{T_1}$ $\left[P_{2,1}, P_{2,2}, ..., P_{2,N-1}\right]_{T_1}$	$\left[L_{2,1}, L_{2,2}, ..., L_{2,N}\right]_{T_2}$ $\left[b_{2,1}, b_{2,2}, ..., b_{2,N-1}\right]_{T_2}$ $\left[P_{2,1}, P_{2,2}, ..., P_{2,N-1}\right]_{T_2}$	...	$\left[L_{2,1}, L_{2,2}, ..., L_{2,N}\right]_{T_x}$ $\left[b_{2,1}, b_{2,2}, ..., b_{2,N-1}\right]_{T_x}$ $\left[P_{2,1}, P_{2,2}, ..., P_{2,N-1}\right]_{T_x}$
...	...	...	...	...
SU_8	$\left[L_{8,1}, L_{8,2}, ..., L_{8,N}\right]_{T_1}$ $\left[b_{8,1}, b_{8,2}, ..., b_{8,N-1}\right]_{T_1}$ $\left[P_{8,1}, P_{8,2}, ..., P_{8,N-1}\right]_{T_1}$	$\left[L_{8,1}, L_{8,2}, ..., L_{8,N}\right]_{T_2}$ $\left[b_{8,1}, b_{8,2}, ..., b_{8,N-1}\right]_{T_2}$ $\left[P_{8,1}, P_{8,2}, ..., P_{8,N-1}\right]_{T_2}$	...	$\left[L_{8,1}, L_{8,2}, ..., L_{8,N}\right]_{T_x}$ $\left[b_{8,1}, b_{8,2}, ..., b_{8,N-1}\right]_{T_x}$ $\left[P_{8,1}, P_{8,2}, ..., P_{8,N-1}\right]_{T_x}$

4.2.3. Central entity

Main tasks of the central entity are (i) to indicate how many PUs appear in the spectrum to the SUs that are perceiving, (ii) to build a REM for each detected PU taking as parameters the location of each SU, the perceived power in each SU and the time in which the radio space was monitored and, (iii) finally to show the covered area by the PU.

To implement these tasks, two different ways were developed. First one, through *classical* digital processing and the second one, using machine learning techniques (specifically, NN). Both ways are detailed below.

4.2.3.1. Central entity implementing classical digital signal processing

Digital signal processing, which uses programmed algorithms to process and analyze data, is a very mature and stable technology that has been used for decades in a wide variety of applications. It is relatively easy to implement and can be very fast and efficient in situations where the data set and tasks are specific. In this case, the central entity is implemented considering classical techniques for the reconstruction of the spectrum occupation from the shared information (edge detector, binary decision, and power vectors) by each SU in the database. The central entity will process the information as shown in Figure 6 and described by Algorithm 1:

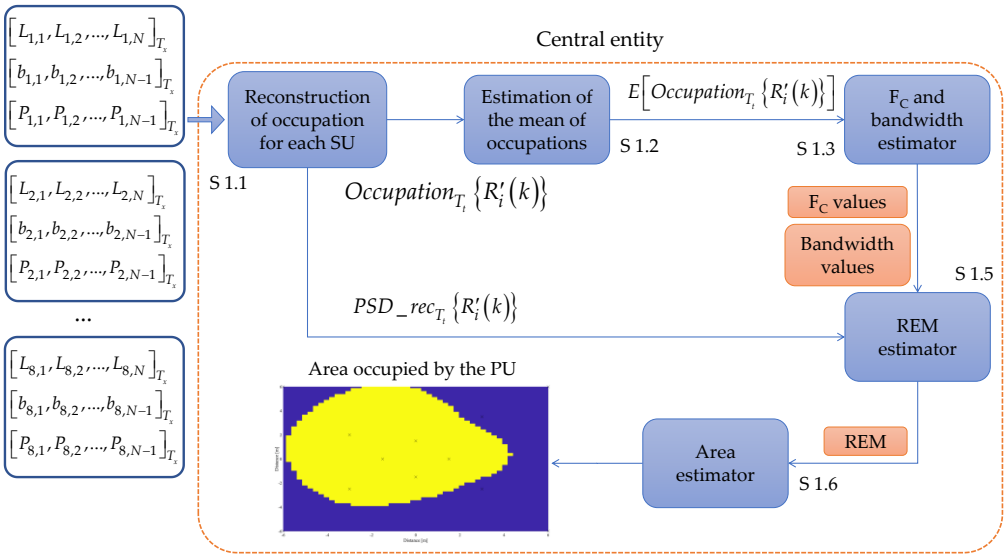

Figure 6. Implementation of the central entity using classical digital signal processing.

Algorithm 1

Step 1.1 The central entity collects the edge detector, binary decision, and power vectors of each PU at determined time instant *t* from the database. With this information, the central entity reconstructs the occupancy signal $Occupation_{T_t}\{R'_i(k)\}$ and constructs simultaneously, an approximation of the power spectral density $PSD\_rec_{T_t}\{R'_i(k)\}$ using only the average powers (power vector) associated to each SU. The length of these frames is set to 1024 samples.

Step 1.2 Once the reconstruction of the occupation of each SU is performed, the mean value $E[Occupation_{T_t}\{R'_i(k)\}]$ is computed.

Step 1.3 After computing the average value of the Occupation, it is possible to infer the presence of one or several PUs, giving two possible vectors. First, the vector $F_{C\_vector} = \left[F_{C1}, F_{C2}, \ldots\right]$ containing the central frequencies values of detected PUs. The length of this vector will be the number of PUs detected by the algorithm. A second vector, containing the occupied bandwidths by each detected PU $B_{vector} = \left[B_{PU_1}, B_{PU_2}, \ldots\right]$, is also obtained. Through the number of singularities and their corresponding widths detected in $E\left[Occupation_{T_i}\left\{R_i'(k)\right\}\right]$, it is possible to estimate how many PUs are in the spectrum and their respective bandwidths ($B_{vector} = \left[B_{PU_1}, B_{PU_2}, \ldots\right]$). The F_{C_vector} is integrated by the central frequency values of estimated bandwidth for each detected PU. This will be shown later (e.g., see Figure 14).

Step 1.4 Knowing now the central frequency and the corresponding bandwidth of each detected PU, it is possible to locate them in each reconstructed PSD $PSD\_rec_{T_i}\left\{R_i'(k)\right\}$. The mean of the interval B_{PU_N} centered on F_{C_N} of the $PSD\_rec_{T_i}\left\{R_i'(k)\right\}$ is computed giving the scalar $aux\_psd_{i,N}$. In the strict sense, this scalar represents the average power in the carrier, sensed by each SU, where the PU_N is or should be.

Step 1.5 With the scalar $aux\_psd_{i,N}$ of each SU located in a specific coordinate, the REM will be built using a double interpolation. First interpolation is done through the IDW method and for the second one, the Kriging method is applied. Due to the fact that in our case only eight geographical points are considered (i.e., only eight SUs are implemented), this double interpolation is carried out with the purpose of having a better precision to describe the behavior of the radio electric space in the environment described later. In this case, each REM is constructed with values $aux\_psd_{i,N}$ that specifically correspond to the average power of the bandwidth B_{PU_N}.

Step 1.6 Finally, the *active area* of each PU will be determined according to the information collected by each secondary entity SU_i. For this, a threshold of -80 dBm is used to classify the area estimated by the REMs that was chosen in [42] for a wireless environment. That is, regions in the REM that are above this threshold correspond to the *active area* of detected PUs.

4.2.3.2. Central entity implementing neural networks

In this section, the central entity is implemented considering ML techniques, specifically NNs. NNs have the ability to learn from data and improve their performance as more information is presented to them. This makes them particularly useful in applications where the data are complex or difficult to understand and process using traditional methods. Furthermore, NNs can perform tasks that traditional methods cannot, such as recognizing patterns in unstructured data or processing input signals that change over time. In this case, some modules in the central entity are replaced by NNs as it is shown in Figure 7. Steps of its operation (Algorithm 2) are described below.

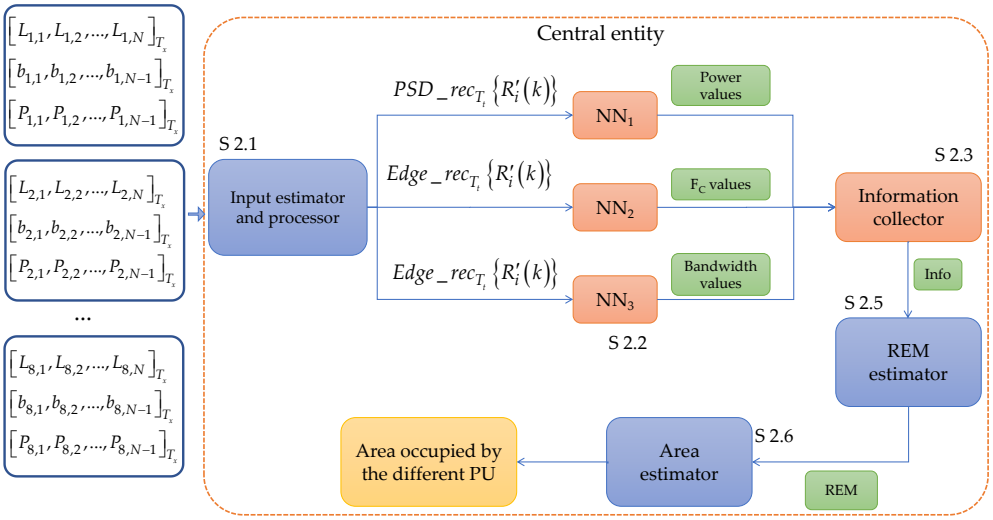

Figure 7. Central entity that determines the parameters for the construction of REM and area maps through NNs.

Algorithm 2

Step 2.1 The central entity collects the data from each of the SUs in the database at a specific time T_x. In the module *input estimator and processor*, a processing is carried out in order to estimate the inputs driving a NNs ensemble. As mentioned above, the three vectors coming from each SU for time T_x do not have the same length as the vectors for time T_{x+t_1} or T_{x-t_2}. Even, the size of the vectors differs from one SU to another SU even though they are at the same time T_x. Based on this fact, this module oversees building the input vectors $PSD\_rec_{T_t}\left\{R_i'(k)\right\}$ and $Edge\_rec_{T_t}\left\{R_i'(k)\right\}$ driving the NNs. For this, these vectors must have always the same length (in our case this length is set to 13 samples). This block is detailed below.

Step 2.2 The vector $PSD\_rec_{T_t}\left\{R_i'(k)\right\}$ is evaluated by NN_1, at the same time the vector $Edge\_rec_{T_t}\left\{R_i'(k)\right\}$ is evaluated by NN_2 and by NN_3. This evaluation is performed sequentially, i.e., each vector of each SU will be evaluated by its corresponding NN one after the other until the result of the *i*-th connected SU be obtained. As a result of this step, NN_1 provides an approximate number of detected PUs and their powers. NN_2 gives an approximate number of detected PUs and their central frequencies. Finally, NN_3 returns an approximate number of detected PUs and their bandwidths. At the end of this section, it is discussed what would happen if the number of PUs detected by each NN is not the same.

Step 2.3 The information obtained from Step 2.2 is evaluated to determine the number of PUs in the spectrum and their corresponding power, bandwidth, and carrier frequency. This evaluation is detailed below.

Step 2.4 The information obtained from Step 2.3 will be shared with the REM estimator module. This block receives (i) the geographic coordinates of the SUs in the network, (ii) the data of the possible PUs in the spectrum (power, carrier, and

bandwidth) and (iii) an identifier that corresponds to time T_x in which the spectrum was monitored.

The rest of the steps of this algorithm that includes NNs correspond to Steps 1.5 and 1.6 of the methodology mentioned in the Algorithm 1 (Section 4.3.1). Next, the *input estimator and processor* (Step 2.1) and the *information collector* (Step 2.3) modules are described in detail.

Figure 8 shows the process to generate $PSD\_rec_{T_i}\left\{R_i'(k)\right\}$, the input vector of NN_1. For this, firstly the power, edge detector and binary decision vectors are used. These vectors are ranging in the intervals of [1-20], [1-20] and [2-21] samples respectively, according to the studies carried out in the preliminary work [41]. Through these vectors, as it was mentioned at the beginning of this section, is known (i) the frequency range that was sensed by each SU, (ii) the number of samples of the estimated power spectral density and (iii) the number of windows in which the sensed spectrum was partitioned. For each detected window an occupation value is assigned, being 1 for a possible PU transmission or 0 for the noise. Accompanying each occupation value is an average power value (power vector element) being the power estimate in each detected window. In this way the occupation and power signal are reconstructed with these average values of each detected window, setting both vectors to a length of 1024 samples each.

After that, the power and the occupation reconstructed vectors are multiplied column by column, resulting in the $Power\_Occupation$ vector of 1024 samples too. In this last vector, locations where noise is estimated will have a value of 0 and, where there is a possible transmission of one or several PUs, they will have a value corresponding to the average power of detected windows. A downsampling of the $Power\_Occupation$ vector is done to keep only 10 samples of this. This downsampled vector is finally concatenated with the coordinates of the SUs (to which the analyzed vectors correspond) and the time T_x in which the signal $Occupation_{T_i}\left\{R_{i,l}'(k)\right\}$ was sensed, integrating in this form the input vector for NN_1. It is important to mention that the time parameter is extremely important since the spectrum behaves differently over time.

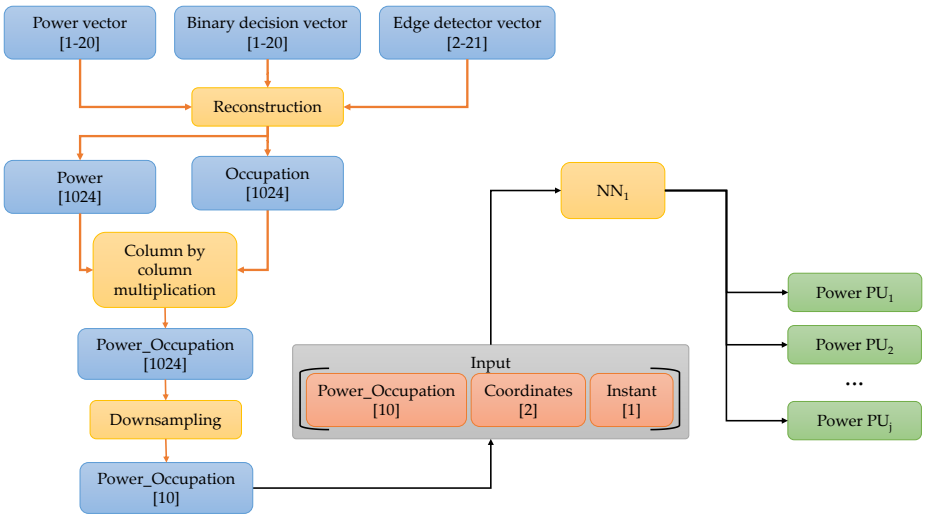

Figure 8. $PSD\_rec_{T_i}\left\{R_i'(k)\right\}$ vector construction.

Figure 9 shows the construction process of the vector $Edge\_rec_{T_i}\{R_i'(k)\}$. Here, the occupancy vector and the frequency bands vector corresponding to the frequency interval that is perceived by the corresponding SU, are reconstructed. This vector is also multiplied column by column with the occupation vector, resulting in the $Freq\_Occupation$ vector of 1024 samples of length. In this way, this latter vector contains zeros where it corresponds to noise and have non-zero values in the samples that correspond to one or several transmissions of the PUs. Again, a downsampling is also applied to reduce this vector to only 10 samples and it is concatenated with the coordinates of the SUs and the time T_x corresponding to the sensing period. This concatenation drives the input of the neural networks NN_2 and NN_3.

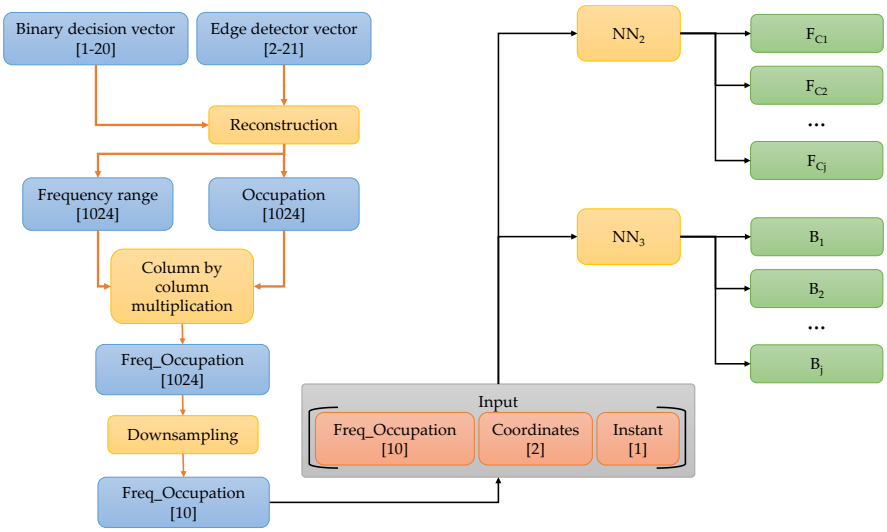

Figure 9. $Edge\_rec_{T_i}\{R_i'(k)\}$ vector construction.

The NN_1 output corresponds to the power of each PU detected by each SU. In this case, if the value of the power of the PU is less than -80 dBm, then it is interpreted that there is no transmission, and it corresponds to noise. The output of NN_2 results in the carrier on which a possible PU has been located. Finally, the NN_3 output delivers the transmission bandwidths corresponding to each detected PU in the analyzed spectrum.

The resulting outputs from NNs, for each SU are shown in Figure 10. These outputs can be classed in three large blocks: powers, carrier's frequencies, and transmission bandwidths of each detected PU by the corresponding SU. The *information collector* block is in charge of analyzing the output for each SU making up the network and sending it to the module permitting to estimate the REM. For example, in Figure 10 a) it is shown that the power of PU_1 is less than -80 dBm, in this case the *information collector* interprets that PU_1 does not correspond to a transmission and by sharing the information delivered by SU_1 with the estimator of REM, assumes that SU_1 observes $(j-1)$ PUs (where j represents the number of possible PU transmissions). Figure 10 b) shows that the carrier frequency of PU_1 and PU_j are outside the space being monitored. So, the *information collector* shares with the REM estimator that SU_2 only observes $(j-2)$ PUs. In Figure 10 c) it is shown that the B of PU_1 is a very small value (this could correspond to impulse noise). Then, the *information collector* shares with the REM estimator module that the SU_n observes $(j-1)$ PUs. It is important to mention that any of these combinations shown in Figure 10, will change the number of observed PUs, i.e., if the power of a first

PU does not exceed the threshold of -80 dBm and the carrier of a second PU is not in a correct frequency range and the B of a third PU tends to zero, then these 3 PUs will be considered as noise.

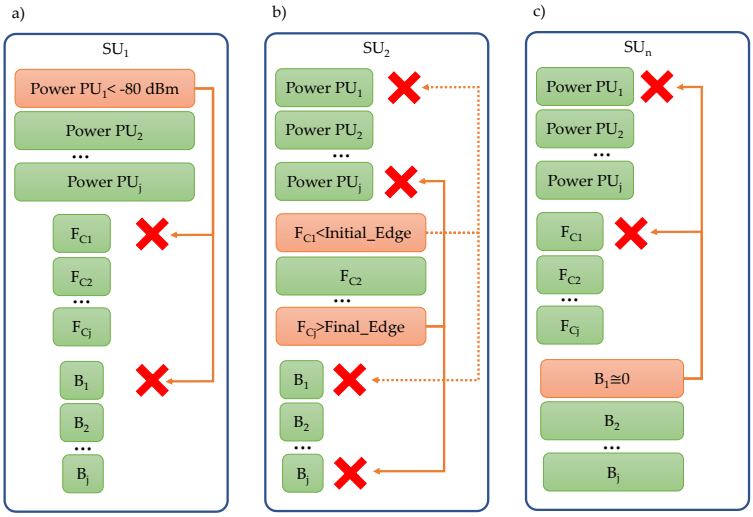

Figure 10. a). the SU_1 observes j PUs, but PU_1 is discarded because it does not meet the threshold. b) SU_2 observes j PUs, nevertheless PU_1 and PU_j are discarded because they are not in the correct frequency range. c) the SU_n observes j PUs, the PU_1 is discarded for not having a large enough bandwidth to be considered as a transmission.

The proposed methodology has been implemented in a real wireless communication environment. This controlled environment is explained in next section.

5. Real wireless communication environment

Figure 11 shows the real controlled environment implemented in this research work. In this proposed scenario, we consider two PUs co-located in the center of the studied area and, at the same time eight SUs are sensing the behavior of these PUs in their influence geographical zone. It is important to mention that SUs and PUs are not sharing information about geographic coordinates among them, and neither that information is used in spectrum sensing. PUs are located in the center looking that most SUs could receive part of the PUs signal. SUs were set randomly in the area of study.

Table 3 specifies involved parameters for both SUs and PUs. These SUs share the channel occupancy to the database and the central entity in order to determine: (i) how many PUs on average are observed in this environment, (ii) the B and the F_C in which they are located the detected PUs, and (iii) finally the area that is occupied by these detected PUs.

PUs and SUs are deployed in an area of $12\times12\text{m}^2$. In this area there exists structures such as walls, doors, windows, columns, etc., as it is indicated in Figure 11.

Table 3. Main parameters of PUs and SUs.

Label	Device	Fc Tx [MHz]	Fc Rx [MHz]	Bandwidth [MHz]	Location coordinate (X,Y) [m]

Label	Device	Fc Tx [MHz]	Fc Rx [MHz]	Bandwidth [MHz]	Location coordinate (X,Y) [m]
PU_1	Mini LimeSDR	699.5	-	0.5	(0, 0)
PU_2	HackRF ONE	700.5	-	1	(0, 0)
SU_1	RTL-SDR	-	700	2.4	(-1.5, 0)
SU_2	RTL-SDR	-	700	2.4	(0, 1.5)
SU_3	RTL-SDR	-	700	2.4	(1.5, 0)
SU_4	RTL-SDR	-	700	2.4	(0, -1.5)
SU_5	RTL-SDR	-	700	2.4	(-3, 2)
SU_6	RTL-SDR	-	700	2.4	(3, 3.5)
SU_7	RTL-SDR	-	700	2.4	(3, -2.5)
SU_8	RTL-SDR	-	700	2.4	(-3, -2.5)

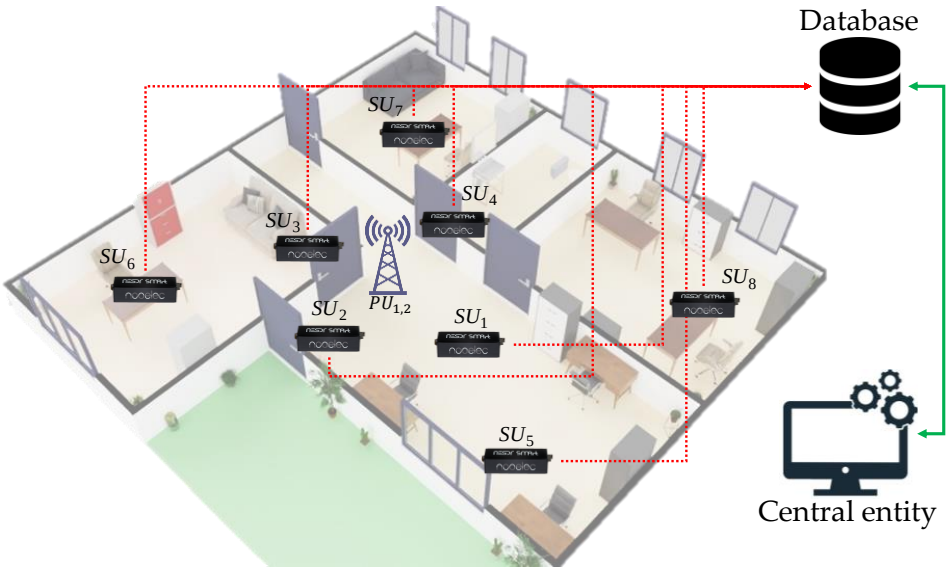

Figure 11. The real implemented scenario.

5. Results

In this section, the results of implementing the proposed methodology under a real environment of wireless communication are shown. They were divided in two sub-sections, the first one corresponds to the central entity based on digital signal processing and the second is for the central entity based on NNs.

5.1. Results with a central entity based on digital signal processing

This section shows the results obtained by the implemented scenario presented in Figure 11. Figure 12 shows the reconstruction of the occupation of each SU by the central entity. In this case, we can highlight the next important points:

- There are SUs who fail to perceive both SUs.
- The central frequency F_C of each PU is perceived at a different frequency by each SU.
- The bandwidth size B for each PU varies for each SU.
- SUs that are farthest from the PUs, fail to detect both PUs.

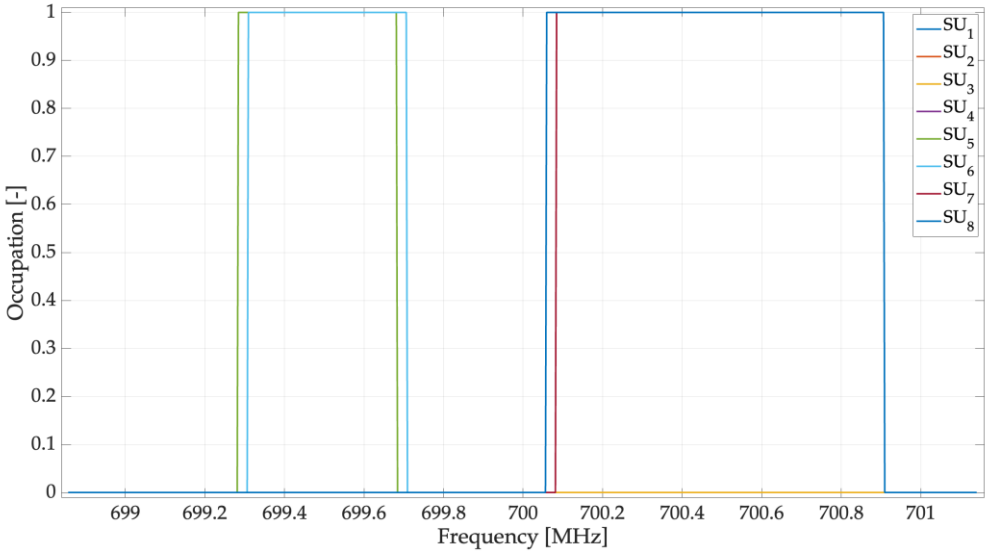

Figure 12. Spectrum occupancy for each SU.

Figure 13 shows the result of constructing an approximation of the PSD for each SU considering only mean values of power for each section of the spectrum. These approximations will be used for the estimation of the REM.

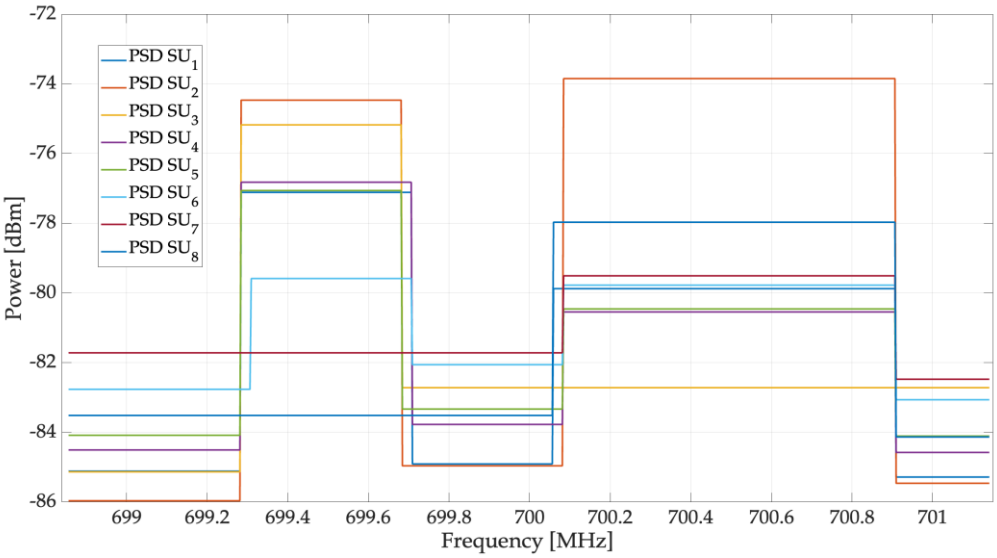

Figure 13. Approximated PSD for each SU.

Figure 14 shows the result of applying the module that estimates the average occupancy of each SU. Also, in that figure appears, the result of the estimation of the F_C and B of each PU. In this case, the first value obtained was $F_{C1} = 699.48$ MHz with bandwidth of $B_1 = 0.4$ MHz. Also, it was obtained $F_{C2} = 700.49$ MHz with bandwidth of $B_2 = 0.825$ MHz. Given that the exact value for $F_{C1} = 699.5$ MHz, $B_1 = 0.5$ MHz,

$F_{C2} = 700.5$ MHz and $B_2 = 1$ MHz. The values obtained show a good estimation of the occupation via the MBSS method in conjunction with the central entity for a specific geographical zone covered by the PUs.

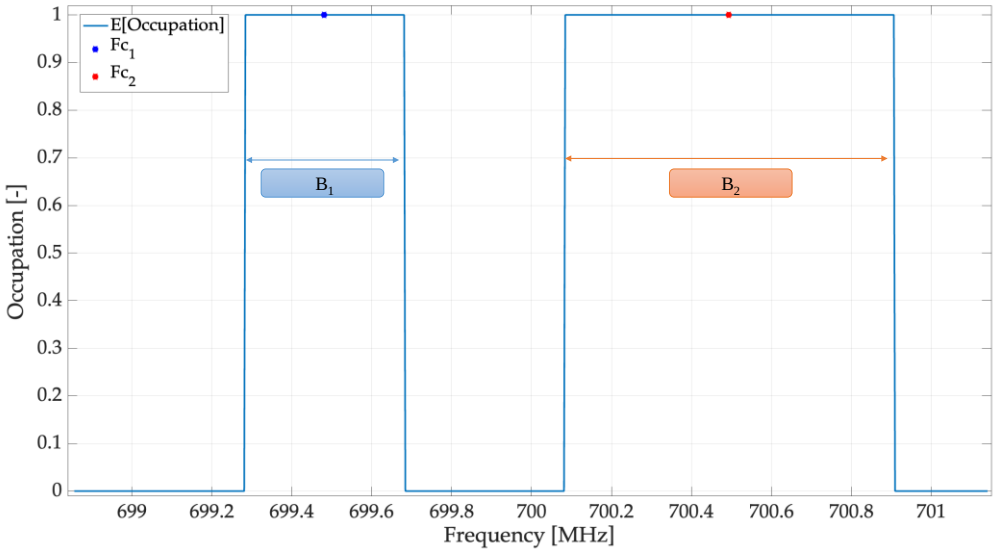

Figure 14. Average of occupancy obtained by the central entity.

Figures 15 a) and 16 a) show the REM of PU_1 and PU_2 respectively. This map is created based on the information collected by the SUs in their different locations. Figures 15 b) and 16 b) show the respective areas occupied by PU_1 and PU_2. Values of the occupied area for each PU are $area_{PU_1} = 60.76\,\mathrm{m}^2$ and $area_{PU_2} = 56.39\,\mathrm{m}^2$. This result is obtained by placing a threshold $L = -80\,\mathrm{dBm}$ in the obtained REM. In this way, the area having a power greater than this threshold is considered as an occupied space by the PU.

Figure 15 b) shows that SU_6 and SU_8 do not observe the transmission of PU_1 (i.e., both SU_6 and SU_8 do not appear in this area of influence). This effect occurs due to the structural distribution in which the implementation of the real scenario was made. Nevertheless, an expected collaboration between SUs could improve this result.

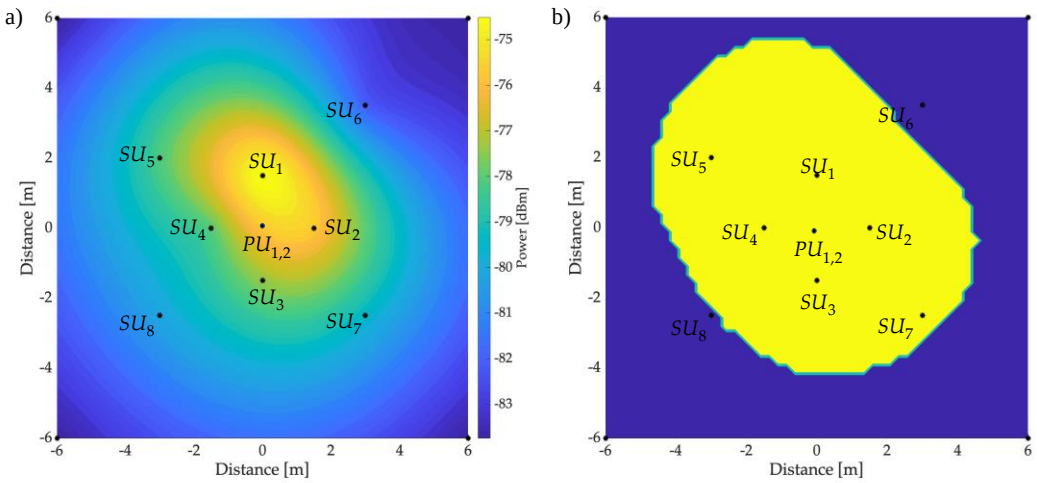

Figure 15. Primary transmission in the 699.48 MHz band with a B of 0.4 MHz. a) REM, b) Occupied area.

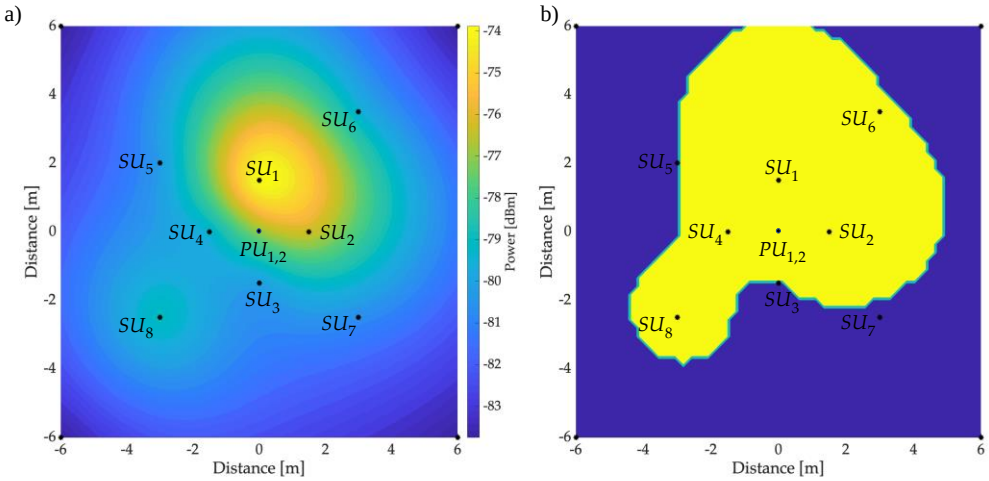

Figure 16. Primary transmission in the 700.49 MHz band with a B of 0.825 MHz. a) REM, b) Occupied area.

5.2. Results with a central entity based on NNs

This section presents in detail the results of using a central entity that considers NNs. First, the details of the training of the NNs will be presented and then the results obtained from applying ML algorithms to this stage will be shown.

5.2.1. Training

For the training stage of NN_1, NN_2 and NN_3, the backpropagation algorithm, the Levenberg-Marquardt activation function, 1000 epochs, the mean square error as loss function and, a low learning rate, were used. Besides, 9,000 vectors of inputs and their corresponding outputs were used. It is common to ask, which is the best NN? How many layers should it be used? Or how many neurons? To answer these questions, 12 NNs were studied to carry out the work of NN_1, 12 NNs to carry out the work of NN_2, and 12 NNs that carried out the work of NN_3. For this, each NN is considered with the number of layers $la = 1,2,3,4$ and with the number of neurons $ne = 16,32,64$. There will be for each NN the 12 possible combinations between neurons and layers. This procedure was done to know which is the most convenient NN configuration for this work.

Due to these configurations, the results obtained for both the entity that uses digital signal processing and the one that uses the three NNs are presented. As the first parameter, the training time used by each NN will be mentioned. Table 4 shows the time spent by each NN for the training stage. Besides, in Figure 17, it can be seen that the more neurons and/or layers an NN has, the training time will be longer. All the processing was done with the same computer (MacBook Pro with 8 GB RAM and a 1st generation M1 processor), the training time for each NN in its different versions tends to have a very similar behavior.

Table 4. Training time of a) NN_1; b) NN_2; c) NN_3.

Training time NN1 [min]				
a)		**Neurons**		
		16	**32**	**64**
Layers	**1**	2	10	60
	2	10	25	180
	3	120	192	1500
	4	162	600	3984

Training time NN2 [min]				
b)		**Neurons**		
		16	**32**	**64**
Layers	**1**	2	6	16
	2	8	33	290
	3	15	80	850
	4	20	129	7610

Training time NN3 [min]				
c)		**Neurons**		
		16	**32**	**64**
Layers	**1**	2	5	16
	2	5	19	280
	3	11	75	875
	4	21	145	3628

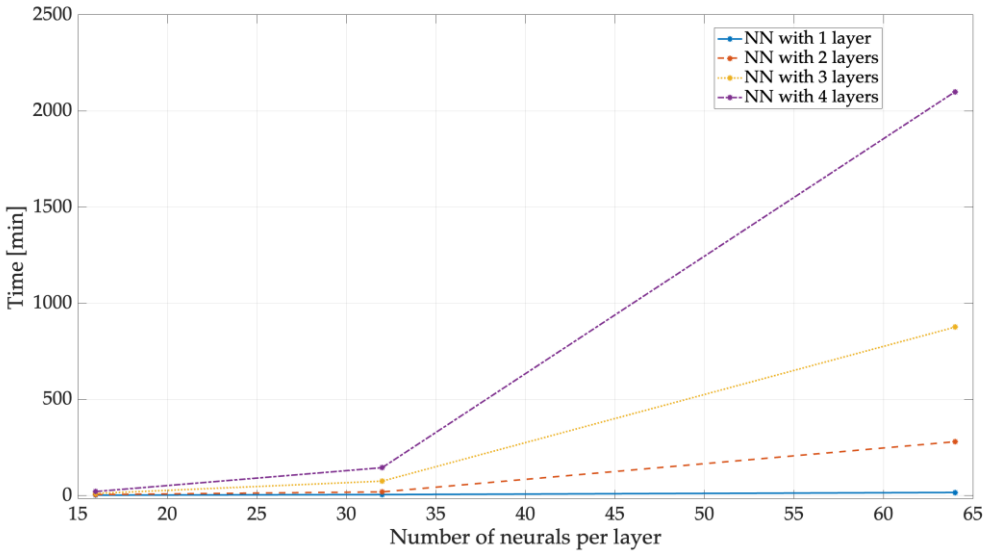

Figure 17. Training time for the different versions of the NN_3.

The statistical results for both proposals are mentioned below.

5.2.1. Statistics

In this section, the statistical results of the three NN used in this proposal are analyzed. To obtain these results, 27,000 entries were considered for each NN. Figure 18 a) shows the results of NN_1, which oversees granting the power value of the detected PUs. This figure shows the difference in power between the real values and those obtained with the NN_1 in its different versions. The difference between the expected real value and the one obtained by the NN_1 on average is -0.01 dBm, this for each version of NN_1, which highlights the accuracy of the predicted power with the NN-based approach. In Figure 18 b) only the average value is shown, indicating that the margin of error between the expected value and that given by the different NNs is very close, practically the same.

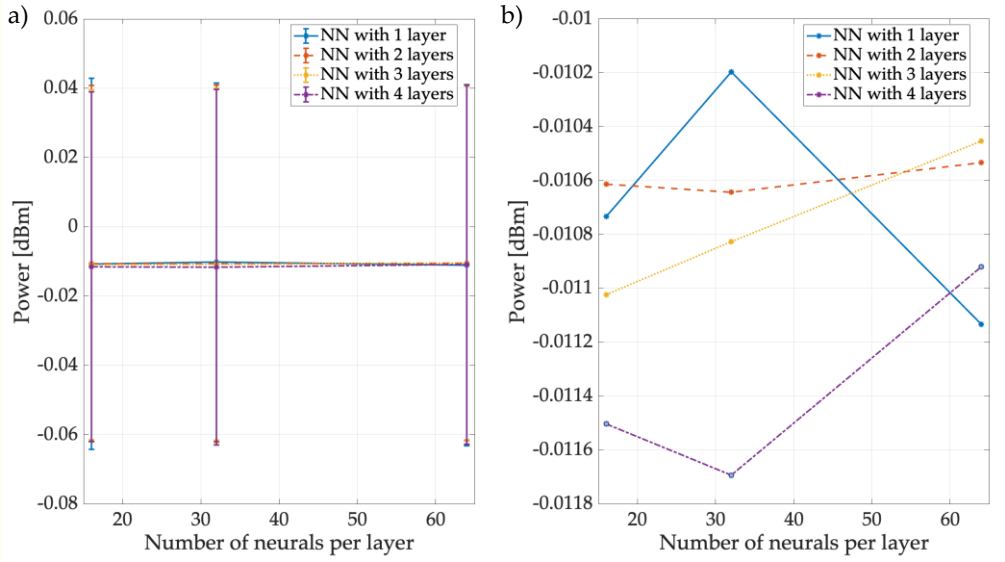

Figure 18. Difference between real values and those obtained by NN_1 a) mean and standard deviation (STD), b) mean.

Figure 19 shows the result of NN_2, which is the network that gives the carrier value of each PU detected. Due to the proposed environment, the NN_2 detected two PUs. The precision to detect the carrier of each of them is shown in that image, where F_{C1} is on average 699.4 MHz, regardless of the configuration of NN_2 (see Figure 19 a)). Figure 19 b) shows the precision of NN_2 for detecting the PU_2 carrier. Here it is observed that on average the carrier is 700.494 MHz and even though the network with 2 layers and 64 neurons per layer slightly deviates from the ideal value (700.5 MHz) compared to the other NN configurations, it continues to provide a reasonably accurate result. In both images, the value of applying digital signal processing can be seen in blue, which is quite close to the ideal value, 699.4934 MHz and 700.4935 MHz for PU_1 and PU_2 respectively, thus resulting in a comparable level of accuracy as the NN-based approach.

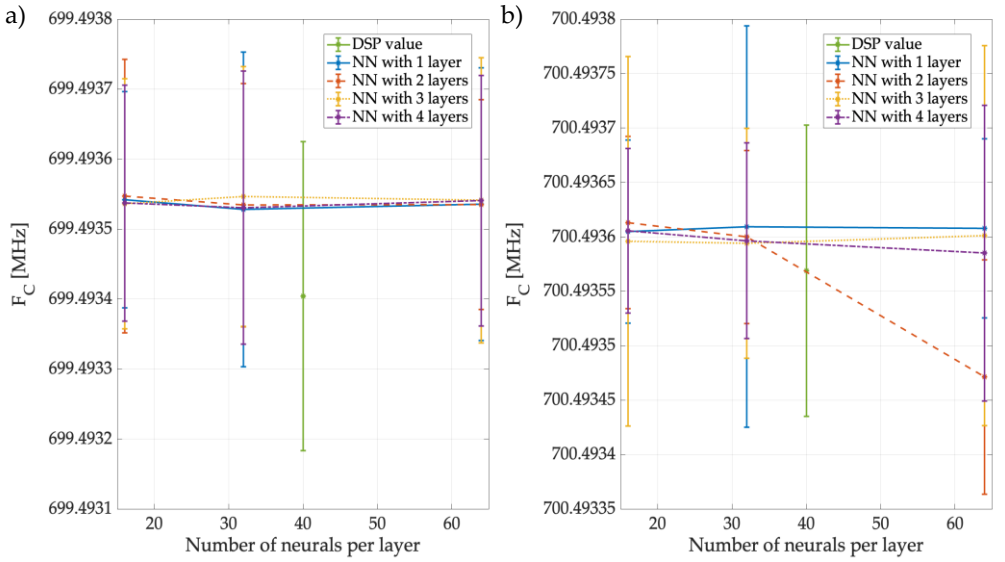

Figure 19. Mean and STD of the precision to detect a) the PU_1 Carrier of the different NN_2 variants. b) the PU_2 carrier of the different networks.

Figure 20 shows the results of NN_3, which is the network in charge of giving the B that each detected PU in the spectrum frames. In this figure, the average B for PU_1 is 0.48 MHz, regardless of the network configuration (see Figure 20 a)). Besides, Figure 20 b) shows that notwithstanding the NN structure, the value of the B for PU_2 is practically 1 MHz, which perfectly coincides with the ideal value (see Table 3). This image also shows by a blue dot the result of using digital signal processing. Giving a result of $B_1 = 0.425$ MHz and $B_2 = 0.824$ MHz. For B_1 the level of error/precision of DSP and NN is quite similar (only that DSP overestimates while NN underestimates) while for B_2 it seems that NN is much more precise, so it could be indicated that the method based on NN tends to improve DSP or, in the worst case, provides a comparable absolute error.

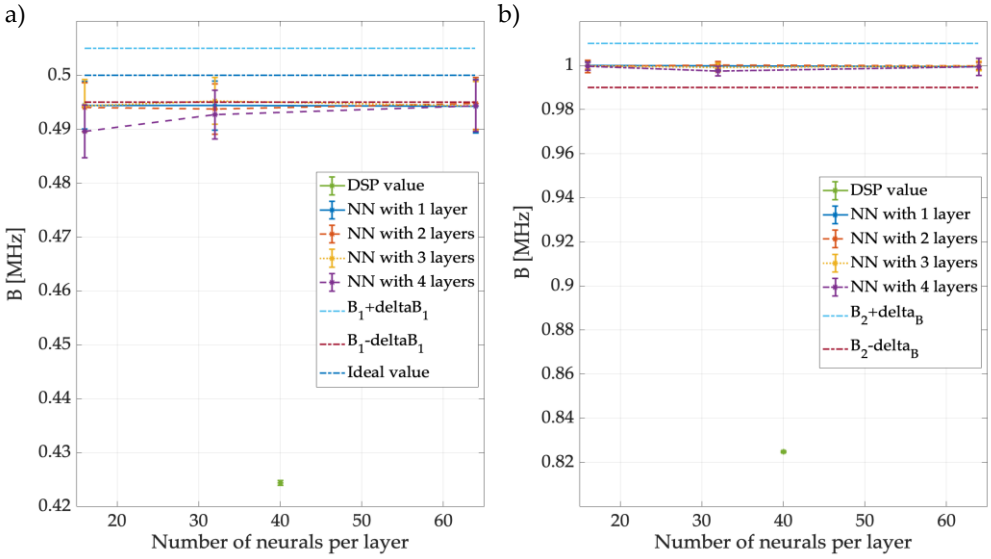

Figure 20. Mean and STD of the precision to detect a) the B_1 of the different NN_3 variants. b) the B_2 of the different networks.

In this case, to measure the precision of this methodology, the F1-score ($F1$) [49] will be used, which is a commonly used evaluation metric in the ML field to assess the accuracy of a binary classification model. This metric combines the accuracy and recall of the model into a single measure.

Precision refers to the ratio of true positives (TP) to the sum of true positives and false positives (FP), while recall, on the other hand, refers to the ratio of true positives to the sum of true positives and false negatives (FN). The value of $F1$ is the harmonic mean of precision and completeness, which means that it gives more weight to low values. The $F1$ formula is as follows:

$$F1 = 2\frac{(precision * recall)}{(precision + recall)} \quad (14)$$

A value of $F1$ equal to 1 indicates that the accuracy and completeness are perfect, while an $F1$ of 0 indicates that the model is unable to correctly classify any of the samples. The $F1$ is a valuable metric for comparing different binary classification models and selecting the best model for a given classification task. To determine this parameter (i.e., $F1$), the next four possible cases are considered (see Figure 21):

1. A window that corresponds to a PU transmission and that SU classifies as a PU transmission is a true positive (TP) value.
2. A frequency window that corresponds to a transmission of the PU and that SU classifies as noise is a false negative (FN) value.
3. A window that corresponds to noise and that SU classifies as a PU transmission is a false positive (FP) value.
4. A frequency window that corresponds to noise and that SU classifies as noise is a true negative (TN) value.

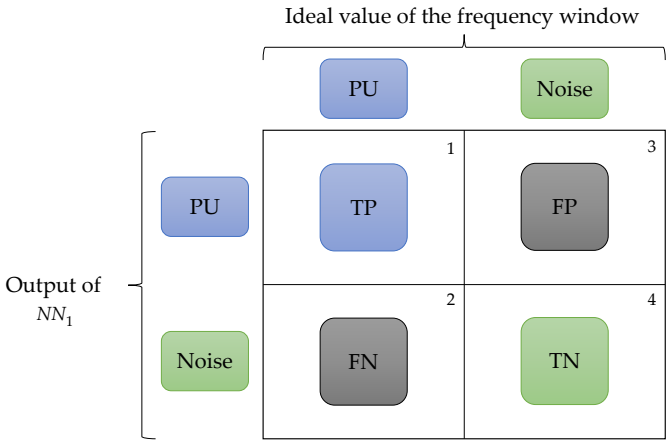

Figure 21. Evaluation outputs of detected windows.

Figure 22 shows the $F1$ of the NN_1, which indicates whether the PUs were correctly located. In this image, all the NNs have a nearly perfect performance around 0.98.

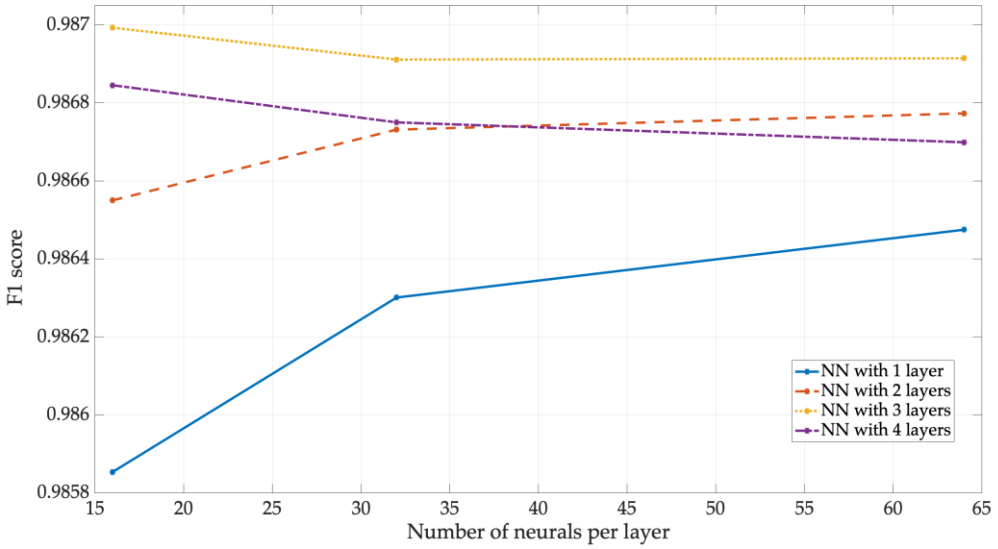

Figure 22. F1 score for NN1.

In the case of B, the $F1$ was estimated from the following cases (see Figure 23).

- The resulting NN_3 B that matches the ideal bandwidth that corresponds to a transmission will be a true positive (TP) case.
- The resulting NN_3 B that corresponds to an ideal bandwidth close to zero will be a true negative (TN) case.

- The resulting NN_3 B that is much greater than an ideal bandwidth that is close to zero will be a false positive (FP) case.
- The resulting NN_3 B that is a value close to zero but should correspond to an ideal transmission B will be a false negative (FN) case.

Figure 23 shows a ΔB value, which is the difference between real and estimated bandwidth, when there is a PU. When this parameter tends to grow, it provides flexibility so that the resulting NN_3 B matches an ideal B that corresponds to one transmission. However, when the value of ΔB is very small, the system becomes more inaccurate since it only detects as TP those values that are close to the ideal value.

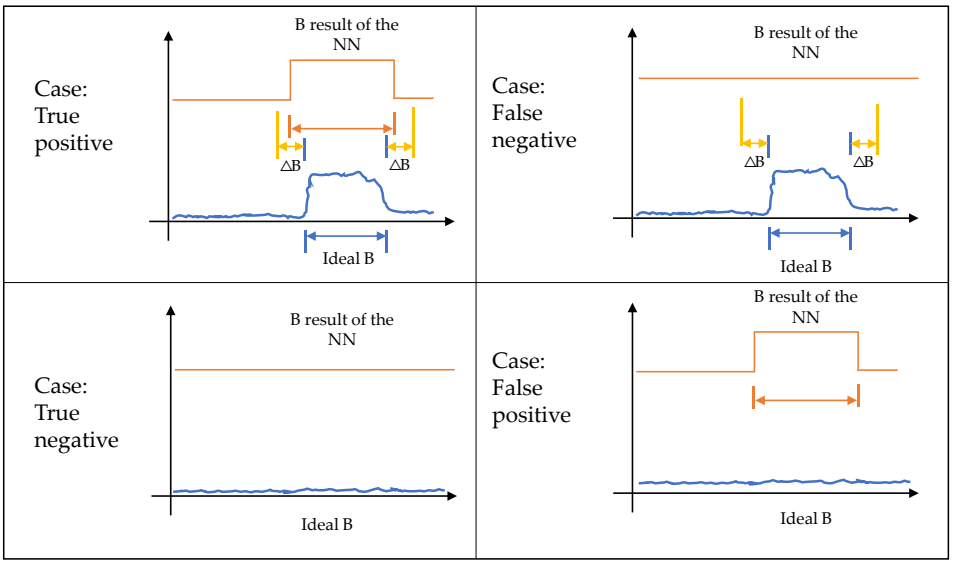

Figure 23. B assessment results.

Figure 24 shows the $F1$ of NN_3. In this image, the NN with four layers and the NN with 2 layers have an undesirable performance. The reason for this poor performance can be explained observing Figure 20, in which the result of B_1 is very close to the minimum value ($ideal\_value - \Delta B$). Nevertheless, this undesirable performance is observed only when a low number of neurons per layer (i.e., 16) is employed. By properly configuring the number of neurons per layer to a sufficiently high value (e.g., 32 or greater), an accurate performance with $F1$ values around 0.95-0.96 is obtained also with 2 and 4 layers in the NN. From this Figure can also be concluded that in the NNs with 2 and 4 layers there is an overfitting. This is due the model for the 2-layer NN and 4-layer NN, both with 16 neurons per layer, fit too well to the training data and as a result does not generalize well to the new data. Despite this, they have an $F1$ of 0.9 and 0.79 respectively.

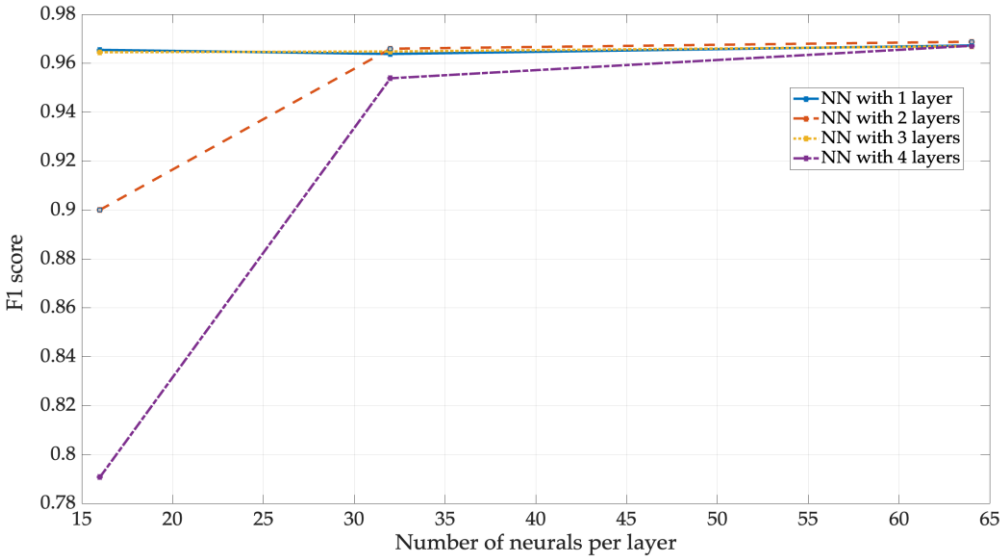

Figure 24. $F1$ of NN_3.

Figure 25 shows the ratio of $F1$ score between training time for NN_1 and NN_3 (Figure 25 a) and b), respectively). The value obtained from this relationship indicates which NN provides the best trade-off based on the F1 accuracy result obtained and the time invested in training the network in order to attain such level of accuracy. As it can be noticed, using a NN with a single layer provides the best F1 performance for training time required to obtain it. Further increasing the number of layers will increase the F1 score performance but will proportionately require a much longer training time, thus leading to a worse trade-off or relation between benefit (represented by the F1 performance) and cost (represented by the required training time).

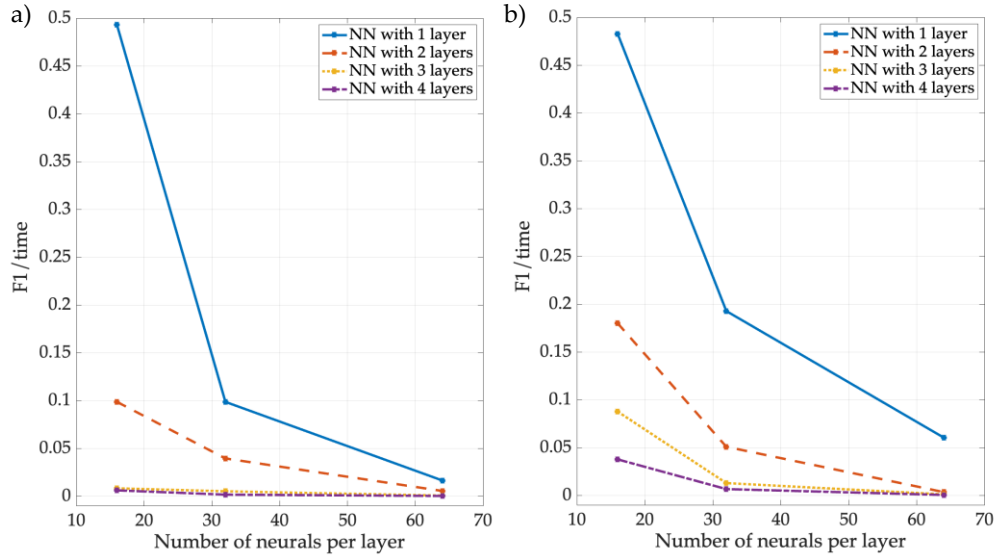

Figure 25. $F1$ and training time ratio of a) NN_1 and b) NN_3.

Finally, Figure 26 shows the behavior of the proposed methodology along time, in which the sensed spectrum is analyzed in a specific geographic area, where SUs collaborate in order to obtain the REM of the different detected PUs. As it can be appreciated, the proposed methodology can characterize the dynamic temporal evolution of the REM in the geographical area of interest, thus providing a valuable tool for the study of CRNs.

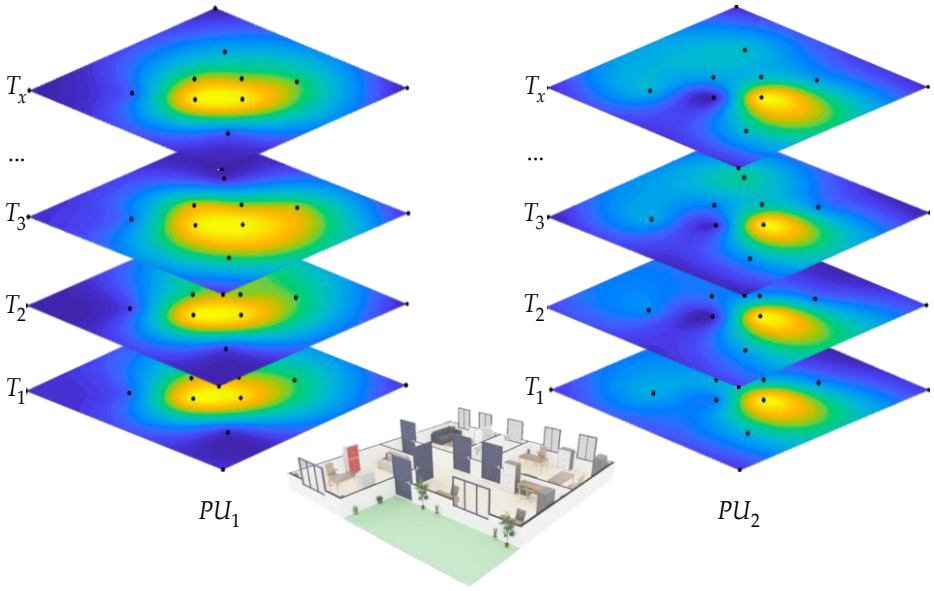

Figure 26. Result of the central entity.

6. Conclusions

In this work, a real wireless communication scenario is implemented to detect occupancy of multiple PUs through several SUs via a central entity, permitting the determination of the area that is being used by the PUs based on REMs estimations. Besides REMs constructions, features (power, bandwidth, central frequency) of possible detected PUs from the multiband spectrum frames are estimated by the considered SUs.

For this task, it is proposed to use neural networks in order to substitute the classical digital signal processing used in preliminary work. With this, it is expected that the performance and processing time will be faster. Clearly, the training stage of the NNs as it was shown in the results section, is a factor to be considered. In the specific case of this work, better precision is preferred to locate the PU and less processing time for the central entity when using the NNs. However, it must be considered that NNs training time is not negligible and can even be high depending on the layers and neurons that each NN contains. The difference between the real values and that of the NNs is very close and it could even be said that they are practically the same and even in some cases, the NN shows better results than the digital signal processing, as it is the case when detecting the carrier of the PUs.

Finally, neural networks are a very powerful and useful tool in many applications, but they are not always the best option. In some cases, classical digital signal processing may be sufficient, while in others, neural networks can provide a significant improvement in performance and accuracy. In future work, it is possible to determine the optimal number of SUs to obtain the area that is occupied by the PU. Based on this, excessive processing of the central entity is avoided.

Author Contributions: All authors conceptualized the original idea of how to use SDR devices and designed the proposed methodology. Y.M.-T. improved these proposals by adding some ideas on how to obtain the data and how to develop the necessary software. Y.M.-T. also performed all the experiments of this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding from the Mexican National Council of Science and Technology (CONACyT), Grant (no. 490180). Institutional Review Board Statement: Not applicable.
Acknowledgments: The authors are grateful to CONACYT, the Department of Electrical Engineering of UAM-Iztapalapa and Department of Electrical Engineering and Electronics of University of Liverpool, for providing the necessary support for this work.
Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Cognitive radio	CR
Primary user	PU
Secondary user	SU
Spectrum sensing	SS
Multiband spectrum sensing	MBSS
Machine learning	ML
Software defined radio	SDR
Universal software radio peripheral	USRP
Radio environment map	REM
Cognitive radio network	CRN
Inverse Distance Weighting	IDW
Neural networks	NN
Multilayer perceptron	MLP
Multiresolution analysis	MRA
Rectified linear unit)	ReLU
Sample Entropy	SampEn
Power Spectrum Density	PSD
Step 1.1	S 1.1
Step 1.2	S 1.2
Step 1.3	S 1.3
Step 1.5	S 1.5
Step 1.6	S 1.6
Step 2.1	S 2.1
Step 2.2	S 2.2
Step 2.3	S 2.3
Step 2.5	S 2.5
Step 2.6	S 2.6
Bandwidth	B
Carrier frequency	Fc
Standard deviation	STD
F1 score	F1
True positive	TP
False positive	FP
False negative	FN
True negative	TN

References

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, Art. no. 4, Aug. 1999, doi: 10.1109/98.788210.

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, Art. no. 13, Sep. 2006, doi: 10.1016/j.comnet.2006.05.001.

[3] G. Hattab and M. Ibnkahla, "Multiband Spectrum Access: Great Promises for Future Cognitive Radio Networks," *Proceedings of the IEEE*, vol. 102, no. 3, Art. no. 3, Mar. 2014, doi: 10.1109/JPROC.2014.2303977.

[4] S. E. El-Khamy, M. S. El-Mahallawy, and E. S. Youssef, "Improved wideband spectrum sensing techniques using wavelet-based edge detection for cognitive radio," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, San Diego, CA: IEEE, Jan. 2013, pp. 418–423. doi: 10.1109/ICCNC.2013.6504120.

[5] A. Kumar, S. Saha, and R. Bhattacharya, "Wavelet transform based novel edge detection algorithms for wideband spectrum sensing in CRNs," *AEU - International Journal of Electronics and Communications*, vol. 84, pp. 100–110, Feb. 2018, doi: 10.1016/j.aeue.2017.11.024.

[6] X. Diao, Q. Dong, Z. Yang, and Y. Li, "Double-Threshold Cooperative Spectrum Sensing Algorithm Based on Sevcik Fractal Dimension," *Algorithms*, vol. 10, no. 3, Art. no. 3, Aug. 2017, doi: 10.3390/a10030096.

[7] J. J. Popoola and R. van Olst, "Application of neural network for sensing primary radio signals in a cognitive radio environment," in *IEEE Africon '11*, Victoria Falls, Livingstone, Zambia: IEEE, Sep. 2011, pp. 1–6. doi: 10.1109/AFRCON.2011.6072009.

[8] N. Shamsi, A. Mousavinia, and H. Amirpour, "A channel state prediction for multi-secondary users in a cognitive radio based on neural network," in *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, Ankara, Turkey: IEEE, Nov. 2013, pp. 200–203. doi: 10.1109/ICECCO.2013.6718263.

[9] Y. Molina-Tenorio, A. Prieto-Guerrero, and R. Aguilar-Gonzalez, "A Novel Multiband Spectrum Sensing Method Based on Wavelets and the Higuchi Fractal Dimension," *Sensors*, vol. 19, no. 6, Art. no. 6, Mar. 2019, doi: 10.3390/s19061322.

[10] D. J. Zaidawi and S. B. Sadkhan, "Blind Spectrum Sensing Algorithms in CRNs: A Brief Overview," in *2021 7th International Engineering Conference "Research & Innovation amid Global Pandemic" (IEC)*, Erbil, Iraq: IEEE, Feb. 2021, pp. 78–83. doi: 10.1109/IEC52205.2021.9476142.

[11] Y. Molina-Tenorio, A. Prieto-Guerrero, and R. Aguilar-Gonzalez, "Real-Time Implementation of Multiband Spectrum Sensing Using SDR Technology," *Sensors*, vol. 21, no. 10, p. 3506, May 2021, doi: 10.3390/s21103506.

[12] C. Politis, S. Maleki, J. M. Duncan, J. Krivochiza, S. Chatzinotas, and B. Ottesten, "SDR Implementation of a Testbed for Real-Time Interference Detection With Signal Cancellation," *IEEE Access*, vol. 6, pp. 20807–20821, 2018, doi: 10.1109/ACCESS.2018.2825885.

[13] R. W. Stewart *et al.*, "A low-cost desktop software defined radio design environment using MATLAB, simulink, and the RTL-SDR," *IEEE Commun. Mag.*, vol. 53, no. 9, Art. no. 9, Sep. 2015, doi: 10.1109/MCOM.2015.7263347.

[14] O. Hiari and R. Mesleh, "A Reconfigurable SDR Transmitter Platform Architecture for Space Modulation MIMO Techniques," *IEEE Access*, vol. 5, pp. 24214–24228, 2017, doi: 10.1109/ACCESS.2017.2761859.

[15] E. Santos-Luna, A. Prieto-Guerrero, R. Aguilar-Gonzalez, V. Ramos, M. Lopez-Benitez, and M. Cardenas-Juarez, "A Spectrum Analyzer Based on a Low-Cost Hardware-Software Integration," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada: IEEE, Oct. 2019, pp. 0607–0612. doi: 10.1109/IEMCON.2019.8936239.

[16] S. Aghabeiki, C. Hallet, N. E.-R. Noutehou, N. Rassem, I. Adjali, and M. Ben Mabrouk, "Machine-learning-based spectrum sensing enhancement for software-defined radio applications," in *2021 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, Cleveland, OH, USA: IEEE, Jun. 2021, pp. 1–6. doi: 10.1109/CCAAW50069.2021.9527294.

[17] A. F. B. Selva, A. L. G. Reis, K. G. Lenzi, L. G. P. Meloni, and S. E. Barbin, "Introduction to the Software-defined Radio Approach," *IEEE LATIN AMERICA TRANSACTIONS*, vol. 10, no. 1, Art. no. 1, 2012.

[18] "About RTL-SDR," *rtl-sdr.com*, Apr. 11, 2013. https://www.rtl-sdr.com/about-rtl-sdr/ (accessed Aug. 08, 2022).

[19] K. Koutlia, B. Bojović, S. Lagén, and L. Giupponi, "Novel radio environment map for the ns-3 NR simulator," in *Proceedings of the Workshop on ns-3*, Virtual Event USA: ACM, Jun. 2021, pp. 41–48. doi: 10.1145/3460797.3460803.

[20] J. Li, G. Ding, X. Zhang, and Q. Wu, "Recent Advances in Radio Environment Map: A Survey," in *Machine Learning and Intelligent Communications*, X. Gu, G. Liu, and B. Li, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 226. Cham: Springer International Publishing, 2018, pp. 247–257. doi: 10.1007/978-3-319-73564-1_25.

[21] C. M. Spooner and N. V. Khambekar, "Spectrum sensing for cognitive radio: A signal-processing perspective on signal-statistics exploitation," in *2012 International Conference on Computing, Networking and Communications (ICNC)*, Maui, HI, USA: IEEE, Jan. 2012, pp. 563–568. doi: 10.1109/ICCNC.2012.6167485.

[22] X. Zhang, Y. Zhao, and H. Chen, "Adaptive Compressed Wideband Spectrum Sensing Based on Radio Environment Map Dedicated for Space Information Networks," in *Wireless and Satellite Systems*, M. Jia, Q. Guo, and W. Meng, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 280. Cham: Springer International Publishing, 2019, pp. 128–138. doi: 10.1007/978-3-030-19153-5_12.

[23] Y. H. Santana, D. Plets, R. M. Alonso, G. G. Nieto, L. Martens, and W. Joseph, "Radio Environment Map of an LTE Deployment Based on Machine Learning Estimation of Signal Levels," in *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Bilbao, Spain: IEEE, Jun. 2022, pp. 01–06. doi: 10.1109/BMSB55706.2022.9828582.

[24] V. Borisov, T. Leemann, K. Sessler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep Neural Networks and Tabular Data: A Survey," *IEEE Trans. Neural Netw. Learning Syst.*, pp. 1–21, 2022, doi: 10.1109/TNNLS.2022.3229161.

[25] M. Xu, Z. Yin, Y. Zhao, and Z. Wu, "Cooperative Spectrum Sensing Based on Multi-Features Combination Network in Cognitive Radio Network," *Entropy*, vol. 24, no. 1, p. 129, Jan. 2022, doi: 10.3390/e24010129.

[26] J. Li, G. Ding, X. Zhang, and Q. Wu, "Recent Advances in Radio Environment Map: A Survey," in *Machine Learning and Intelligent Communications*, X. Gu, G. Liu, and B. Li, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 226. Cham: Springer International Publishing, 2018, pp. 247–257. doi: 10.1007/978-3-319-73564-1_25.

[27] N. S.-N. Lam, "Spatial Interpolation Methods: A Review," *The American Cartographer*, vol. 10, no. 2, pp. 129–150, Jan. 1983, doi: 10.1559/152304083783914958.

[28] P. A. Burrough, R. McDonnell, and P. A. Burrough, *Principles of geographical information systems*. in Spatial information systems. Oxford ; New York: Oxford University Press, 1998.

[29] B. I. Harman, H. Koseoglu, and C. O. Yigit, "Performance evaluation of IDW, Kriging and multiquadric interpolation methods in producing noise mapping: A case study at the city of Isparta, Turkey," *Applied Acoustics*, vol. 112, pp. 147–157, Nov. 2016, doi: 10.1016/j.apacoust.2016.05.024.

[30] Arseni, Voiculescu, Georgescu, Iticescu, and Rosu, "Testing Different Interpolation Methods Based on Single Beam Echosounder River Surveying. Case Study: Siret River," *IJGI*, vol. 8, no. 11, p. 507, Nov. 2019, doi: 10.3390/ijgi8110507.

[31] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, no. 8, pp. 1246–1266, Dec. 1963, doi: 10.2113/gsecongeo.58.8.1246.

[32] M. A. Oliver and R. Webster, "Kriging: a method of interpolation for geographical information systems," *International journal of geographical information systems*, vol. 4, no. 3, pp. 313–332, Jul. 1990, doi: 10.1080/02693799008941549.

[33] K. S. Gundogdu and I. Guney, "Spatial analyses of groundwater levels using universal kriging," *J Earth Syst Sci*, vol. 116, no. 1, pp. 49–55, Feb. 2007, doi: 10.1007/s12040-007-0006-6.

[34] N. A. C. Cressie, *Statistics for spatial data*, Revised edition. Hoboken, NJ: John Wiley & Sons, Inc, 2015.

[35] E. H. Isaaks and R. M. Srivastava, *Applied geostatistics*. New York: Oxford University Press, 1989.

[36] J. Han and M. Kamber, *Data mining: concepts and techniques*, 3rd ed. Burlington, MA: Elsevier, 2012.

[37] W. Wang and Y. Yang, "Development of convolutional neural network and its application in image classification: a survey," *Opt. Eng.*, vol. 58, no. 04, p. 1, Apr. 2019, doi: 10.1117/1.OE.58.4.040901.

[38] S. S. Haykin and S. S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[40] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[41] Y. Molina-Tenorio, A. Prieto-Guerrero, and R. Aguilar-Gonzalez, "Multiband Spectrum Sensing Based on the Sample Entropy," *Entropy*, vol. 24, no. 3, p. 411, Mar. 2022, doi: 10.3390/e24030411.

[42] Y. Molina-Tenorio, A. Prieto-Guerrero, R. Aguilar-Gonzalez, and S. Ruiz-Boqué, "Machine Learning Techniques Applied to Multiband Spectrum Sensing in Cognitive Radios," *Sensors*, vol. 19, no. 21, Art. no. 21, Oct. 2019, doi: 10.3390/s19214715.

[43] "Nooelec - Nooelec NESDR SMArt v4 SDR - Premium RTL-SDR w/ Aluminum Enclosure, 0.5PPM TCXO, SMA Input. RTL2832U & R820T2-Based - Software Defined Radio." https://www.nooelec.com/store/sdr/nesdr-smart-sdr.html (accessed Mar. 08, 2021).

[44] "HackRF One - Great Scott Gadgets," Mar. 08, 2021. https://greatscottgadgets.com/hackrf/one/ (accessed Mar. 08, 2021).

[45] "LimeSDR Mini is a $135 Open Source Hardware, Full Duplex USB SDR Board (Crowdfunding) - CNX Software," *CNX Software - Embedded Systems News*, Sep. 18, 2017. https://www.cnx-software.com/2017/09/18/limesdr-mini-is-a-135-open-source-hardware-full-duplex-usb-sdr-board-crowdfunding/ (accessed Mar. 13, 2022).

[46] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, Jun. 1967, doi: 10.1109/TAU.1967.1161901.

[47] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, Art. no. 7, Jul. 1989, doi: 10.1109/34.192463.

[48] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, Art. no. 8, Jun. 2010, doi: 10.1016/j.patrec.2009.09.011.

[49] Y. Sasaki, "The truth of the f-measure," Oct. 2007, Accessed: Apr. 07, 2007. [Online]. Available: https://www.cs.odu.edu/~mukka/cs795sum11dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf