

## Article

# Cooperative Multiband Spectrum Sensing Using Radio Environment Maps and Neural Networks

Yanqueleth Molina-Tenorio <sup>1</sup>, Alfonso Prieto-Guerrero <sup>2</sup>, Rafael Aguilar-Gonzalez <sup>3,4,\*</sup>  
and Miguel Lopez-Benitez <sup>5,6</sup>

- <sup>1</sup> Information Science and Technology Ph.D., Metropolitan Autonomous University, Mexico City 09360, Mexico; yanqueleth@xanum.uam.mx
- <sup>2</sup> Electrical Engineering Department, Metropolitan Autonomous University, Mexico City 09360, Mexico; apg@xanum.uam.mx
- <sup>3</sup> Faculty of Science, Autonomous University of San Luis Potosi, San Luis Potosi 78210, Mexico
- <sup>4</sup> Engineering Department, Arkansas State University Campus Queretaro, Queretaro 76270, Mexico
- <sup>5</sup> Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, UK; mlopben@liverpool.ac.uk
- <sup>6</sup> ARIES Research Centre, Antonio de Nebrija University, 28040 Madrid, Spain
- \* Correspondence: rafael.aguilar@uaslp.mx

**Abstract:** Cognitive radio networks (CRNs) require high capacity and accuracy to detect the presence of licensed or primary users (PUs) in the sensed spectrum. In addition, they must correctly locate the spectral opportunities (holes) in order to be available to nonlicensed or secondary users (SUs). In this research, a centralized network of cognitive radios for monitoring a multiband spectrum in real time is proposed and implemented in a real wireless communication environment through generic communication devices such as software-defined radios (SDRs). Locally, each SU uses a monitoring technique based on sample entropy to determine spectrum occupancy. The determined features (power, bandwidth, and central frequency) of detected PUs are uploaded to a database. The uploaded data are then processed by a central entity. The objective of this work was 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 radioelectric environment maps (REMs). To this end, we compared the results of classical digital signal processing methods and neural networks performed by the central entity. Results show that both proposed cognitive networks (one working with a central entity using typical signal processing and one performing with neural networks) accurately locate PUs and give information to SUs to transmit, avoiding the hidden terminal problem. However, the best-performing cognitive radio network was the one working with neural networks to accurately detect 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:** Molina-Tenorio, Y.; Prieto-Guerrero, A.; Aguilar-Gonzalez, R.; Lopez-Benitez, M. Cooperative Multiband Spectrum Sensing Using Radio Environment Maps and Neural Networks. *Sensors* **2023**, *23*, 5209. <https://doi.org/10.3390/s23115209>

Academic Editors: Jia Liu, Kechen Zheng, Xiaoying Liu, Zhao Li and Yang Xu

Received: 13 April 2023

Revised: 3 May 2023

Accepted: 17 May 2023

Published: 30 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed 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 capable of knowing the spectral behavior in its environment and adapting to it. Taking advantage of spectral gaps (or holes) not utilized by licensed users, also known as primary users (PUs), CR technology allows nonlicensed or secondary users (SUs) to detect these available parts of the spectrum [1]. Specifically, the operation of CR involves four stages or functions: spectrum sensing, spectrum sharing, spectrum decision, and spectrum mobility. Spectrum sensing (SS) is a fundamental task of detecting one or more PUs; this stage shows whether the sensed spectrum is occupied or empty [2]. Usually, this task is carried out in single bands; however, the current paradigm of multiband spectrum sensing (MBSS) involves multiple bands that are not necessarily contiguous [3].

Many MBSS techniques issued from digital signal processing and machine learning (ML), such as wavelets, compressed sensing, energy detectors, and blind or semiblind methods, have been proposed, primarily in simulated scenarios [4–10]. 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 affordability and good performances [17,18]. These generic communication devices offer radio equipment the flexibility of a programmable system, allowing the modifications to the communication system behavior simply by changing its software, permitting anyone, including hobbyists on a budget, access to the full radio spectrum.

In addition to knowing the PUs' behavior in a given frequency domain and avoiding the hidden terminal issue, knowledge of the behavior of the radio spectrum in its specific geographical area of influence is necessary. For this reason, the idea of including a geographic tool constructed with the radioelectric information provided by the SUs issued from a real environment is pertinent. Under this perspective, the ability to build a radio environment map (REM) has become very important in recent years. 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]. Thus, REMs permitting the characterization of the position, directivity, power, and modulation type of PUs have become a challenging task in cognitive radio network (CRN) design [20]. Indeed, in [21], REMs were used to locate relevant PUs in a geographic region of interest, characterizing their positions, directivities, powers, and modulation types. Likewise, in [22], REMs were used to sense the spectrum based on an adaptive compressed spectrum-sensing algorithm, contributing spatial information to the network capable of adapting to the radio environment. REMs are very flexible tools, as shown in [23], where they are combined with ML to determine the effective coverage area perceived by a cognitive sensor network, correctly estimating it at around 92%.

Another important tool, owing to the computing power and the amount of available data, is neural networks (NNs), used widely from pattern recognition and image classification to financial market behavior prediction and autonomous vehicle driving [24]. For spectrum sensing, for example, in [25], an NN was implemented to obtain the local information on single-node spectrum detection (spatial and temporal features). The information (extracted features) from multiple nodes fed another NN, thus permitting cooperation in the CRN. On the basis of REM and NN paradigms, this work proposed a new CRN methodology, implementing an MBSS involving a network integrated with low-cost SDR devices in a controlled, realistic wireless communications environment.

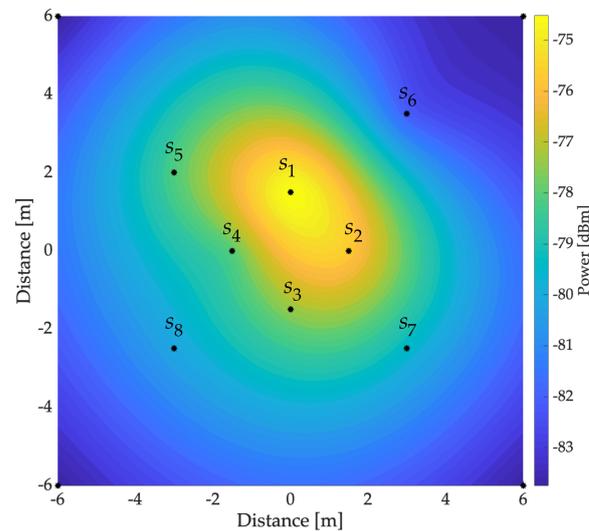
This work developed a CRN for monitoring multiband spectrum and locating PUs with their characteristic bandwidth, carrier frequency, and power in specific areas. In addition, our CRN located the spectral opportunities where the SUs could be located and, unlike models of other studies, could be implemented in real time. In the following, Section 2 briefly lays out the theoretical basis of REMs and NNs, and Section 3 presents the authors' previous work in implementing the MBSS used as a basis for the new proposal presented in this work. Section 4 explains in detail the new proposed methodology, and Section 5 details the implemented real scenario. Finally, Section 6 offers the results, conclusions, and discussion [20].

## 2. Theoretical Basis of REMs and NNs

### 2.1. Construction of Radio Environment Maps

REMs are used in telecommunications research to represent the power distribution at a specific area emitted by different radio sources (one or more). A REM is constructed by collecting power measurements at different points in a specific area and interpolating them to produce a graphical representation (map) of the power distribution (or coverage)

of the emitted signals, as shown in Figure 1. These maps are generally used to evaluate and predict the behavior of radio signals in a given environment, which is essential 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 [20].



**Figure 1.** Example of a REM: Black dots indicate locations where the power measurements are taken using eight sensors. In this case, only a transmission source with  $-60$  dBm is considered.

Collecting and interpolating power measurements are the two essential processes in constructing a *correct* map. In this case, interpolation methods [26] are critical for *correctly* estimating 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 a few known values of measured power. Two interpolation methods widely used to build REMs are inverse distance weighting (IDW) and Kriging, and both are explained below.

#### 2.1.1. IDW Method

The inverse distance weighting interpolation method [27,28] is simple and easy to implement, and it is widely used in applications such as precipitation estimation, topographic data interpolation, and air pollution estimation. IDW is a classical interpolation method in spatial analysis and is most commonly used in geostatistical and mathematical interpolation [29].

IDW estimates the value of a variable at a specific point based on known values at nearby points. This method rests on the idea that nearby points have a greater impact on the estimate than more distant points. Thus, IDW uses a formula that assigns a weight to each known point based on the distance between them and the unknown point. The weights are determined by considering the inverse of the distance, so that the closest points have higher weights and the farther ones have lower weights. For the IDW method, we considered the data transposed vector  $\mathbf{z} = (z(s_1), \dots, z(s_n))^T$  as observations from a random process. In our case,  $\mathbf{z}$  contained the average power at each point of the considered sampled region (see Figure 1)

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

at known geographic locations  $s_1, \dots, s_n$ . The final estimate was obtained using the weighted sum of the known values in the nearby points, given by [28]:

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

where  $Z(x)$  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  $s_i$ ,  $d(s_i, s)$  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  $w_i(s_i)$  is the weight assigned to each known point  $s_i$  in order to account for the uncertainty in the known data. In practice, a value of  $p = 2$  or  $p = 3$  is usually used. The estimated values of  $Z(s)$  result in a mesh of unknown points. However, this method has limitations, such as a tendency to smooth out data variability and produce an inaccurate estimate in areas with few known points. In this work,  $Z(s)$  represents the power spectral density (PSD) in the different points around the sensed area.

### 2.1.2. Kriging Method

The Kriging method [30] is an interpolation technique derived from regionalized variable theory. It depends on expressing the spatial variation of the property using the variogram, and it minimizes the estimated prediction errors [31]. The goal of Kriging is to find the most accurate and least uncertain estimate by considering 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. Following [30], the Kriging method can be established by taking the points and the region mentioned in (1), assuming  $\mu$  is known and

$$z(s) = \mu + \delta(s); s \in D, \quad (3)$$

are considered with known covariance function:

$$C(s, u) \equiv \text{cov}(z(s), z(u)); s, u \in D, \quad (4)$$

where  $\delta(\cdot)$  is a zero-mean stochastic process. Thus, the best linear unbiased predictor  $z(s_0)$  is obtained by minimizing [32]

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

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

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

The optimal values were computed using the method of Lagrange multipliers. For this, the mean-squared prediction error (Equation (5)) was obtained by

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

where

$$c \equiv (C(s_0, s_1), \dots, C(s_0, s_n))', \quad (8)$$

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

and

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

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

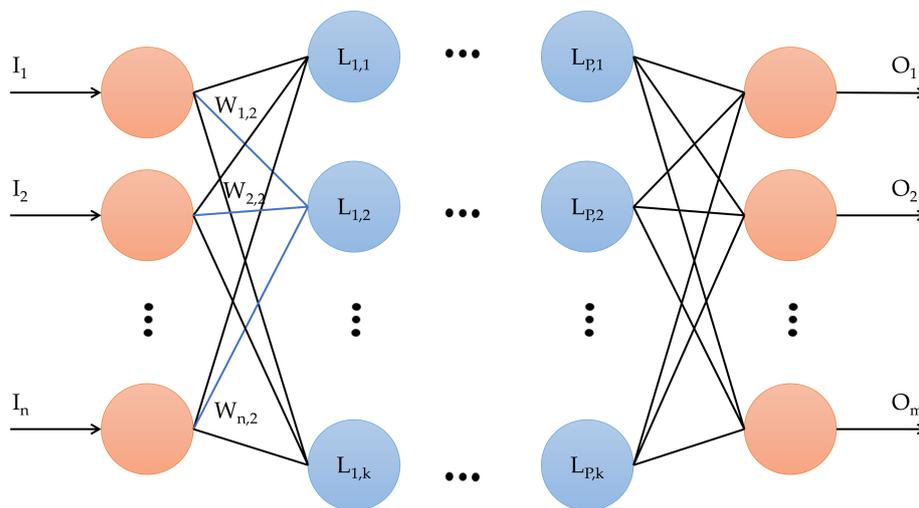
where  $\gamma \equiv (\gamma(s_0, s_1), \dots, \gamma(s_0, s_n))'$ ,  $\Gamma$  is an  $n \times n$  matrix whose  $(i, j)^{th}$  element is  $\gamma(s_i, s_j)$ , and

$$2\gamma(s_i, s_j) \equiv C(s_i, s_i) + C(s_j, s_j) - 2C(s_i, s_j), \tag{12}$$

is the variogram  $[\gamma(s_i, s_j) \gamma(s_i, s_j)]$ , named the semivariogram [33]. The variogram, a valuable tool in modeling spatial variables, describes how the data are correlated with distance and, in doing so, allows us to accomplish spatial interpolation using the sampled data and the variogram information to estimate the variance of the values of the variable at the unsampled points [33,34]. In summary, Kriging was more precise than other interpolation methods, such as IDW or cubic interpolation [29], and suitable for applications where the data have a certain spatial dependence.

### 2.2. Neural Networks

Neural networks are artificial intelligence models inspired by the structure and functioning of the human brain. These networks comprise many nodes, known as „neurons“, connected to each other through „synapses“. Each neuron receives inputs from other neurons, processes them through an activation function, and sends its output to other neurons, as 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. Neural networks are a powerful tool for machine learning because they can model complex relationships between inputs and outputs and make accurate inferences from real data. However, these tasks require a large amount of training data and are computationally intensive (sometimes impossible) to train and use in real-time implementations [35].



**Figure 2.** Neural network concept: the multilayer perceptron.

In this work, we used the multilayer perceptron (MLP), as shown in Figure 2. 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 to the output layers, passing through the

hidden intermediate layers between the input and the output [36]. The general formula for calculating the output of a multilayer perceptron is given by [37]

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

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

The formula is applied to every neuron in every network layer, including the input, hidden, and output layers. Each neuron in a hidden layer takes the output of all the neurons in the previous layer (i.e., assuming a fully connected network) as its input, and the output of one neuron in the output layer is the network's final output. During the training phase, the weights and the bias are adjusted to minimize the error between the network output and the desired output. This is achieved by a learning algorithm, such as the backpropagation algorithm [38], which updates the weights and bias in the direction of the downward gradient [39].

### 3. Preliminary Work

This section briefly describes the preliminary work on a novel MBSS technique based on the sample entropy (SampEn) developed and published by the authors in [39], constituting the basis of the new proposal. However, the MBSS technique presented in [39] was the result of the authors' previous efforts to refine and complement it with modules implemented in real time to, for instance, diminish the noise introduced by SDR devices. Hence, several comparative studies have been carried out in order to measure the performance of the proposed MBSS technique compared with other classical methods, such as energy detectors. The efficiency of locating the PUs by this MBSS technique stands out in SNR values close to 0 dB [11]. Table 1 highlights the principal contributions of these 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].	<ul style="list-style-type: none"> <li>- Two MBSS techniques: Wavelets and multiresolution analysis (MRA). Decision rule based on the Higuchi fractal dimension (HFD).</li> <li>- 95% effective in detecting PUs for SNR values higher than 0 dB.</li> <li>- Controlled simulated environment.</li> <li>- Blind technique.</li> </ul>
"Multiband Spectrum Sensing Based on the Sample Entropy" [40].	<ul style="list-style-type: none"> <li>- MBSS technique based on SampEn.</li> <li>- Technique implemented in a real environment of wireless communications.</li> <li>- Technique with real-time operation (updated every 100 ms).</li> <li>- Probability of 0.99 to correctly detect the PU for SNR values greater than 0 dB.</li> <li>- Cooperation of the different SUs to sense a wide range of frequency.</li> <li>- Blind technique.</li> </ul>

Table 1. Cont.

Previous Work.	Contributions
“Real-Time Implementation of Multiband Spectrum Sensing Using SDR Technology” [11].	<ul style="list-style-type: none"> <li>- MBSS technique based on MRA, ML, and HFD implemented in a real wireless communications environment.</li> <li>- Real-time operation (update every 100 ms).</li> <li>- Implemented with SDR devices.</li> <li>- A module is proposed for the elimination through the detailed coefficients obtained with the MRA.</li> <li>- Efficiency of 95% for SNR values greater than 0 dB.</li> <li>- Blind technique.</li> </ul>
“Machine Learning Techniques Applied to Multiband Spectrum Sensing in Cognitive Radios” [41].	<ul style="list-style-type: none"> <li>- Three machine learning methods are applied to the MBSS</li> <li>- Effectiveness of 98% in detecting PUs in the spectrum for SNR values greater than 0 dB.</li> <li>- Controlled simulated environment.</li> <li>- Blind technique.</li> </ul>

The last MBSS technique from Table 1 was implemented in a real environment using low-cost SDR devices. Each connected SDR device was considered an SU, independent from others, processing its information locally in a determined single band to achieve ensemble wideband spectrum sensing. The MBSS technique used in this proposal, however, combines different SDR devices, namely the RTL-SDR [42], the HackRF One [43], and the LimeSDR mini [44], to conform each considered SU, forming a CR infrastructure, which permits having an MBSS in a specific location. Figure 3 below shows how this MBSS method works. The MBSS blocks are as follows:

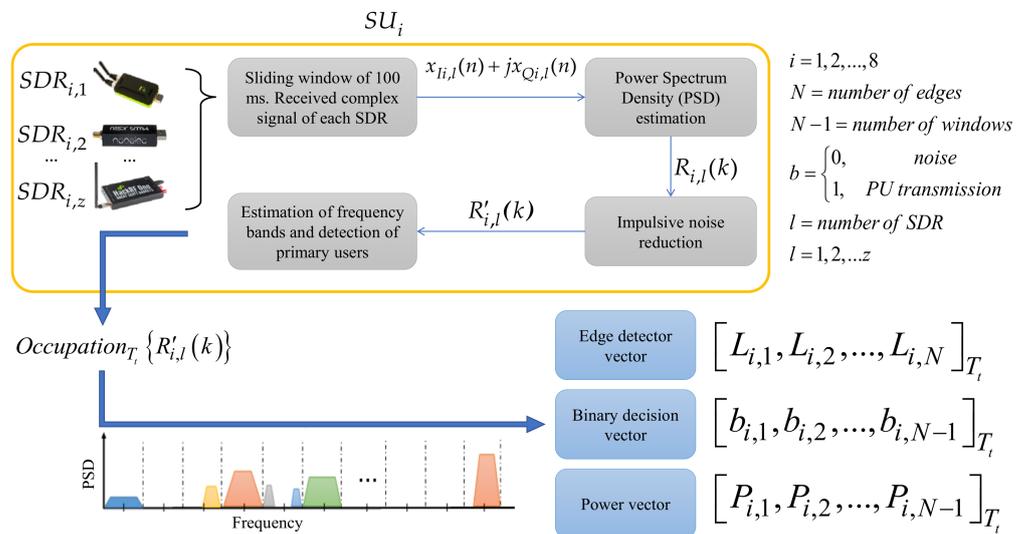


Figure 3. General scheme of implemented MBSS technique [40].

- **Sliding window of 100 ms.** In this block, the complex signal  $x_{i,l}(n) + jx_{Q,i,l}(n)$  from each SDR in the time domain was 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 [45] was applied to each signal  $x_{i,l}(n) + jx_{Q,i,l}(n)$  in order to obtain, on a linear scale, the wideband PSD  $R_{i,l}(k)$  from the SU 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  $R_{i,l}(k)$  were eliminated or diminished through discrete wavelets via the multiresolution analysis [46], 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 SampEn, K-means algorithm [47] (permitting to optimize specific detection parameters), and discrete wavelets was applied to each SU's analyzed wideband spectrum in order to obtain the spectrum occupation given by  $Occupation_{T_i} \{R'_{i,l}(k)\}$ . This result included three vectors  $[b_{i,1}, b_{i,2}, \dots, b_{i,N-1}]_{T_i}$  which contained binary values indicating occupied ("1") and empty ("0") bands. The second vector was  $[L_{i,1}, L_{i,2}, \dots, L_{i,N}]_{T_i}$  and contained the corresponding computed boundaries for each detected band, and the third was  $[P_{i,1}, P_{i,2}, \dots, P_{i,N-1}]_{T_i}$ , which contained the power for each detected band.

One of the primary motivations for this technique was to propose an MBSS method (i) adequate to correctly detect a primary user and the operating parameters, such as carrier frequency, bandwidth, and power, with (ii) computational complexity permitting a real-time implementation. Both aims were fulfilled, and the results were used to create a compact CRN capable of sharing spectral information with a central entity in order to estimate the area occupied by different detected PUs in the studied zone. A wireless communications environment was proposed considering these SDR devices.

#### 4. New Proposal and Methodology

This section presents the new research proposal and the implemented methodology.

##### 4.1. Proposal

Our previous work's novel idea was to consider a computer integrating different SDR devices detecting each PU in a single band to sense a wide frequency range, as shown in Figure 4a. This model was seen as the only entity containing different SUs. Our follow-up work proposed replicating this entity containing different connected SDR devices and creating a cooperative CR network for sensing a wide frequency range in a broader geographical region. Each entity was considered an SU containing different technologies for sensing a broad spectrum, as shown in Figure 4b. In addition, to sense the radio spectrum and uncover the PUs' behavior in a given geographical region, this cooperative CR network had the task of avoiding the hidden terminal problem. One of the main strengths of this new proposal (see Figure 4b) over previous work is the cooperation of the different SUs in geographically locating the PUs through the REMs and the ability to carry out this processing in real time.

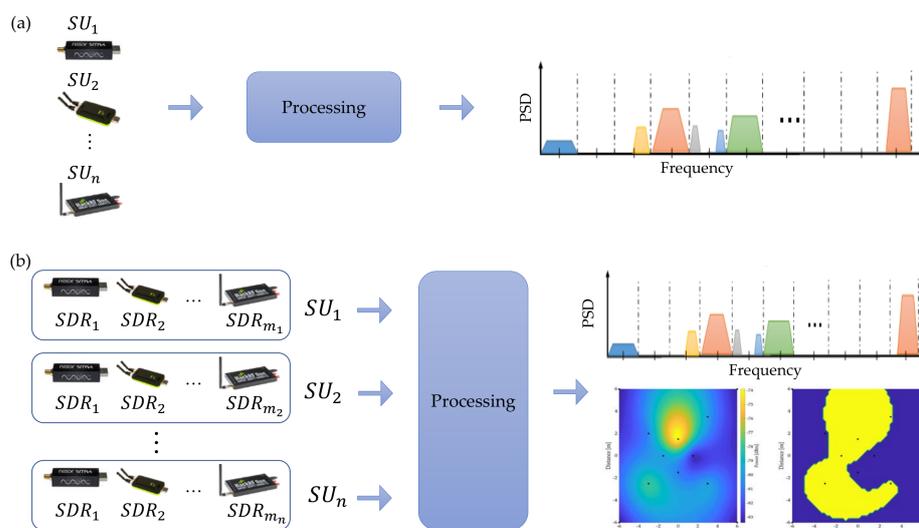
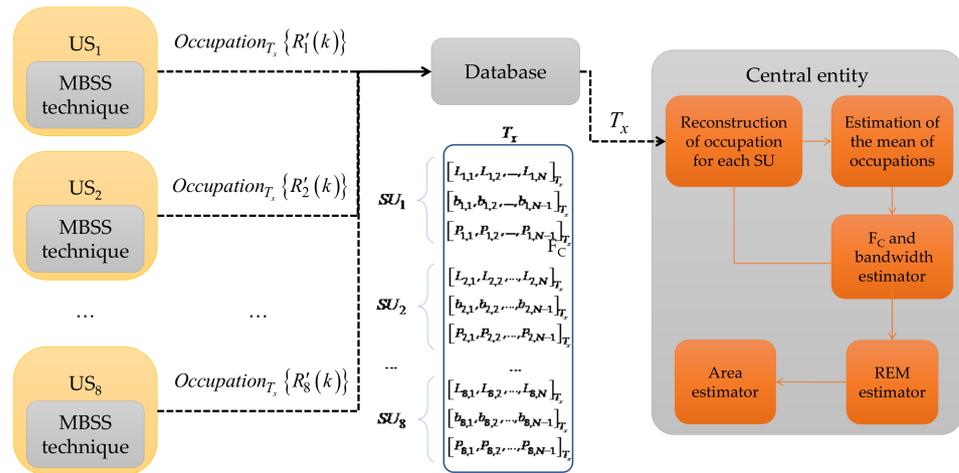


Figure 4. (a) Previous work and (b) new proposal.

#### 4.2. Methodology

The proposed methodology can be outlined in three big blocks, shown in Figure 5:

- The collection of information obtained by SUs. For this, each SU locally processes the sensed data to be sent to a central entity, including the occupancy of the observed spectrum in its geographic location and the frequency band edges and estimated power vectors;
- The database for storing the information obtained by each SU at a specific time;
- The central entity overseeing the processing determines the geographic area occupied by the detected PUs in the radio spectrum.



**Figure 5.** General scheme of the methodology.

Each block is described in detail below.

##### 4.2.1. Collection of Information Locally by the SUs

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

##### 4.2.2. Database

The edge detector, binary decision, and power vectors were stored in the database. It is important to note that the information stored in the database stemming from each SU was not necessarily similar; the lengths of the vectors were different for each SU. Indeed, each SU observed a different behavior of the radioelectric spectrum, because they were placed randomly in different geographic coordinates. To account for this discrepancy, all vectors were labeled with the exact sensing time  $T_x$  (see Table 2), i.e., synchronized. Information was uploaded to a server, saving only enough relevant data (three different vectors for each SU) in the database to permit fast storage and extraction, taking advantage of SUs' hardware performing the MBSS technique locally. Indeed, the three uploaded vectors for each SU take up very little memory space. In the best case (without even a PU in the sensed spectrum), the edge detection vector had a length of two, while the binary decision vector and the power vector had a length of one. In the worst case, when the SNR had a low value (close to 0 dB), multiple windowing appeared due to noise, thus increasing the vectors' lengths. However, regardless of the size of the three vectors, the information was compact

and facilitated the reconstruction of the spectrum signal occupation, including the average power for each reconstructed window.

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

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

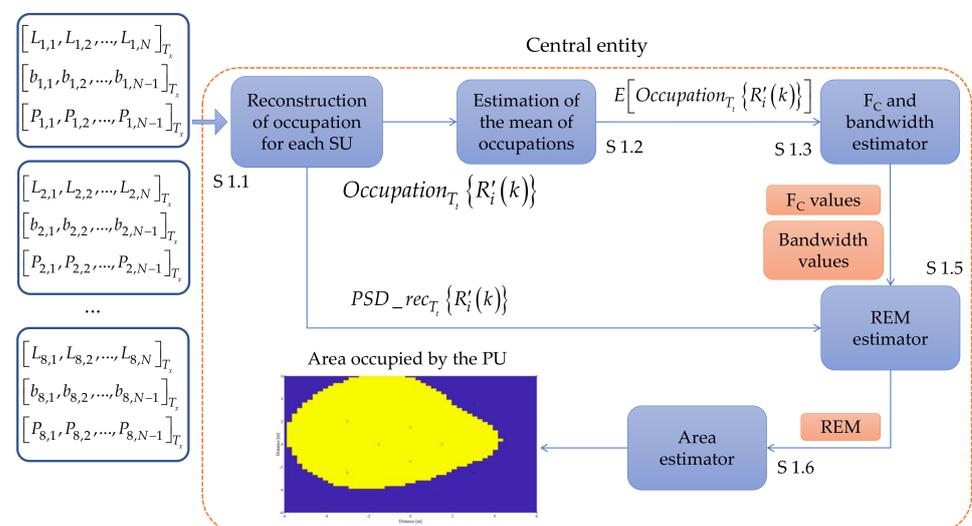
#### 4.2.3. Central Entity

The main tasks of the central entity were (i) to indicate how many PUs appear in the spectrum compared with the perceived SUs, (ii) to build a REM for each detected PU using the location of each SU, the perceived power in each SU, the time of radio space monitoring as parameters, and finally (iii) to show the area covered by the PUs.

Two ways of implementing these tasks were developed, the first through *classical* digital processing and the second using machine learning techniques (specifically, the NN). Both approaches 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 well-established and stable technology that has been used for decades in a variety of applications. It is relatively easy to implement and can be very fast and efficient in situations where the datasets and tasks are specific. In this case, the central entity was implemented with classical techniques for reconstructing the spectrum occupation from the shared information (edge detector, binary decision, and power vectors) obtained by each SU in the database. The central entity processing the information is shown in Figure 6 and described by Algorithm 1.



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

---

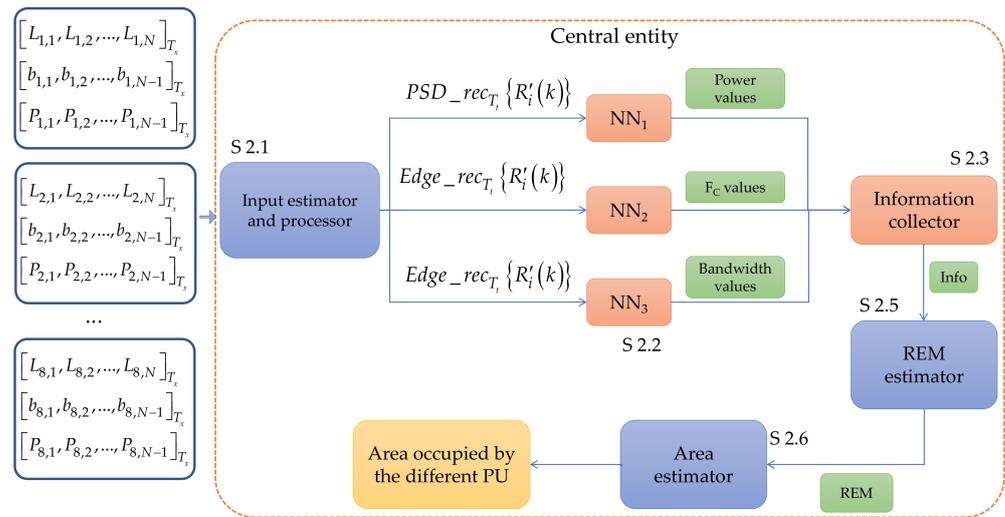
**Algorithm 1** Central Entity Computed by Traditional Signal Processing
 

---

- 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  $Occupation_{T_i}\{R'_i(k)\}$  and constructs simultaneously, an approximation of the power spectral density  $PSD_{rec_{T_i}}\{R'_i(k)\}$  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  $E[Occupation_{T_i}\{R'_i(k)\}]$  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  $F_{C\_vector} = [F_{C1}, F_{C2}, \dots]$  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  $B_{vector} = [B_{PU_1}, B_{PU_2}, \dots]$ , is also obtained. Through the number of singularities and their corresponding widths detected in  $E[Occupation_{T_i}\{R'_i(k)\}]$ , it is possible to estimate how many PUs are in the spectrum and their respective bandwidths ( $B_{vector} = [B_{PU_1}, B_{PU_2}, \dots]$ ). The  $F_{C\_vector}$  is integrated by the central frequency values of estimated bandwidth for each detected PU.
- 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  $PSD_{rec_{T_i}}\{R'_i(k)\}$ . The mean of the interval  $B_{PU_N}$  centered on  $F_{C_N}$  of the  $PSD_{rec_{T_i}}\{R'_i(k)\}$  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 PU 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 [41] 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 Classical Digital Signal Processing

The central entity was implemented using ML techniques, specifically NNs. NNs have the ability to learn from data and improve their performance as they absorb more information. This makes them particularly useful in applications where the data are complex, difficult to understand, and processed 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 were replaced by NNs, as shown in Figure 7. The steps of this operation (Algorithm 2) are described below.



**Figure 7.** The central entity determining the REM construction parameters and area maps through NNs.

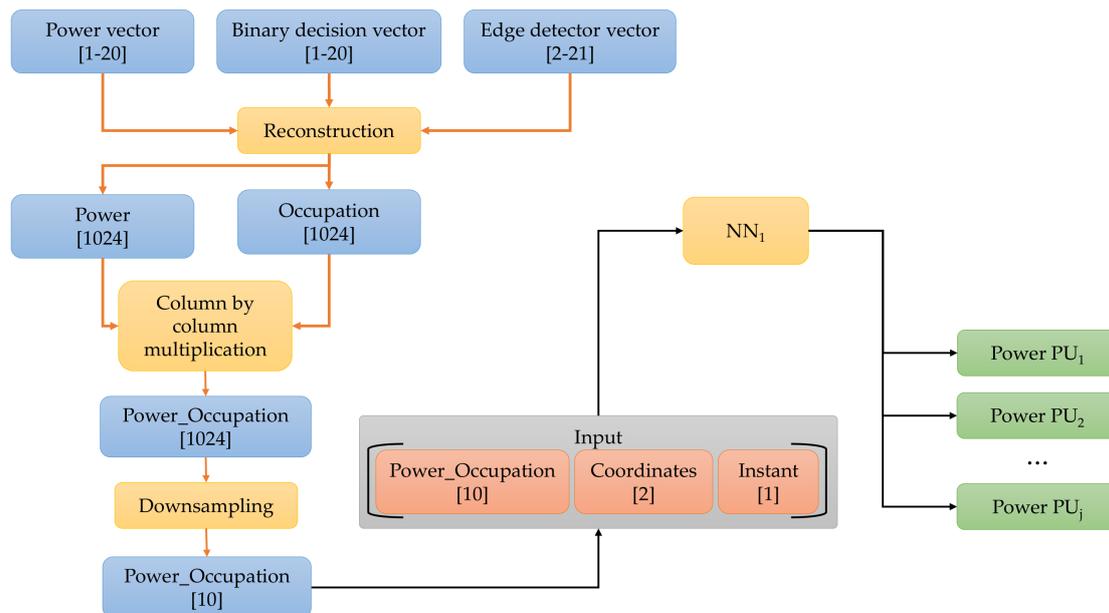
#### Algorithm 2 Central Entity Computed by NNs

- 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  $PSD\_rec_{T_x}\{R'_i(k)\}$  and  $Edge\_rec_{T_x}\{R'_i(k)\}$  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  $PSD\_rec_{T_x}\{R'_i(k)\}$  is evaluated by  $NN_1$ , at the same time the vector  $Edge\_rec_{T_x}\{R'_i(k)\}$  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 include NNs corresponds to Steps 1.5 and 1.6 of the methodology described in Algorithm 1 (Section 4.2.3.1). The *input estimator and processor* (Step 2.1) and the *information collector* (Step 2.3) modules are described below.

Figure 8 shows the process of generating  $PSD\_rec_{T_x}\{R'_i(k)\}$ , the input vector of  $NN_1$ . First, power, edge detector, and binary decision vectors were used, with samples ranging in the intervals of [1–20], [1–20], and [2–21], respectively, according to the studies carried out

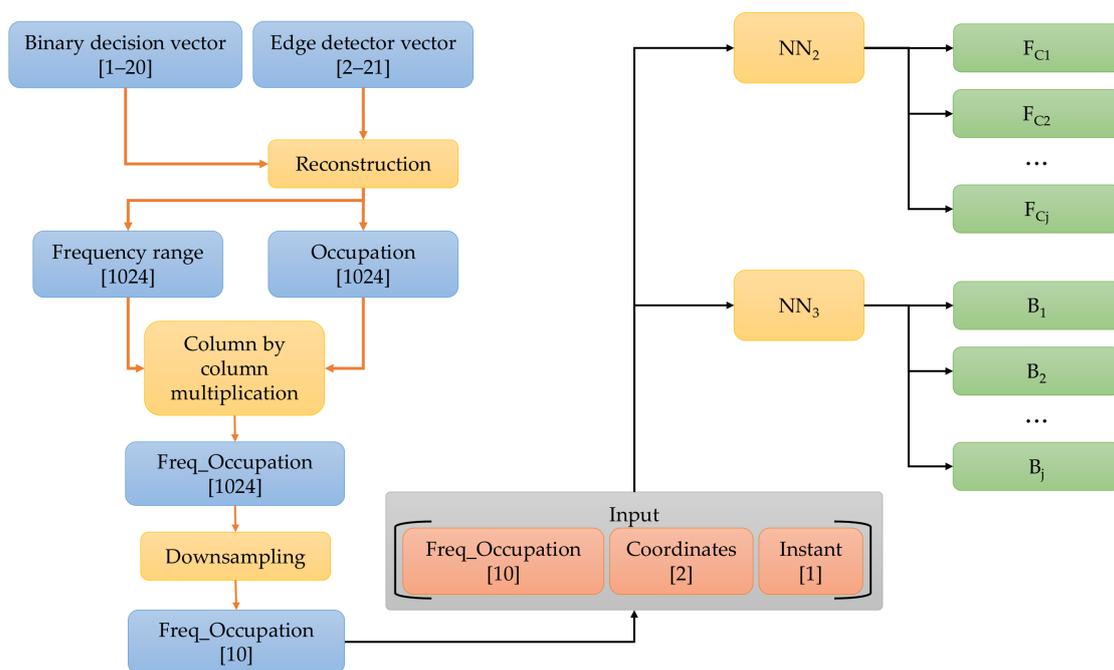
in our preliminary work [39]. As mentioned above, using these vectors, we obtained (i) the frequency range that was sensed by each SU, (ii) the number of samples of the estimated power spectral density, and (iii) the number of windows in which the sensed spectrum was partitioned. An occupation value was assigned for each detected window, 1 for a possible PU transmission or 0 for the noise. Accompanying each occupation value was an average power value or the power vector element representing the power estimate in each detected window. The occupation and the power signal were reconstructed using these average values of each detected window, setting both vectors to a length of 1024 samples each.



**Figure 8.**  $Edge\_rec_{T_i}\{R'_i(k)\}$  vector construction.

After that, the power- and occupation-reconstructed vectors were multiplied column by column, resulting in the *Power\_Occupation* vector of 1024 samples. In this last vector, locations with estimated noise had a value of 0, and those with the possible transmission of one or several PUs had a value corresponding to the average power of detected windows. A downsampling of the *Power\_Occupation* vector was performed to reduce it to only 10 samples and to combine it with the coordinates of the SUs (to which the analyzed vectors correspond) and the time  $T_x$  at which the signal  $Occupation_{T_i}\{R'_{i,l}(k)\}$  was sensed, thus finally integrating the input vector for  $NN_1$ . It should be noted that the time parameter is extremely important since the spectrum behaves differently over time.

Figure 9 shows the construction of the vector  $Edge\_rec_{T_i}\{R'_i(k)\}$ . Here, the occupancy and the frequency bands vectors corresponding to the frequency interval perceived by the corresponding SU were reconstructed. This vector was also multiplied column by column with the occupation vector, resulting in the *Freq\_Occupation* vector being 1024 samples in length. This latter vector was assigned zeros where it corresponded to noise and nonzero values for the samples that corresponded to one or several PU transmissions. Again, downsampling was applied to reduce this vector to only 10 samples and join it with the coordinates of the SUs and the time  $T_x$  corresponding to the sensing period. This concatenation drives the input of  $NN_2$  and  $NN_3$ .

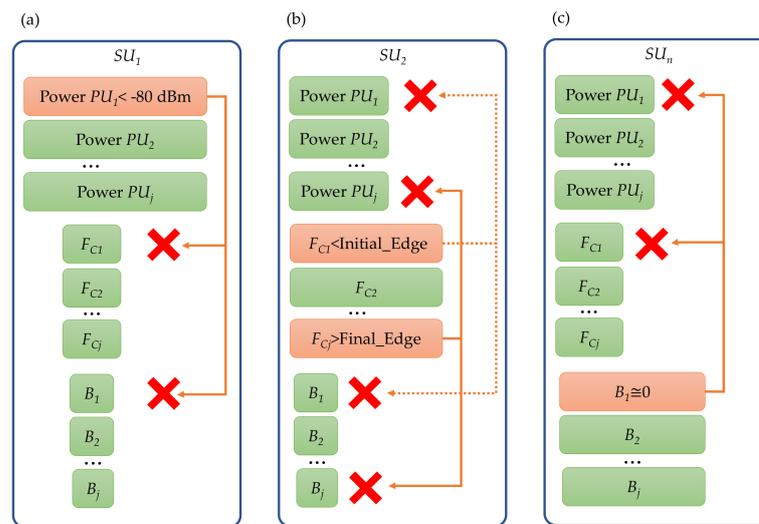


**Figure 9.**  $Edge\_rect_i\{R'_i(k)\}$  vector construction.

The  $NN_1$  output corresponds to the power of each PU detected by each SU. When the PU power value was less than  $-80$  dBm, it was interpreted as no transmission, corresponding to noise. The output of  $NN_2$  resulted in the carrier on which a possible PU was located. Finally, the  $NN_3$  output delivered 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 into three large blocks: powers, carrier frequencies, and transmission bandwidths of each PU detected by the corresponding SU. The *information collector* block analyzes the output of each SU making up the network and sends it to the module permitting the REM estimate. For example, Figure 10a shows that the power of  $PU_1$  is less than  $-80$  dBm; 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 10b shows that the carrier frequencies of  $PU_1$  and  $PU_j$  are outside the monitored space, so the *information collector* shares with the REM estimator that  $SU_2$  only observes  $(j - 2)$  PUs. In Figure 10c, the  $B$  of  $PU_1$  is a very small value (possibly corresponding to impulse noise); thus, the *information collector* shares with the REM estimator module that the  $SU_n$  observes  $(j - 1)$  PUs. It is important to point out that any of these combinations shown in Figure 10 may change the number of observed PUs, i.e., if the power of a first PU does not exceed the threshold of  $-80$  dBm, 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 three PUs will be considered as noise.

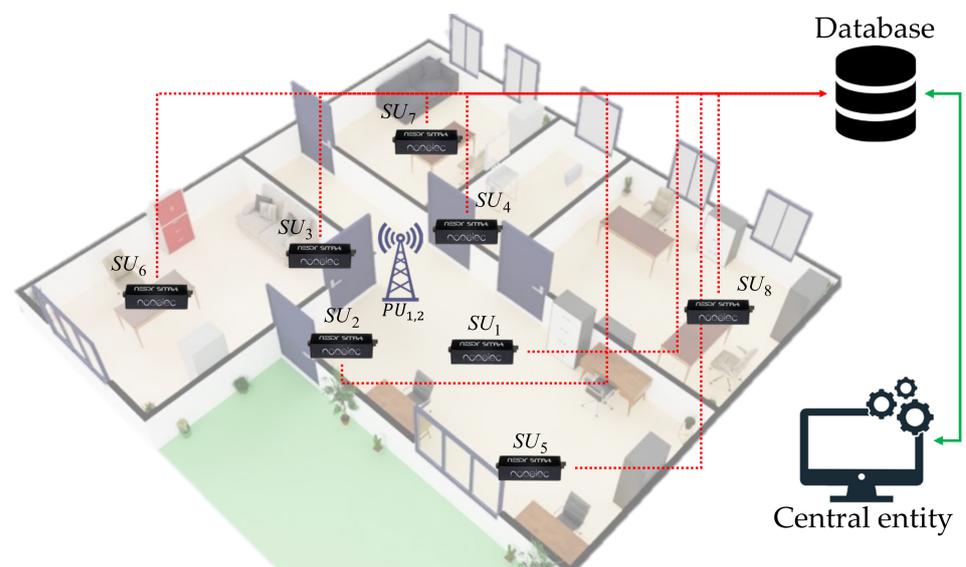
The proposed methodology has been implemented in a real wireless communication environment. This controlled environment is explained in the next section.



**Figure 10.** (a). The  $SU_1$  observes  $j$  PUs, but  $PU_1$  is discarded because it does not meet the threshold. (b) The  $SU_2$  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, but the  $PU_1$  is discarded for not having a large enough bandwidth to be considered a transmission.

## 5. Real Wireless Communication Environment

Figure 11 shows the real, controlled environment implemented in our research. In this proposed scenario, we considered two PUs collocated at the center of the studied area while, at the same time, eight SUs were sensing their behavior in their geographical zone of influence. It is important to note that the SUs and PUs did not share the geographic coordinates among themselves, nor was that information used in spectrum sensing. PUs were located at the center, hoping most SUs could receive part of the PUs' signal. SUs were set randomly in the area of study.



**Figure 11.** The real implemented scenario.

Table 3 specifies the involved parameters of both SUs and PUs. These SUs shared the channel occupancy with the database and the central entity in order to determine (i) how many PUs on average were observed in this environment, (ii) the  $B$  and the  $F_C$  in which they located the detected PUs, and (iii) finally the area occupied by the detected PUs.

**Table 3.** PU and SU main parameters.

Label	Device	$F_C$ Tx (MHz)	$F_C$ 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)

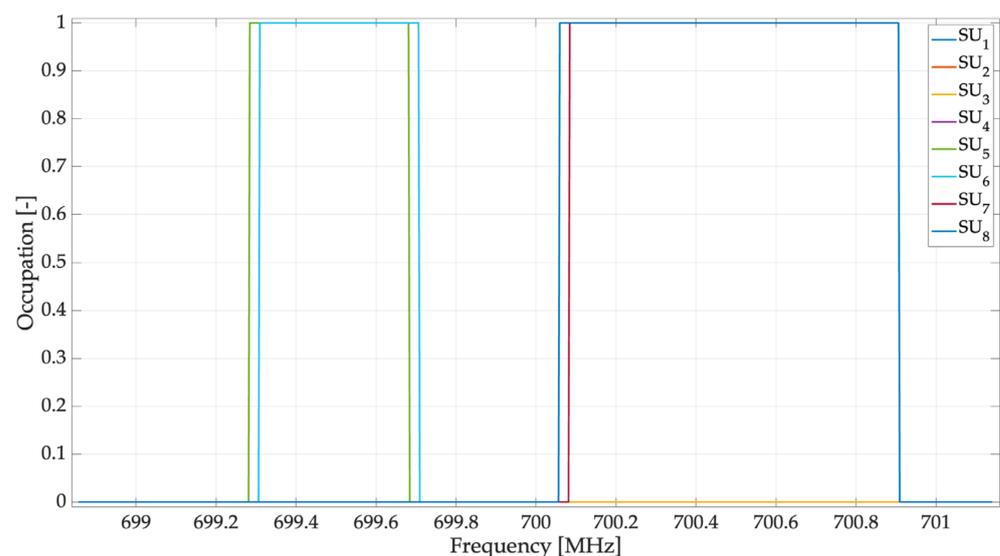
PUs and SUs were deployed in an area  $12 \times 12 \text{ m}^2$ , containing structures such as walls, doors, windows, columns, etc., as indicated in Figure 11.

## 6. Results

This section outlines the results of implementing the proposed methodology in a real wireless communication environment. They are divided into two subsections; the first corresponds to the central entity based on digital signal processing, and the second concerns the central entity based on NNs.

### 6.1. Results with a Central Entity Based on Digital Signal Processing

This section shows the results of 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 following important points:

**Figure 12.** Spectrum occupancy for each SU.

- There were SUs who failed to perceive both PUs;
- The central frequency  $F_C$  of each PU was perceived at a different frequency by each SU;
- The bandwidth size  $B$  for each PU varied for each SU;

- SUs that were the farthest from the PUs failed to detect both PUs.

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

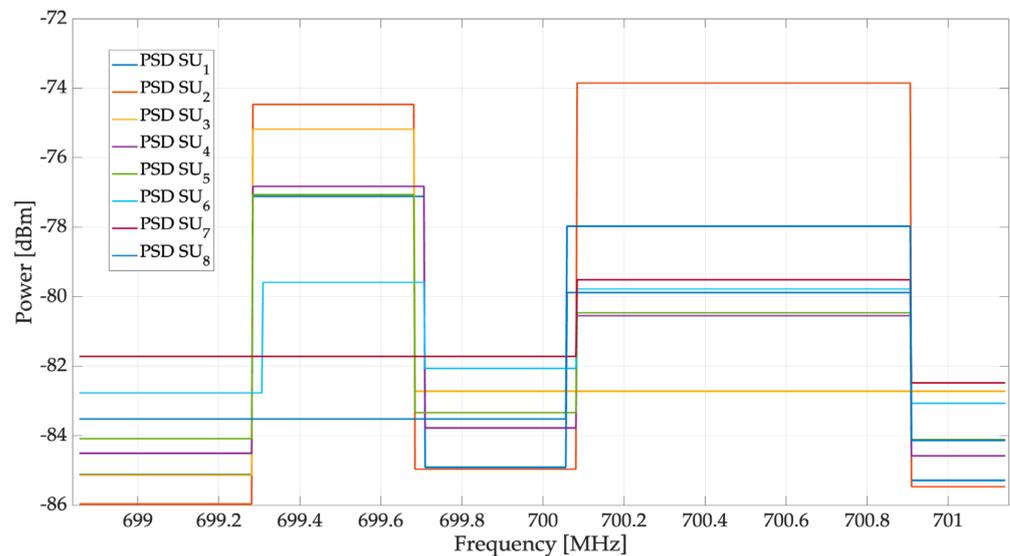


Figure 13. Approximated PSD for each SU.

Figure 14 shows the result of applying the module for estimating the average occupancy of each SU in addition to the result of the  $F_C$  and  $B$  estimation for each PU. The first value obtained was  $F_{C1} = 699.48$  MHz with a bandwidth of  $B_1 = 0.4$  MHz.  $F_{C2} = 700.49$  MHz with a bandwidth of  $B_2 = 0.825$  MHz was also obtained. Given that the exact values were  $F_{C1} = 699.5$  MHz,  $B_1 = 0.5$  MHz,  $F_{C2} = 700.5$  MHz, and  $B_2 = 1$  MHz, the values we obtained here show a good occupancy estimation of the MBSS method in conjunction with the central entity for a specific geographical zone covered by the PUs.

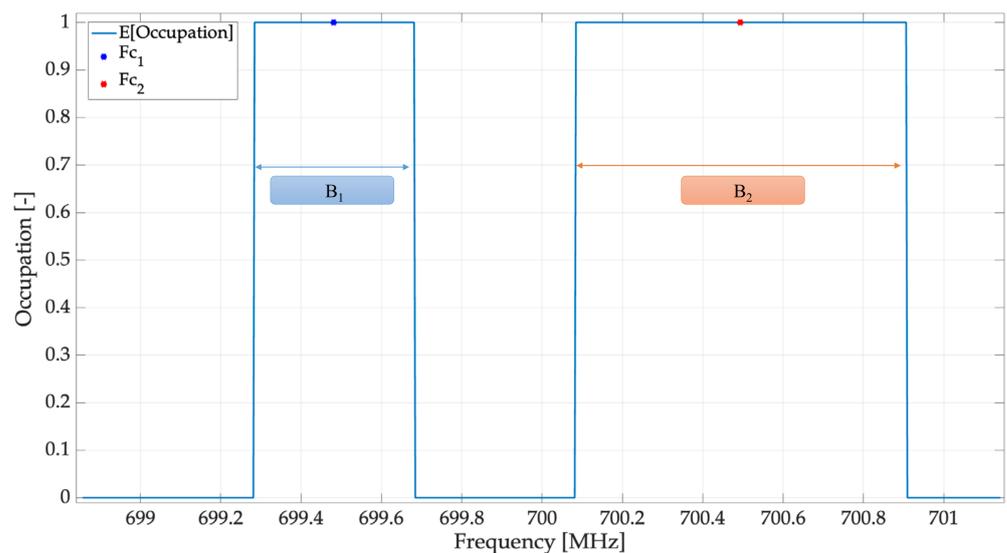
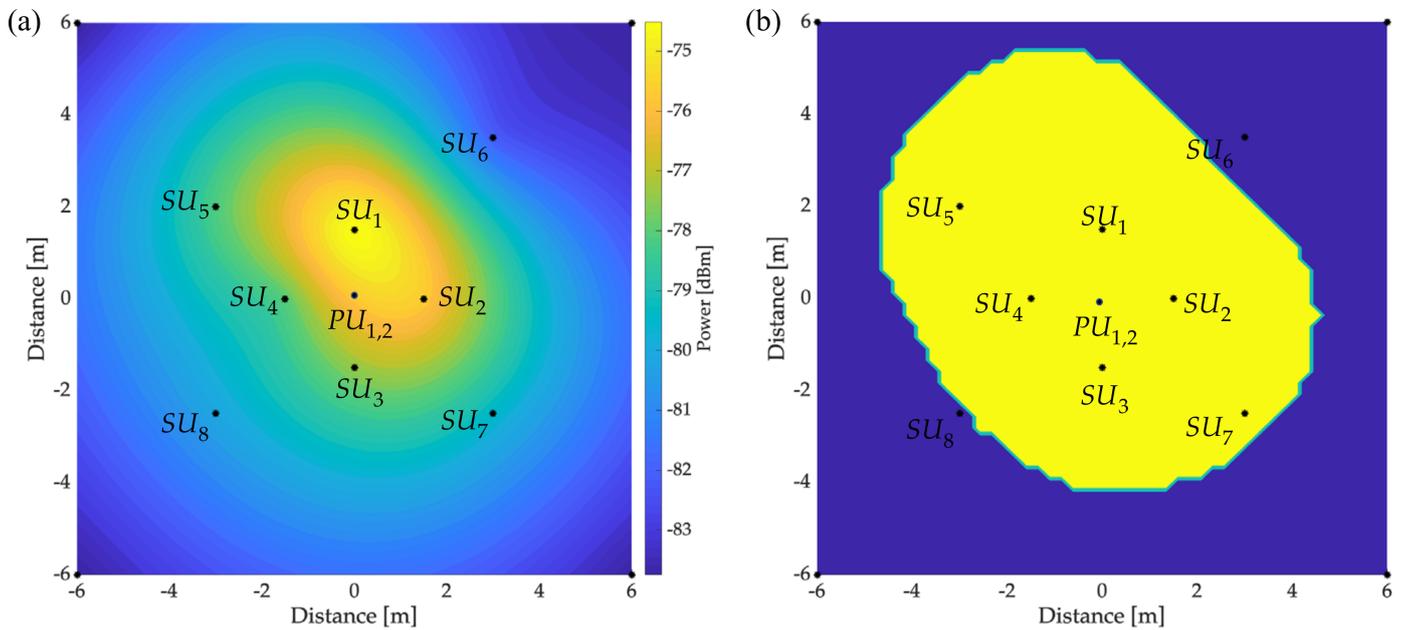


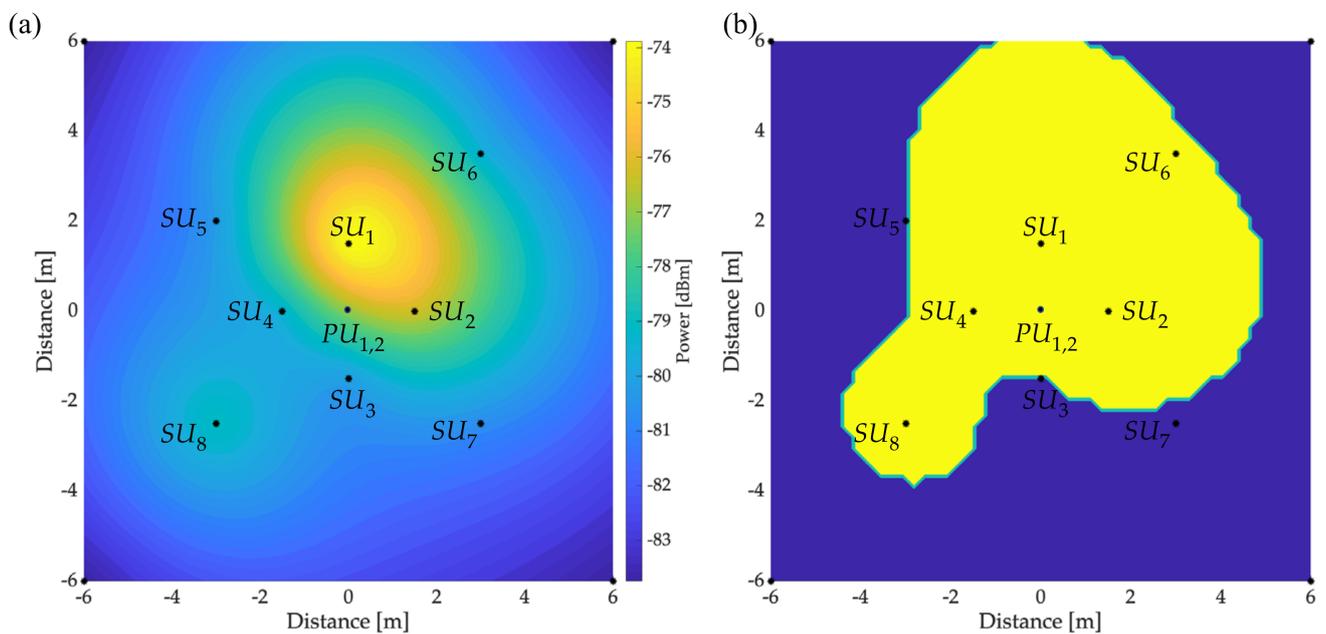
Figure 14. Average occupancy obtained by the central entity.

Figures 15a and 16a show the REM of  $PU_1$  and  $PU_2$ , respectively. These maps were created based on the information collected by the SUs in their different locations. Figures 15b and 16b show the respective areas occupied by  $PU_1$  and  $PU_2$ . Values of the occupied area for each PU are  $area_{PU_1} = 60.76 \text{ m}^2$  and  $area_{PU_2} = 56.39 \text{ m}^2$ . These results

were obtained by placing a threshold  $L = -80$  dBm in the obtained REM. In this way, the area with a power greater than this threshold was considered a space occupied by the PU.



**Figure 15.** Primary transmission in the 699.48 MHz band with a  $B$  of 0.4 MHz: (a) REM and (b) occupied area.



**Figure 16.** Primary transmission in the 700.49 MHz band with a  $B$  of 0.825 MHz: (a) REM and (b) occupied area.

Figure 15b shows that  $SU_6$  and  $SU_8$  did not observe the transmission of  $PU_1$  (i.e., both  $SU_6$  and  $SU_8$  did not appear in this area of influence). This effect was a result of the structural distribution in which the implementation of the real scenario occurred. Nevertheless, an expected collaboration between SUs might improve this result.

## 6.2. Results with a Central Entity Based on NNs

This section details the results obtained using a central entity and NNs. First, the NN training is presented, and then the results obtained by applying ML algorithms to this stage are shown.

### 6.2.1. Training

For the  $NN_1$  training,  $NN_2$  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. In addition, 9000 vectors of inputs and their corresponding outputs were used. To determine the best NN?, the number of layers that should be used, and how many neurons, 12 NNs were studied to carry out the work of  $NN_1$ ,  $NN_2$ , and  $NN_3$ . Each NN was considered with the number of layers  $la = 1, 2, 3, 4$  and with the number of neurons  $ne = 16, 32, 64$ . For each NN, there were 12 possible combinations between neurons and layers. They were performed to find the most convenient NN configuration for this work.

The results are presented for both the entity that uses digital signal processing and the one that uses the three NNs. As the first parameter, the training time used by each NN is shown in Table 4, indicating the time each NN spent in the training stage. Additionally, Figure 17 shows that the more neurons and layers an NN had, the longer its training time was. All the processing was carried out using the same computer (MacBook Pro with 8 GB RAM and a 1st-generation M1 processor), and the training time for each NN in its different versions tended to follow similar behavior.

**Table 4.** The training time of (a)  $NN_1$ , (b)  $NN_2$ , and (c)  $NN_3$ .

Training Time $NN_1$ [min]				
		Neurons		
(a)		16	32	64
Layers	1	2	10	60
	2	10	25	180
	3	120	192	1500
	4	162	600	3984
Training Time $NN_2$ [min]				
		Neurons		
(b)		16	32	64
Layers	1	2	6	16
	2	8	33	290
	3	15	80	850
	4	20	129	7610
Training Time $NN_3$ [min]				
		Neurons		
(c)		16	32	64
Layers	1	2	5	16
	2	5	19	280
	3	11	75	875
	4	21	145	3628

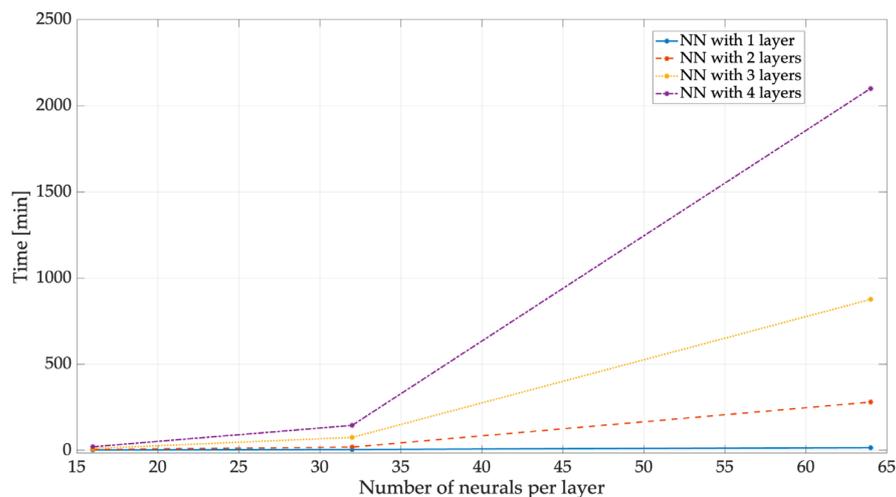


Figure 17. The training time for the different versions of the  $NN_3$ .

### 6.2.2. Statistics

This section analyzes the statistical results of the three NNs used in this proposal. To obtain these results, 27,000 entries were considered for each NN. Figure 18a presents the results of  $NN_1$ , responsible for granting the power value of the detected PUs, showing the power difference between the real values and those obtained with the different versions of  $NN_1$ . The difference between these values was, on average,  $-0.01$  dBm for each version of  $NN_1$ , highlighting the accuracy of the predictive ability of the NN-based approach. Figure 18b only shows the average value, indicating that the margin of error between the expected value and those given by the different NNs was very small; the values were practically the same.

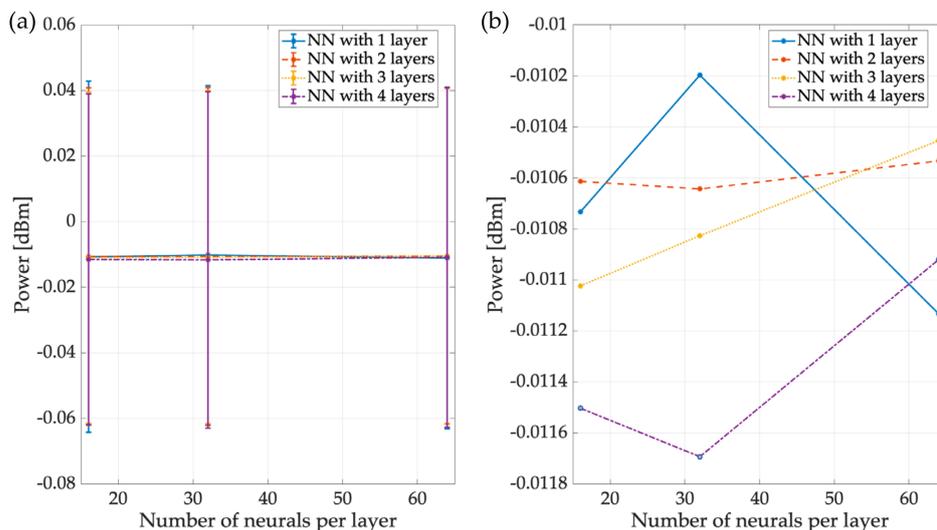
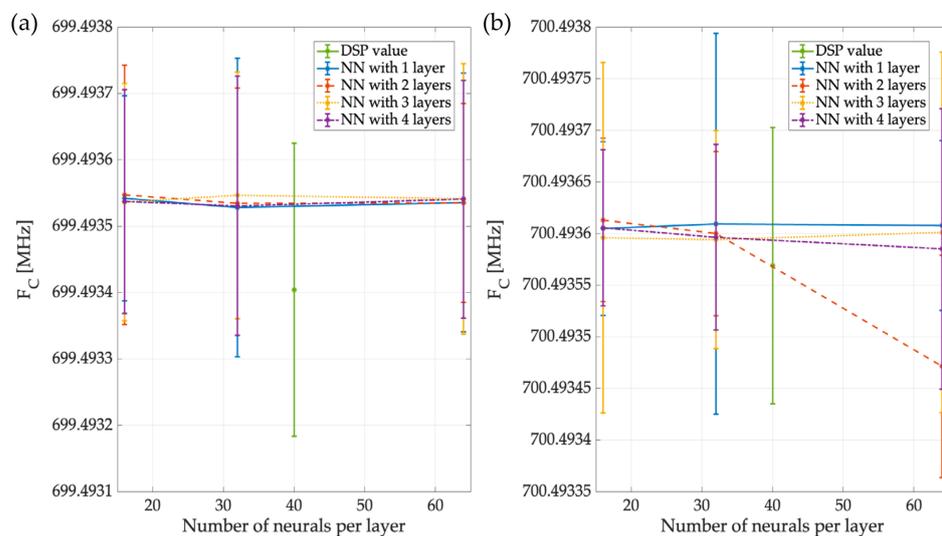


Figure 18. Difference between real values and those obtained by  $NN_1$ : (a) mean and standard deviation (STD) and (b) mean.

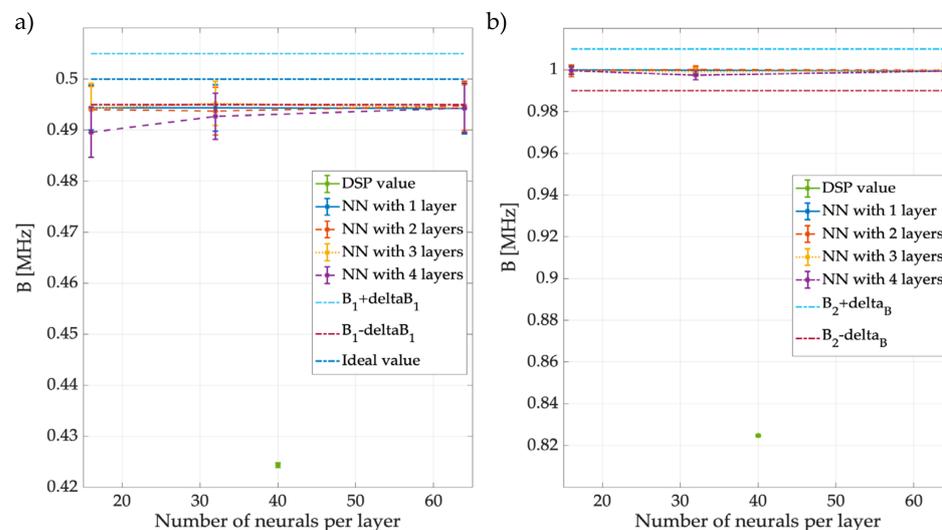
Figure 19 shows the result of  $NN_2$ , the network that gives the carrier value of each PU detected. In the proposed environment, the  $NN_2$  detected two PUs, and Figure 19a shows its precision in detecting each PU carrier, with  $F_{C1}$ , on average, at 699.4 MHz, regardless of the configuration of  $NN_2$ . Figure 19b shows the precision of  $NN_2$  in detecting the  $PU_2$  carrier, indicating that, on average, the carrier value was 700.494 MHz. Even though the network with 2 layers and 64 neurons per layer deviated slightly from the ideal

value (700.5 MHz) compared with the other NN configurations, it continued to provide reasonably accurate results. In both images, the value of applying digital signal processing can be seen in blue and is quite close to the ideal values of 699.4934 MHz and 700.4935 MHz for  $PU_1$  and  $PU_2$ , respectively, thus resulting in a comparable level of accuracy of the NN-based approach.



**Figure 19.** Mean and STD of the precision in detecting (a) the  $PU_1$  carrier of the different  $NN_2$  variants and (b) the  $PU_2$  carrier of the different networks.

Figure 20 presents the results of  $NN_3$ , the network in charge of giving the  $B$  that each detected PU in the spectrum frames. In this figure, the average  $B$  of  $PU_1$  is 0.48 MHz, regardless of the network configuration (see Figure 20a). Moreover, Figure 20b 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). The blue dot in the image indicates the result of digital signal processing, as  $B_1 = 0.425$  MHz and  $B_2 = 0.824$  MHz. For  $B_1$ , the margins of error/precision of DSP and NN were quite similar (only DSP overestimates while NN underestimates). For  $B_2$ , however, it seems that the NN was much more precise, suggesting that the method based on NNs tends to improve DSP or, in the worst case, provide a comparable absolute error.



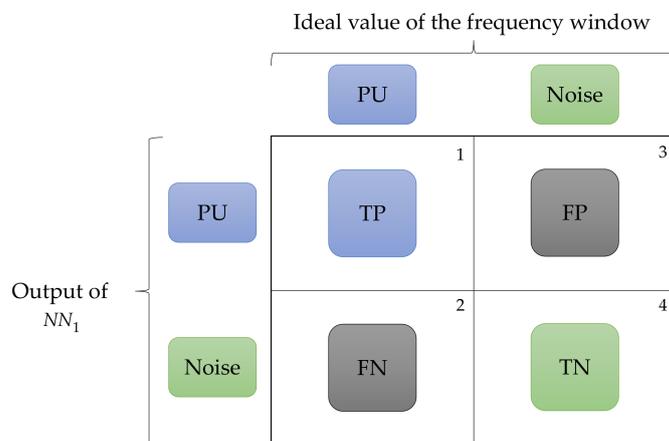
**Figure 20.** Mean and STD of the precision in detecting (a) the  $B_1$  of different  $NN_3$  variants and (b) the  $B_2$  of the different networks.

To measure the precision of this methodology, the F1 score ( $F1$ ) [48] was used as a common evaluation metric in the ML field for assessing the accuracy of binary classification models. 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 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  $F1$  value expresses the harmonic mean of precision and completeness, giving more weight to low values. The  $F1$  formula is

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

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  $F1$  metric is valuable for comparing different binary classification models and selecting the best model for a given classification task. To determine this parameter (i.e.,  $F1$ ), the following four possible cases were considered (see Figure 21):



**Figure 21.** Evaluation outputs of detected windows.

1. A window corresponding to a PU transmission and classified as such by the SU is considered a true positive (TP) value;
2. A frequency window corresponding to a PU transmission classified as noise by the SU is a false negative (FN) value;
3. A window corresponding to noise classified by the SU as a PU transmission is a false positive (FP) value;
4. A frequency window corresponding to noise classified by the SU as such is a true negative (TN) value.

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.

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

- The resulting  $NN_3 B$  matching the ideal bandwidth that corresponds to a transmission is a true positive (TP) case;
- The resulting  $NN_3 B$  corresponding to an ideal bandwidth close to zero is a true negative (TN) case;
- The resulting  $NN_3 B$  much greater than an ideal bandwidth that is close to zero is a false positive (FP) case;
- The resulting  $NN_3 B$  that should correspond to an ideal transmission  $B$  but is a value close to zero will be a false negative (FN) case.

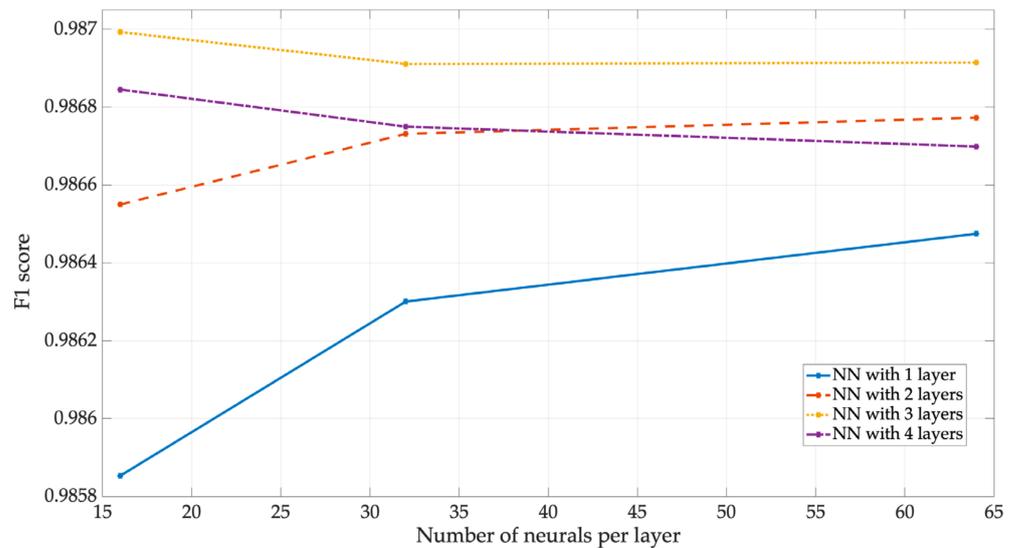


Figure 22. F1 score for NN<sub>1</sub>.

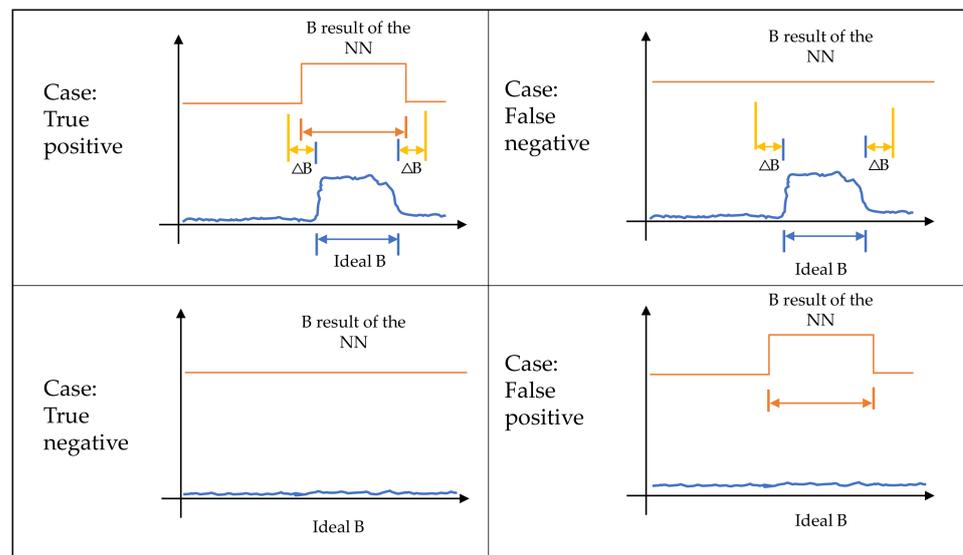
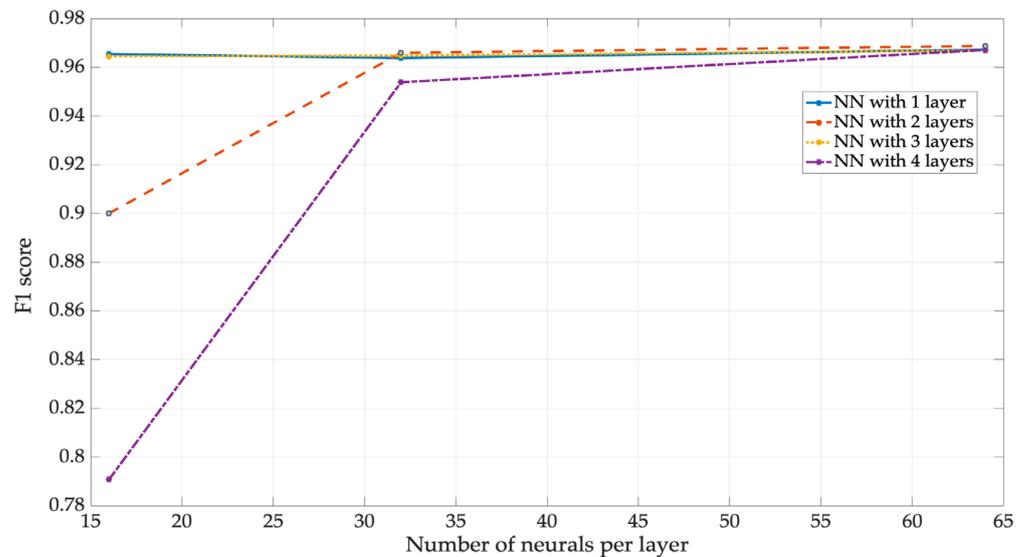


Figure 23. B assessment results.

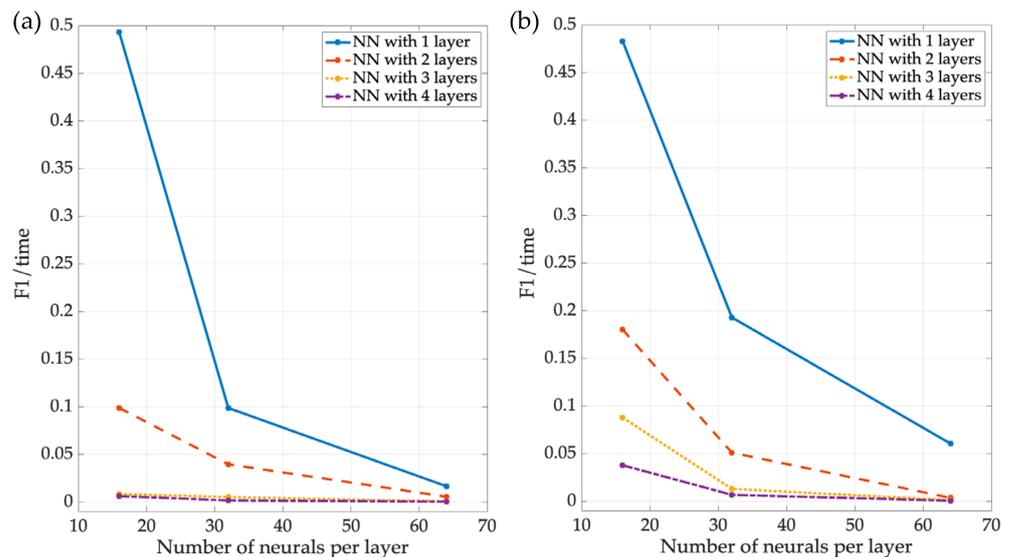
Figure 23 shows a  $\Delta B$  value, which is the difference between real and estimated bandwidths when there is a PU. When this parameter tends to grow, it provides flexibility so that the resulting NN<sub>3</sub> B matches an ideal B that corresponds to one transmission. However, when the value of  $\Delta B$  is very small, the system becomes more inaccurate, since it only detects as TP those values that are close to the ideal value.

Figure 24 shows the F1 of NN<sub>3</sub>. The NN with four layers and the NN with two layers both had an undesirable performance. This result could be explained by observing Figure 20, which shows the result of  $B_1$  is very close to the minimum value ( $ideal\_value - \Delta B$ ). Nevertheless, this undesirable performance was observed only when a low number of neurons per layer (i.e., 16) were employed. By properly configuring the number of neurons per layer to a sufficiently high value (e.g., 32 or greater), accurate performance with F1 values around 0.95 and 0.96 was obtained, as well as with two and four layers in the NN. Figure 24 also suggests overfitting in the NNs with two and four layers. This is because the models for the 2-layer and 4-layer NNs, both with 16 neurons per layer, fit too well to the training data and, as a result, do not generalize well to the new data. Despite this, they have an F1 of 0.9 and 0.79, respectively.



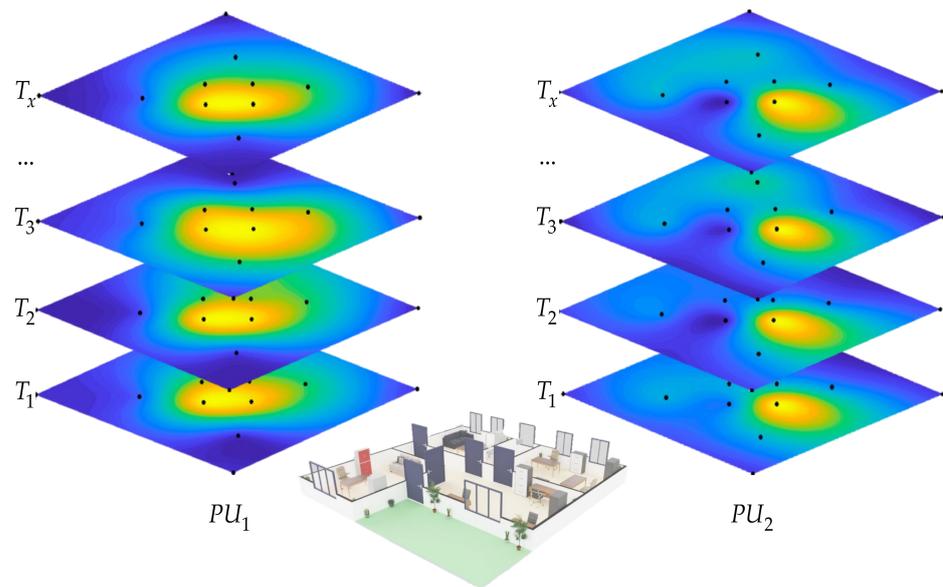
**Figure 24.** F1 of  $NN_3$ .

Figure 25 shows the ratio of F1 scores between training time for  $NN_1$  and  $NN_3$  (Figure 25a and 25b, respectively). The value obtained from this relationship indicates which NN provides the best trade-off based on the F1 accuracy result and the time invested in training the network to attain such accuracy. As can be noticed, using an NN with a single layer provides the best F1 performance for the 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 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 over time, during which the sensed spectrum was analyzed in a specific geographic area and where SUs collaborated to obtain the REM of the different detected PUs. As can be appreciated, the proposed methodology was able to 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.

## 7. Conclusions

In this work, a real wireless communication scenario was implemented to detect the occupancy of multiple PUs through several SUs via a central entity, permitting the determination of the area used by the PUs based on REM estimations. Apart from the REM constructions, other features, such as power, bandwidth, and central frequency of possible detected PUs from the multiband spectrum frames, were estimated by the considered SUs.

For this task, we proposed using neural networks to substitute the classical digital signal processing used in our preliminary work. It was expected that the performance and processing time would be faster. Clearly, the training stage of the NNs, as shown by the results, was a factor to be considered. In this work, higher precision was preferred to locate the PUs and less processing time for the central entity when using the NNs. However, it must be considered that NN training time was not negligible and could even be high, depending on the number of layers and neurons each NN contained. The difference between the real values and that of the NNs was minimal, and it could even be said they were practically the same. In some cases, the NN showed even better results than the digital signal processing, for instance, when detecting the PUs' carriers.

Neural networks are a powerful and useful tool in many applications but are not always the best option. In some cases, classical digital signal processing may be sufficient, while in others, neural networks can significantly improve performance and accuracy. Future work will aim to determine the optimal number of SUs needed to obtain the area occupied by the PUs, thus avoiding the excessive processing of the central entity.

**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 Humanities, Science, and Technology (CONAHCyT), Grant (no. 490180).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing not applicable.

**Acknowledgments:** The authors are grateful to CONAHCYT, 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. Mitola, J.; Maguire, G.Q. Cognitive radio: Making software radios more personal. *IEEE Pers. Commun.* **1999**, *6*, 13–18. [\[CrossRef\]](#)
2. Akyildiz, I.F.; Lee, W.-Y.; Vuran, M.C.; Mohanty, S. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Comput. Netw.* **2006**, *50*, 2127–2159. [\[CrossRef\]](#)
3. Hattab, G.; Ibnkahla, M. Multiband Spectrum Access: Great Promises for Future Cognitive Radio Networks. *Proc. IEEE* **2014**, *102*, 282–306. [\[CrossRef\]](#)
4. El-Khamy, S.E.; El-Mahallawy, M.S.; Youssef, E.S. Improved wideband spectrum sensing techniques using wavelet-based edge detection for cognitive radio. In Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC), San Diego, CA, USA, 28–31 January 2013; pp. 418–423. [\[CrossRef\]](#)
5. Kumar, A.; Saha, S.; Bhattacharya, R. Wavelet transform based novel edge detection algorithms for wideband spectrum sensing in CRNs. *AEU-Int. J. Electron. Commun.* **2018**, *84*, 100–110. [\[CrossRef\]](#)
6. Diao, X.; Dong, Q.; Yang, Z.; Li, Y. Double-Threshold Cooperative Spectrum Sensing Algorithm Based on Sevcik Fractal Dimension. *Algorithms* **2017**, *10*, 96. [\[CrossRef\]](#)
7. Popoola, J.J.; van Olst, R. Application of neural network for sensing primary radio signals in a cognitive radio environment. In Proceedings of the IEEE Africon '11, Victoria Falls, Zambia, 13–15 September 2011; pp. 1–6. [\[CrossRef\]](#)

8. Shamsi, N.; Mousavinia, A.; Amirpour, H. A channel state prediction for multi-secondary users in a cognitive radio based on neural network. In Proceedings of the 2013 International Conference on Electronics, Computer and Computation (ICECCO), Ankara, Turkey, 7–9 November 2013; pp. 200–203. [CrossRef]
9. Molina-Tenorio, Y.; Prieto-Guerrero, A.; Aguilar-Gonzalez, R. A Novel Multiband Spectrum Sensing Method Based on Wavelets and the Higuchi Fractal Dimension. *Sensors* **2019**, *19*, 1322. [CrossRef] [PubMed]
10. Zaidawi, D.J.; Sadkhan, S.B. Blind Spectrum Sensing Algorithms in CRNs: A Brief Overview. In Proceedings of the 2021 7th International Engineering Conference “Research & Innovation amid Global Pandemic” (IEC), Erbil, Iraq, 24–25 February 2021; pp. 78–83. [CrossRef]
11. Molina-Tenorio, Y.; Prieto-Guerrero, A.; Aguilar-Gonzalez, R. Real-Time Implementation of Multiband Spectrum Sensing Using SDR Technology. *Sensors* **2021**, *21*, 3506. [CrossRef] [PubMed]
12. Politis, C.; Maleki, S.; Duncan, J.M.; Krivochiza, J.; Chatzinotas, S.; Ottesten, B. SDR Implementation of a Testbed for Real-Time Interference Detection with signal cancellation. *IEEE Access* **2018**, *6*, 20807–20821. [CrossRef]
13. Stewart, R.W.; Crockett, L.; Atkinson, D.; Barlee, K.; Crawford, D.; Chalmers, I.; McLernon, M.; Sozer, E. A low-cost desktop software defined radio design environment using MATLAB, simulink, and the RTL-SDR. *IEEE Commun. Mag.* **2015**, *53*, 64–71. [CrossRef]
14. Hiari, O.; Mesleh, R. A Reconfigurable SDR Transmitter Platform Architecture for Space Modulation MIMO Techniques. *IEEE Access* **2017**, *5*, 24214–24228. [CrossRef]
15. Santos-Luna, E.; Prieto-Guerrero, A.; Aguilar-Gonzalez, R.; Ramos, V.; Lopez-Benitez, M.; Cardenas-Juarez, M. A Spectrum Analyzer Based on a Low-Cost Hardware-Software Integration. In Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 17–19 October 2019; pp. 0607–0612. [CrossRef]
16. Aghabeiki, S.; Hallet, C.; Noutehou, N.E.-R.; Rassem, N.; Adjali, I.; Mabrouk, M.B. Machine-learning-based spectrum sensing enhancement for software-defined radio applications. In Proceedings of the 2021 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAS), Cleveland, OH, USA, 21–23 June 2021; pp. 1–6. [CrossRef]
17. Selva, A.F.B.; Reis, A.L.G.; Lenzi, K.G.; Meloni, L.G.P.; Barbin, S.E. Introduction to the Software-defined Radio Approach. *IEEE Lat. Am. Trans.* **2012**, *10*, 1156–1161.
18. About RTL-SDR, rtl-sdr.com, 11 de Abril de 2013. Available online: <https://www.rtl-sdr.com/about-rtl-sdr/> (accessed on 8 August 2022).
19. Koutlia, K.; Bojović, B.; Lagén, S.; Giupponi, L. Novel radio environment map for the ns-3 NR simulator. In Proceedings of the Workshop on ns-3, Virtual Event USA: ACM, New York, NY, USA, 23–24 June 2021; pp. 41–48. [CrossRef]
20. Li, J.; Ding, G.; Zhang, X.; Wu, Q. Recent Advances in Radio Environment Map: A Survey. In *Machine Learning and Intelligent Communications*; Gu, X., Liu, G., Li, B., Eds.; En Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Springer International Publishing: Cham, Switzerland, 2018; Volume 226, pp. 247–257. [CrossRef]
21. Spooner, C.M.; Khambekar, N.V. Spectrum sensing for cognitive radio: A signal-processing perspective on signal-statistics exploitation. In Proceedings of the 2012 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 30 January–2 February 2012; pp. 563–568. [CrossRef]
22. Zhang, X.; Zhao, Y.; Chen, H. Adaptive Compressed Wideband Spectrum Sensing Based on Radio Environment Map Dedicated for Space Information Networks. In *Wireless and Satellite Systems*; Jia, M., Guo, Q., Meng, W., Eds.; En Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Springer International Publishing: Cham, Switzerland, 2019; pp. 128–138. [CrossRef]
23. Santana, Y.H.; Plets, D.; Alonso, R.M.; Nieto, G.G.; Martens, L.; Joseph, W. Radio Environment Map of an LTE Deployment Based on Machine Learning Estimation of Signal Levels. In Proceedings of the 2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Bilbao, Spain, 15–17 June 2022; pp. 1–6. [CrossRef]
24. Borisov, V.; Leemann, T.; Sessler, K.; Haug, J.; Pawelczyk, M.; Kasneci, G. Deep Neural Networks and Tabular Data: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**. early access. [CrossRef] [PubMed]
25. Xu, M.; Yin, Z.; Zhao, Y.; Wu, Z. Cooperative Spectrum Sensing Based on Multi-Features Combination Network in Cognitive Radio Network. *Entropy* **2022**, *24*, 129. [CrossRef] [PubMed]
26. Lam, N.S.-N. Spatial Interpolation Methods: A Review. *Am. Cartogr.* **1983**, *10*, 129–150. [CrossRef]
27. Burrough, P.A.; McDonnell, R.; Burrough, P.A. *Principles of Geographical Information System*; en Spatial Information Systems; Oxford University Press: Oxford, UK, 1998.
28. Harman, B.I.; Koseoglu, H.; Yigit, C.O. Performance evaluation of IDW, Kriging and multiquadric interpolation methods in producing noise mapping: A case study at the city of Isparta, Turkey. *Appl. Acoust.* **2016**, *112*, 147–157. [CrossRef]
29. Arseni, M.; Voiculescu, M.; Georgescu, L.P.; Iticescu, C.; Rosu, A. Testing Different Interpolation Methods Based on Single Beam Echosounder River Surveying. Case Study: Siret River. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 507. [CrossRef]
30. Matheron, G. Principles of geostatistics. *Econ. Geol.* **1963**, *58*, 1246–1266. [CrossRef]
31. Oliver, M.A.; Webster, R. Kriging: A method of interpolation for geographical information systems. *Int. J. Geogr. Inf. Syst.* **1990**, *4*, 313–332. [CrossRef]

32. Gundogdu, K.S.; Guney, I. Spatial analyses of groundwater levels using universal kriging. *J. Earth Syst. Sci.* **2007**, *116*, 49–55. [[CrossRef](#)]
33. Cressie, N.A.C. *Statistics for Spatial Data, Revised ed.*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2015.
34. Isaaks, E.H.; Srivastava, R.M. *Applied Geostatistics*; Oxford University Press: New York, NY, USA, 1989.
35. Han, J.; Kamber, M. *Data Mining: Concepts and Techniques*, 3rd ed.; Elsevier: Burlington, MA, USA, 2012.
36. Wang, W.; Yang, Y. Development of convolutional neural network and its application in image classification: A survey. *Opt. Eng.* **2019**, *58*, 1. [[CrossRef](#)]
37. Haykin, S.S.; Haykin, S.S. *Neural Networks and Learning Machines*, 3rd ed.; Prentice Hall: New York, NY, USA, 2009.
38. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
39. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)] [[PubMed](#)]
40. Molina-Tenorio, Y.; Prieto-Guerrero, A.; Aguilar-Gonzalez, R. Multiband Spectrum Sensing Based on the Sample Entropy. *Entropy* **2022**, *24*, 411. [[CrossRef](#)] [[PubMed](#)]
41. Molina-Tenorio, Y.; Prieto-Guerrero, A.; Aguilar-Gonzalez, R.; Ruiz-Boqué, S. Machine Learning Techniques Applied to Multiband Spectrum Sensing in Cognitive Radios. *Sensors* **2019**, *19*, 4715. [[CrossRef](#)]
42. Nooelec-Nooelec NESDR SMARt v4 SDR-Premium RTL-SDR w/Aluminum Enclosure, 0.5PPM TCXO, SMA Input. RTL2832U & R820T2-Based-Software Defined Radio. Available online: <https://www.nooelec.com/store/sdr/nesdr-smart-sdr.html> (accessed on 8 March 2021).
43. HackRF One-Great Scott Gadgets, 8 de Marzo de 2021. Available online: <https://greatscottgadgets.com/hackrf/one/> (accessed on 8 March 2021).
44. LimeSDR Mini is a \$135 Open Source Hardware, Full Duplex USB SDR Board (Crowdfunding)-CNX Software, CNX Software-Embedded Systems News, 18 de Septiembre de 2017. Available online: <https://www.cnx-software.com/2017/09/18/limesdr-mini-is-a-135-open-source-hardware-full-duplex-usb-sdr-board-crowdfunding/> (accessed on 13 March 2022).
45. Welch, P. 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.* **1967**, *15*, 70–73. [[CrossRef](#)]
46. Mallat, S.G. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 674–693. [[CrossRef](#)]
47. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
48. Sasaki, Y. The Truth of the F-Measure, oct. 2007 [En Línea]. Available online: <https://www.cs.odu.edu/~mukka/cs795sum11dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf> (accessed on 7 April 2007).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.