future internet

MDPI

Article

Multiple PUE Attack Detection in Cooperative Mobile Cognitive Radio Networks

Ernesto Cadena Muñoz *, Gustavo Chica Pedraza * and Alexander Aponte Moreno

School of Telecommunications Engineering, Universidad Santo Tomás, Bogotá 110311, Colombia; jhonaponte@usta.edu.co

* Correspondence: ernestocadena@usta.edu.co (E.C.M.); gustavochica@usantotomas.edu.co (G.C.P.)

Abstract: The Mobile Cognitive Radio Network (MCRN) are an alternative to spectrum scarcity. However, like any network, it comes with security issues to analyze. One of the attacks to analyze is the Primary User Emulation (PUE) attack, which leads the system to give the attacker the service as a legitimate user and use the Primary Users' (PUs) spectrum resources. This problem has been addressed from perspectives like arrival time, position detection, cooperative scenarios, and artificial intelligence techniques (AI). Nevertheless, it has been studied with one PUE attack at once. This paper implements a countermeasure that can be applied when several attacks simultaneously exist in a cooperative network. A deep neural network (DNN) is used with other techniques to determine the PUE's existence and communicate it with other devices in the cooperative MCRN. An algorithm to detect and share detection information is applied, and the results show that the system can detect multiple PUE attacks with coordination between the secondary users (SUs). Scenarios are implemented on software-defined radio (SDR) with a cognitive protocol to protect the PU. The probability of detection (PD) is measured for some signal-to-noise ratio (SNR) values in the presence of one PUE or more in the network, which shows high detection values above 90% for an SNR of -7dB. A database is also created with the attackers' data and shared with all the SUs.

Keywords: cognitive radio network; spectrum management; primary user attack; network security

Citation: Cadena Muñoz, E.; Chica Pedraza, G.; Aponte Moreno, A. Multiple PUE Attack Detection in Cooperative Mobile Cognitive Radio Networks. *Future Internet* **2024**, *16*, 456. https://doi.org/10.3390/fi16120456

Academic Editors: Jose I. Moreno Novella and Yuezhi Zhou

Received: 4 September 2024
Revised: 24 October 2024
Accepted: 29 November 2024
Published: 4 December 2024

1. Introduction

Emerging telecommunication technologies have become a crucial part of our daily lives, whether for work, study, or entertainment. Nevertheless, there is an increasing need for more bandwidth, especially in wireless networks. Cognitive radio networks (CRNs) are an alternative to increase performance and use the frequency spectrum, whereas the primary user (PU) does not use the spectrum at specific frequencies and times. A secondary user (SU) uses the resources while there is no PU in the CRN, increasing the spectrum usage and helping with the spectrum scarcity problem. By enabling dynamic spectrum access, cognitive radios significantly enhance spectrum utilization and improve the performance of wireless networks [1].

A CRN operates on the principle of spectrum sensing, where cognitive devices continuously monitor the spectrum to detect unused frequencies, and there is a vulnerability that attackers can use because decisions are based on that sensing. The CRN is vulnerable to some attacks, mainly because there is an SU in the system, and they can be malicious users. The CRN is also susceptible to traditional attacks. Nevertheless, new attacks can affect functions like spectrum sensing, mobility, management, or sharing, the main aspects of the CRN, causing network malfunctions, affecting PU and SUs, and decreasing spectrum availability. The malicious users obtain access to a frequency and can use it to communicate or influence the network [2].

One attack affecting the CRN is the Primary User Emulation attack. An adversary intentionally mimics the signal of a PU, misleading SUs into believing that a licensed user is

occupying the frequency band. This forces the SU to release the channel, even when it is not an authorized user. The PUE can lead to inefficient spectrum utilization, as secondary users are unnecessarily displaced from frequencies that could otherwise be used. The disruption caused by PUE results in degraded quality of service (QoS) and connectivity issues for PU [3].

Multiple SUs sense the PU spectrum in a cooperative spectrum-sensing (CSS) network to counter PUE attacks. This increases the sensing capabilities and coverage and can work with a centralized or decentralized architecture. In the first one, the local sensing data or PU presence decisions are sent to a fusion center (FC), which makes a global decision of PUE presence based on the data. It is more effective for the CSS than a single SU for sensing, and it has been used against selfish PUE attacks, but the malicious user can attack with different strategies. The first method used to analyze the users was to measure the energy. Nevertheless, with time, attackers learn to change power parameters, making it difficult to recognize an attack with an energy detector [4].

Research into counteracting PUE attacks includes strategies like advanced spectrum-sensing algorithms in a single or cooperative environment, authentication protocols, and intrusion detection systems. These countermeasures aim to enhance the reliability of spectrum sensing, ensure the integrity of primary user identification, and minimize the impact of malicious interference. Evaluating the effectiveness of these strategies in different network environments and attack scenarios is a critical aspect of advancing the security of cognitive radio networks. Various PUEA prevention strategies are suggested for a PU working as a TV transmitter, mobile FM microphones, and methods like Fenton Approximation, Fingerprint Verification, or applying artificial intelligence (AI) techniques like ANN [5].

Although some techniques to detect PUE have been implemented, obtaining different detection performances, the literature does not include papers that work with multiple PUE attacks in a single environment and which are implemented on radio devices. This paper proposes an AI technique for detecting multiple PUE attacks in a cooperative CSS MCRN with SDR devices such as a PU, SU, and MU.

The previous work is shown in Section 2 of this paper; the methodology is explained in Section 3; the experiments are defined in Section 4; Section 5 analyzes the results; and Section 6 presents the discussion and conclusions.

2. Previous Work

A CRN represents a transformative advancement in wireless communication technology; it can work with mobile technology and IoT systems. These networks leverage cognitive radio technology, allowing the devices to change their operating parameters based on the surrounding radio environment measures; devices continuously sense the spectrum frequencies, making them vulnerable to some attacks. Even if there are spectrum holes or white spaces, the SU needs to check the spectrum to prevent the PU affectation; if there is a free space, it can use it for its purposes, taking advantage of dynamic spectrum management. However, intruders can use Denial of Service (DoS) attacks and interfere with a PU's communication services differently [6].

The PUE attacks are a type of DoS when an SU mimics the PU signal characteristics with two main objectives: selfish or malicious. Selfish is when it takes the spectrum for communications and malicious is when it gains spectrum resources to interrupt other SU or PU communications, acting like a jammer. The interference is so high that even the PU cannot use the spectrum even if they are paying for the license. The key to the attack being effective is to imitate the PU signal; depending on the service, it can be possible to delay discovering if there is a genuine PU or a PUE attack. The first attacks in the literature tended to increase the power, but attacks have changed this parameter to confuse the spectrum-sensing process [7].

Previous research into detecting PUE attacks in CRN has focused on distinguishing between legitimate PU and malicious users (MUs). The PU can be mobile, sensors, or

IoT devices depending on the services. The emulators have been designed to capture the parameters or patterns of the signal and reproduce these later to gain access to the spectrum, with this being unauthorized access to the resources. In MCRN, network base stations are distributed and include critical user data of power, distance, or frequency. By capturing this information, the emulator creates a copy of it. To detect it, a series of techniques have been designed for the simplest to the most complex systems. AI techniques like K-Nearest Neighbors (KNN) and Artificial Neural Networks (ANNs) are used to classify the users and determine the PUE attack. They depend on a database with user information and try to classify the attack by comparing the data. The channel is released if a PUE is detected, and other PUs or SUs can use it [8].

A PUE attack can also be addressed by disconnecting the MU from the base station with a classification process. The KNN classifier has been employed to identify the MU using data rate, distance from the base station, power received, and frequency, and the results are similar to those of an ANN trained with the same data. To increase security, Elliptic Curve Cryptography (ECC) is proposed for data encryption, showing high accuracy in detecting the PUE. In MCRN, base stations work with mobile subscribers or a PU to optimize the use of radio resources and manage frequency bands. PUEA detection and classification are performed at the base station, which maintains historical data about connected users. Various classifiers can be combined with a single-user energy detector. The KNN classifier is then used to analyze the behavior of the parameters, and network security is enhanced through ECC encryption [8].

For some PUE attackers, countermeasures using deep learning techniques have been developed, knowing that some information is needed for the training part of the algorithms. The authors use deep learning with the Extreme Value Theory (EVT) technique, which increases the detection of PUE attacks without previous information needed in the training part. This helps to discover the legitimate PU and recognize the MU in the network. It works in two parts: the first one needs to extract the features of the known PU, and the other analyzes the unknown user's features. There is a dependence on EVT and Weibull distributions to make the decisions, and it needs to know the PU parameters for the classification process [9].

Another way to address a PUEA is to explore intruder classification and focus on its impact on the CRN system. It uses a two-level auxiliary database that, as an input, has the energy consumption and verifies the device's location. It uses a method to reserve some channels for handover. Another intrusion detection uses the real PU locations and the received power. This traditional technique is based on a fixed PU position but does not work with a mobile user. It analyzes the signal parameters, calculates the energy based on the power, and locates the user with the help of a sensor network, indicating the need for some additional nodes to calculate it. Then, it decides whether a PUE is present in the network. It needs the precise position of the PU to operate adequately, and it does not work with an MCRN. Based on this, a practical machine-learning-based intrusion detection approach is used with some additional nodes using a comparison based on accuracy and precision and showing a high performance compared to previous techniques [10].

Two strategies have been used for spectrum sensing in the presence of PUE attacks: local sensing and the CSS. Local sensing can use match filter, energy, features, eigenvalue, or wavelet detection, while the CSS is based on centralized, distributed, or relay-assisted cooperation. The centralized base station or fusion center is used to communicate with all the SUs through a common control channel and share global decisions with all the SUs, decreasing the impact of the hidden node; this solution is used with the MCRN. In the distributed, it only communicates with another node to help in the decision, and the relay assistant uses a cognitive user to sense the channel and use an independent channel to communicate, helping in the decision of the PUE presence [7].

Previous work has highlighted that the requirement for feature extraction in traditional machine learning techniques can limit the full potential of raw data. Then, a framework with one-dimensional deep learning is used to identify PUE attacks. It uses a CR envi-

ronment with a PU node, an SU node, and a PUE attacker. This allows for spectrum exploitation if there is a frequency space with the PUE. By implementing a one-dimensional CNN (1D-CNN), there is no need for explicit feature extraction. It enhances detection performance for PUE and jamming attacks (JAs), eliminating the dependence on feature extraction for PUE detection. The performance is evaluated by analyzing the ROC curves of the false alarm probability, and the results are better than those of the feature characteristics selection technique [6].

Recurrent Neural Networks (RNNs) have been used to detect PUE attacks when signal activity patterns can be recognized or associated with PU signal behavior, and it depends on past states. It leverages the temporal dependencies in time series data to increase prediction accuracy and identify suspicious activities. To address the gradient vanishing problem associated with RNNs, an advanced RNN variant incorporating Long Short-Term Memory (LSTM) units was introduced. This improves the analysis and processing of large datasets and significantly increases the performance of PUE attack detectors [11].

Another approach has focused on improving the security of the cognitive sensing part process using a hybrid Genetic Artificial Bee Colony (GABC) algorithm. This is useful for detecting PUE and shows a promising increase in detection performance. It combines Genetic Algorithm (GA) operators with the Artificial Bee Colony (ABC) algorithm, refining the detection process and increasing the likelihood of accurate attack identification. It balances exploration and exploitation, seeking an optimal solution and optimizing spectrum sensing [12].

In mobile networks, the challenge of managing connections among numerous CR-enabled 5G massive machine-type communications (mMTC) devices was studied. The PUE attack was highlighted as a critical threat undermining network performance in extensive CR-based 5G Internet of Things (5G-IoT) frameworks. This attack results in significant bandwidth wastage and substantial interference with primary user (PU) transmissions. A novel detection methodology with two key components was introduced to address this. First, secondary users (SUs) determine and share the transmitter's location with other authorized SUs. This localization is achieved by evaluating received signal strength (RSS) and integrating the spatial coordinates of neighboring receivers. Second, during the sensing phase, each SU collects and archives signal energy vectors in a database. When a new node is detected, its energy vector is matched against the stored vectors to discern potential malicious entities. Adversarial detection hinges on the degree of similarity between the newly acquired energy vector and those in the database. This approach effectively mitigates the missed detection rate of PU, thereby alleviating interference with PU communications [13].

Finally, with a single PUE attack, an approach includes a countermeasure strategy that leverages the entropy of signal data, DNN, and CSS to detect the attacks effectively. It creates a blacklist feature within the FC to save the MU data systematically. The proposed system is simulated and tested using an SDR testbed, demonstrating an improved method with the ability to detect PUE attacks. Furthermore, it facilitates the recording of attack-related data for future analysis and ensures the sharing of these data among all SUs within the network [14].

Although various techniques exist to detect PUE attacks, even with AI, only a few papers have implemented algorithms in MCRN devices for a single PUE attack. Nevertheless, in a natural environment, multiple PUE attacks on the network need to be studied, especially in a CSS scenario where a common control channel is required to share information and make a global decision about the presence of PUE.

3. DNN Detection of Multiple PUE Attacks

A PUE attack occurs when an attacker pretends to be a legitimate PU of a spectrum band in a cognitive radio network. The attack's goal is to deceive SUs into believing that a PU is active, which can result in SUs vacating the channel unnecessarily. This manipulation can lead to inefficient spectrum utilization and may enable attackers to monopolize

the channel for their use. Interfering with all devices and not sending data can be malicious PUE and selfish PUE if the attacker wants the channel for its own communications purposes [13].

As can be seen in Figure 1, a Primary Base Station (PBS) communicates with the PUs, and the cognitive base station (CBS) communicates with the SUs when a PU frequency is not used. If a PU appears, the system must release the channel. The PUE attack mimics the PU information and behavior, and the CBS and SUs recognize it as an authentic PU and release the channel.

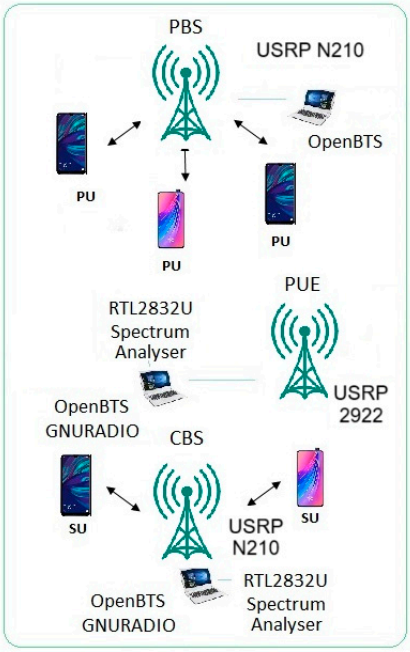

Figure 1. PUE attack scenario (source: own).

The proposed methodology for detecting multiple PUE attacks uses a DNN to decide the PUE's presence in the network based on the communication with the SUs. It starts with an energy detector, allowing the network to detect a PU/¨PUE and release the channel to protect users. It automatically starts an authentication process that, in combination with the energy detector, allows the system to classify if there is a PU or a PUE. This information is calculated on each SU and shared with the FC that takes the global decision that is shared along with the PUE data to all the SUs as a blacklist. In the presence of multiple malicious PUE attacks in the environment, the algorithm continues working on each SU, continuously transmitting the detection signal and data to the FC.

In the MCRN, each PUE attack can be seen as a base station that can manage malicious users (MUs). An MU can be identified as a PUE attack using the Downlink (DL) channel and not the Uplink (UL) channel. This information, combined with the authentication process information, allows the system to detect and share a single PUE attack. Each MU continuously senses the spectrum to select the frequency of the attack. In this case, it is looking for an empty channel, but it can even transmit signals if there is an SU in the network. In this paper, we analyze a case where each PUE attack is individual, and there are no shared data between them. This makes it difficult to detect it with simple methods.

In this model, each SU has algorithms to detect the energy and authenticate the PUE at each frequency on the range. When it sends data to the FC, the process can start again, looking for a new frequency or detecting probable PUE attacks. Then, the data of each PUE attack are saved in the blacklist that is shared with all the SUs. Next time, if a PU/PUE signal is detected, the data are compared with the blacklist, and if it is a known attacker, the process is fast, and communication can continue forcing the MU to release the channel. The model can be seen in Figure 2.

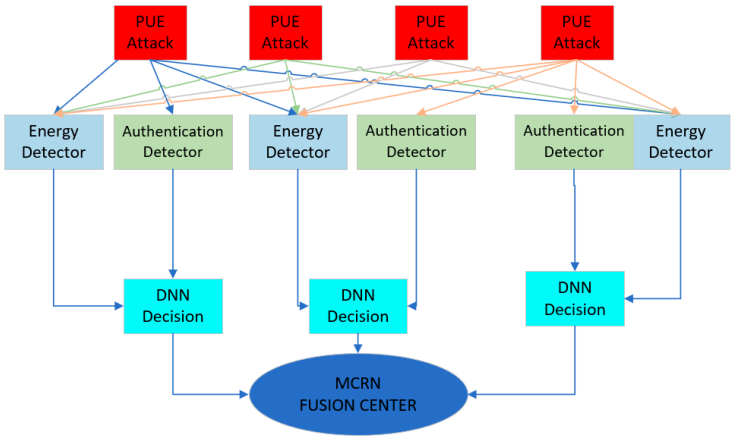

Figure 2. Model for multiple PUE attack detection.

In this solution, each SU must continuously monitor the spectrum to perform energy detection, which involves analyzing signal power levels and defined thresholds. It also needs to communicate its findings to the FC, which can add additional communication overhead. However, in this case, the data transmitted if it is not an attack are just a binary decision. In contrast, if the attacker is identified, its identification data are transmitted once, and some bytes of data are sent.

The computational complexity is medium in the learning process, which needs the DNN to estimate the model parameters, while it is low in the running process because the algorithms are trained and the calculations are lower.

In the following subsections, the detectors will be explained in detail.

3.1. Cooperative Energy Detection

The Centralized Control of FC simplifies decision-making and coordination. Centralized data collection can enhance the accuracy of detection algorithms, such as those using DNN. Nevertheless, some issues need to be addressed. High traffic to the FC can slow down response times and add delays, especially under heavy attack scenarios.

In Decentralized Control, each SU operates independently, maintaining network functionality even if some nodes fail, and does not need a central system. However, ensuring that SUs effectively share detection information can be more complex without a central authority, and maintaining up-to-date knowledge of PUE attacks across a decentralized network may require robust protocols. In this case, a centralized control is implemented to keep a database of the attackers.

The energy detection system aims to identify a threshold that enables an MCRN to differentiate a PU from a PUE. However, as the users move, the energy detector alone cannot determine if there is a PU or a PUE.

In this model, the energy detector identifies a possible PU/PUE signal. It releases the channel and initiates the detection process. If there is a PU, it is permanently released while it is transmitting, but if the system recognizes a PUE, it can continue using the channel.

The energy detector is shown in Equation (1) [15].

$$f(t) = \begin{cases} n(t) & SU \\ i(t) * s(t) + n(t) & PU/PUE \end{cases} \quad (1)$$

where n is the noise, i represents the impulse, and s is the received signal. Then, a binary hypothesis is used as described in Equation (2).

$$F(\mathrm{n}) = \begin{cases} a(\mathrm{n}) & H0 \\ s(\mathrm{n}) + a(\mathrm{n}) & H1 \end{cases} \quad (2)$$

where a is the noise channel, n is the sample index, and s is the signal. In the energy calculation, N is the number of samples of the energy [16].

$$Z(F_n) = \frac{1}{N}\sum_{n=1}^{N} |F(n)|^2 \tag{3}$$

Compared to a threshold λ, the value of Z decides the PU/PUE signal presence. The probabilities of false alarm (P_{fa}) and detection (P_d) for a CSS are given by [17]:

$$\begin{aligned} P_d &= P\{T(X) > \lambda / H1\} \\ Q_d &= 1 - \prod_{i=1}^{K} (1 - P_{d,i}) \end{aligned} \tag{4}$$

$$\begin{aligned} P_{fa} &= P\{T(X) > \lambda / H0\} \\ Q_{fa} &= 1 - \prod_{i=1}^{K} (1 - P_{fa,i}) \end{aligned} \tag{5}$$

The energy detected is then sent to the FC, and the frequency is released while the whole process is carried out.

3.2. Authentication Detection

In this process, any PUE in an MCRN works as a base station, meaning an MU can authenticate on it, but an SU cannot. A PUE attack occurs when a PUE acts like an actual primary base station (PBS), copying the data from it. An SU in the MCRN, when a signal from a PU/PUE is detected, releases the channel and tries to connect to the possible PU/PUE. As a result, the PUE attack cannot authenticate the user, confirming that there is a PUE on the network. All SU authentication results and data captured in this process are transmitted to the FC.

The additional data shared by any PBS or PUE are mobile country code (MCC), mobile network code (MNC), short name, location area code (LAC), and cell identification (CID). These data are recorded in the blacklist to quickly identify the attacker the next time it attempts to gain access to the MCRN frequencies.

Even though there is no direct connection with the operator's databases, this authentication process detects the data from the PU and PUE attackers and can record these as a reference.

For example, the short ID is shared with all the users in a mobile network. It can be seen in Figure 3 on a phone, but it can also be an SU.

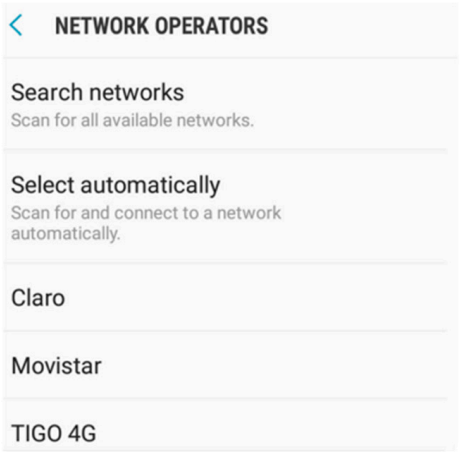

Figure 3. Example of global information shared by a base station.

3.3. DNN for Detection

DNNs have become a powerful tool for addressing complex pattern recognition tasks in various domains. In the realm of CRN, DNNs offer promising solutions for enhancing spectrum-sensing capabilities, particularly in the detection and classification of PUE. These consist of neurons connected in several layers; the neuron is connected to another in the subsequent layer. The core components of a DNN architecture include the following:

Input layer: This layer accepts the input feature vectors derived from spectrum-sensing data. Features may include signal strength, frequency patterns, and temporal information. The input layers for the model are energy and the authentication detection results.

Hidden layers: Each layer comprises dense layers, including neurons activated by functions such as Rectified Linear Units (ReLUs). A ReLU is commonly used to train models and efficiently decrease the vanishing gradient. Some neurons are deactivated during the training process to minimize overfitting. It is defined as follows:

$$\mathrm{ReLU}(x) = \max(x) \tag{6}$$

This means that a ReLU allows only positive values to pass through, effectively "rectifying" the input. Although it looks linear for positive inputs, the function introduces non-linearity into the model, enabling it to learn complex patterns [18].

Output layer: it uses a softmax activation function for classification, allowing the system to separate the input data into categories for different purposes, like the user type [19].

Softmax is a mathematical function commonly used in machine learning, particularly in classification problems involving multiple classes. It converts a vector of raw scores (logits) into probabilities that sum to 1, making it suitable for interpreting a model's output as probabilities for each class. It is often used in the final layer of a neural network model for classification tasks. It converts raw output scores into probabilities by taking the exponential of each output and normalizing these values by dividing by the sum of all the exponentials. This process ensures the output values are in the range (0, 1) and sum up to 1, making them interpretable as probabilities [18].

The input to the softmax function is a vector of K elements, where z without an arrow represents an element of the vector:

$$\vec{Z} = Z_0, Z_1 \ldots, Z_K \tag{7}$$

Formally, the standard (unit) softmax function is as follows:

$$\mathrm{softmax}(xi) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{8}$$

where x defines an output.

These elements are shown in Figure 4 [20].

As with other neural networks, two main steps are required: practical training with extensive and diverse datasets. The data should represent the problem domain and include various scenarios to ensure robustness; the second step is validating the network by testing it.

In the first part, data preprocessing involves normalization, augmentation, and splitting into training, validation, and test sets. It starts by randomly initializing each neuron's weight matrices and bias vectors; it is essential to initialize the parameters to values between 0 and 1.

In forward propagation within a neural network, let denote the label of a particular layer. The term Z^L is the pre-activation vector for layer L and represents the weighted sum

of the input values X plus the bias terms b associated with each neuron, and w is the weight matrix that quantifies the influence of each input on this summation [21].

$$Z^L = w^L X^{L-1} + b^L \tag{9}$$

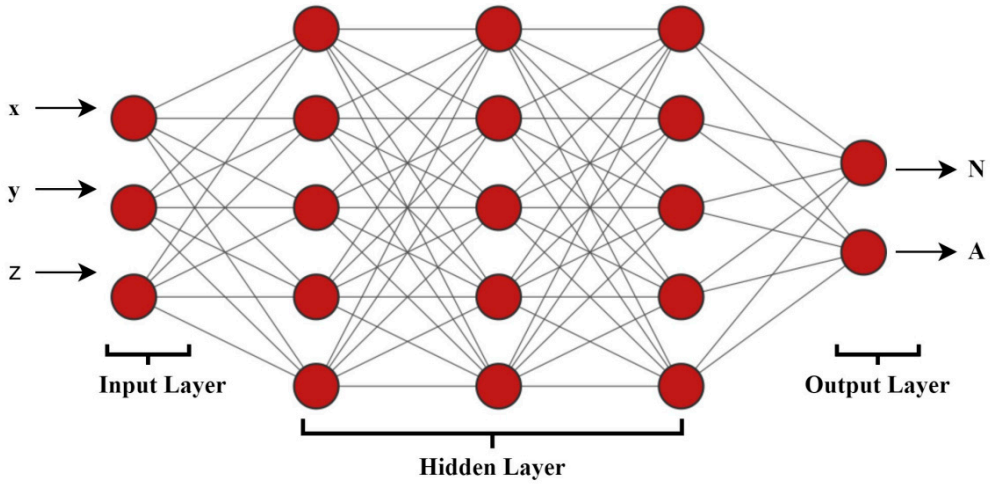

Figure 4. Deep artificial neural network [20].

Function a is the activation input to Z to obtain the response X^L, which is the activation vector after applying the activation function σ to Z^L. Common activation functions include the ReLU and the sigmoid function.

$$X^L = \sigma(Z^L) \tag{10}$$

The error functions show how close the predicted outputs are to the desired outputs by calculating the cost function. This function can be used for regression problems or classification [22]. The error is calculated as in Equation (8).

$$Error = X - X^L = C(a^L(Z^L)) \tag{11}$$

The error function quantifies how well the model's predictions match the target values. The goal during training is to minimize this error, allowing the model to learn and improve its predictions.

For this model, the cross-entropy loss is used.

$$Loss = -\sum_{i=1}^{K} y_i * \log(\hat{y}_i) \tag{12}$$

Here, y_i is the true value, $\hat{y}_i$ is the predicted value, and K is the number of classes.

Backpropagation enables the network to adjust its parameters to minimize the loss function. This optimization process involves calculating gradients and updating weights and biases accordingly. It aims to develop a solution that enhances the network's generalization ability, thereby improving prediction accuracy on new data. This is reflected in the error observed in the final layer [23]. Backpropagation computes gradients of the loss function concerning each weight and bias using the chain rule.

$$\delta^L = \frac{\partial C}{\partial a^L} \frac{\partial a^L}{\partial z^L} \tag{13}$$

The backpropagation is implemented in Equation (10).

$$\delta^{L-1} = W^L \delta^L \frac{\partial a^{L-1}}{\partial z^{L-1}} \tag{14}$$

And the derivatives are shown in Equation (11).

$$\frac{\partial C}{\partial b^{L-1}} = \delta^{L-1} \tag{15}$$

$$\frac{\partial C}{\partial w^{L-1}} = \delta^{L-1} a^{L-2} \tag{16}$$

The weights WL and biases b gradients are computed by backpropagating the errors from the output layer through the network to the preceding layers. This process involves calculating the gradients and then updating the weights and biases using optimization techniques such as gradient descent. These updates are designed to minimize the loss function iteratively. Following these adjustments, the network's performance is evaluated using test data to assess its effectiveness. If the performance does not meet expectations, hyperparameters are modified, reiterating the process [23].

DNNs provide robust solutions for spectrum sensing in CRNs, with careful attention to data preprocessing, network training, and iterative optimization required to achieve optimal performance.

4. Experiments

An MCRN testbed environment is implemented on SDR devices to evaluate the detection technique for mitigating multiple PUE attacks. The experiment uses the following hardware and software components:

Hardware:

SDR Platforms: USRP N210, NI USRP 2922, and RTL2832 for signal transmission and reception of a PU, SU, and PUE attacker.

Computing resources: a Linux-based workstation with an Intel Core i5 multi-core processor and 16 GB of RAM was used with a Gigabit ethernet connection for signal processing and analysis.

Software:

GNU Radio 3.10.11.0 (https://www.gnuradio.org/): used for creating the MCRN environment, the attacks, and the detection system [24].

MATLAB/Simulink 9.14 (https://www.mathworks.com/products/matlab.html): utilized for additional signal analysis and visualization.

Custom scripts: Python and GNU Radio Companion scripts were developed to automate the testing and data collection processes.

4.1. Experimental Environment

The experiment is carried out in a controlled indoor environment to minimize external interference. The setup included:

- PU: a mobile phone.
- SU: Implemented using SDR platforms to receive and analyze signals. Multiple SUs were used to simulate an MCRN environment.
- PUE attacker: a dedicated SDR platform was configured to perform multiple PUE attacks by sending false PU signals.

The experiment focused on evaluating the detection technique under various attack scenarios:

- Single PUE attack: one attacker simulates a single PU signal.
- Multiple PUE attacks: multiple attackers (four) simultaneously simulate PU signals at different frequencies and strengths.

4.2. Detection Technique

Energy Detection

Energy detection is the initial method for detecting PUE attacks and releasing the channel for the possible PU/PUE. The following steps outline the energy detection process:

Signal collection: SDR receivers collect signals over a specified bandwidth.

Energy measurement: the received signal's energy is measured using the energy detection algorithm, which involves computing the signal's power in a given time window.

Threshold comparison: the measured energy is compared against a threshold to determine if a PU signal is present or an attack is occurring.

A PU is moved in the environment to define the threshold, and the energy data are saved for the DNN's learning process.

All users are in a grid of 10 × 10 m. The FC is located in the center of the grid, and the 3 SUs are at the corners of the downside and in the center. The PU is randomly distributed in the environment, as are the PUE attacks.

For the DNN's learning process, a PU is moved for all the parts of the grid, and the energy is calculated. These data calculate the threshold to recognize a device transmitting at some frequency in UL and DL. Figure 5 shows an example of the user's distribution on the grid.

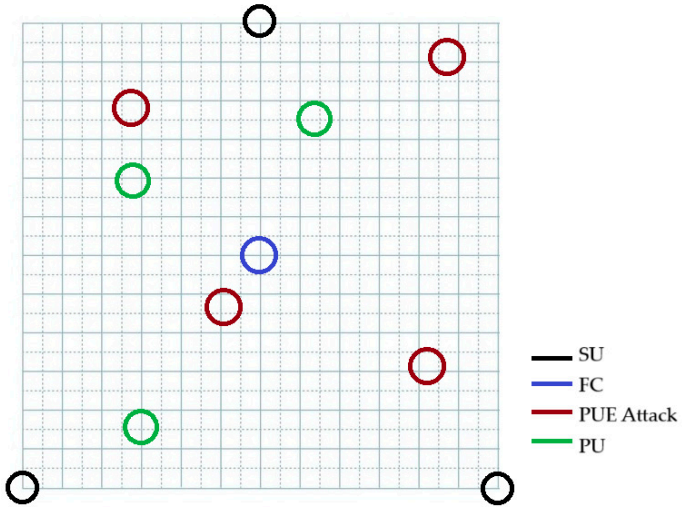

Figure 5. Example of the user's position in the environment (source: own).

Authentication Detection

In the learning process, for authentication detection, the SU attempts to connect to the PU's base stations and obtain accurate data on the stations in the zone. A database with these data is generated and shared with the FC and all the SUs in the zone.

Then, in the detection, when a possible PU is detected for some of the SUs, the SU tries to connect with it. If it is a PU, the data are compared with the database, saved, and shared with the FC. If the SU authentication process is invalid, it is marked as a PUE and shared with the FC.

After the detections are achieved, the DNN algorithm takes data from all the SUs. Based on the energy and authentication results, the PUE attack presence in the network is estimated.

The whole process is achieved when one PUE attack and four PUE attackers are in the network, and the results are compared.

5. Results

In the learning process, 100 samples are taken at each grid point for the energy results. A PU device is moved around the grid, and signals are taken in a 10 × 10 grid, obtaining 10,000 samples for the threshold calculation in all the coverage. The parameters of the experiments are shown in Table 1.

Table 1. Parameters for energy detection learning.

Parameter	Value
Number of Samples	10,000 samples
Averaged Values	100 samples (10 × 10)
Noise Signal	AWGN
Service	Phone Call-PUEA
Frequency	876.8 MHz
Confidence Level	95%
Margin of Error	5%
Users	3 SUs, 1 FC, 1/4 PUE

5.1. Energy Results

For example, Figure 6 shows the energy calculation for the DL signal in 876.8 MHz, one of the frequencies in the selected range (876–880 MHz). The energy values are averaged, and if they are above the threshold, there is a signal of a PU/PUE, and the data are sent to the FC.

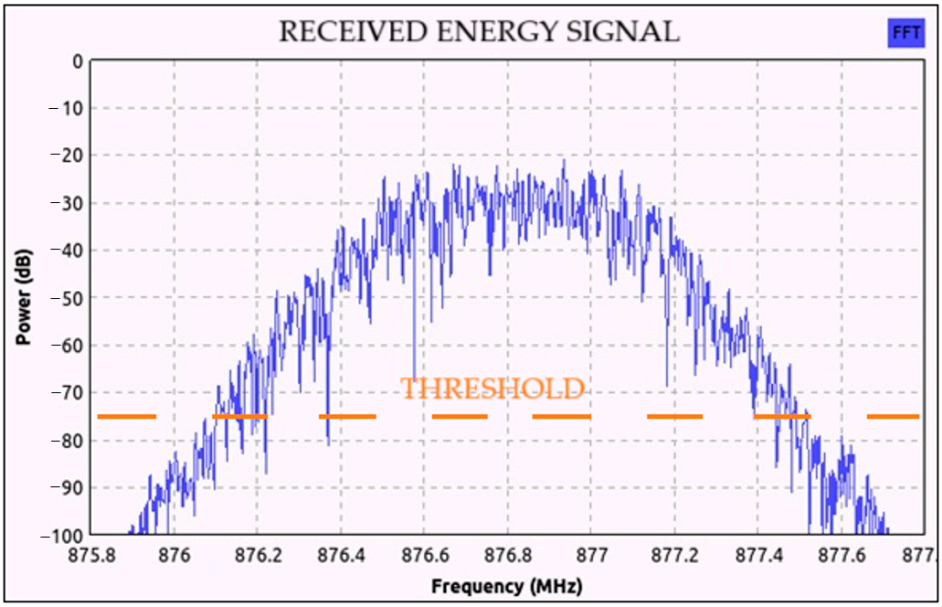

Figure 6. Example of energy detection (source: own).

This process is achieved for the DL and UL frequencies, and the results are sent to the DNN algorithm. In the learning process, the algorithm obtains the threshold. After this, in the detection process, when the system detects a suspicious signal, energy is measured in DL and UL and sent to the DNN algorithm.

The SU achieves a site survey for authentication detection, which puts the operators' information in the coverage zone. THE MCC, MCN, and short name are saved and shared in the base station database. When the detection system starts, the SU tries to connect to the suspicious device and compare it with the database. The device is identified as a PU if the authentication process is successful. Nevertheless, it is a PUE if the information does not correspond to the database or the authentication process fails. After the authentication detection, the data are saved on the blacklist and shared with the FC. The parameters for the authentication process are shown in Table 2.

The detection results are calculated and sent to the DNN system so that the final decision on each SU can be made. For the learning process, 1000 authentication samples were made in different grid positions. In the detection process, each SU makes one authentication process. The detection is valid for 923 samples, obtaining a PD of 92%.

Table 2. Parameters for authentication.

Parameter	Value
Number of Samples	1000
Noise Signal	AWGN
Service	Phone Call-PUEA
Frequency	876.8 MHz
Confidence Level	95%
Margin of Error	5%
Authentication Time	20 s
Users	3 SUs, 1 FC, 1/4 PUE

The USRP is connected to a PC with Ubuntu and OpenBTS. The system's base is GNURadio, and the algorithms are programmed in Python. The base system can be seen in Figure 7.

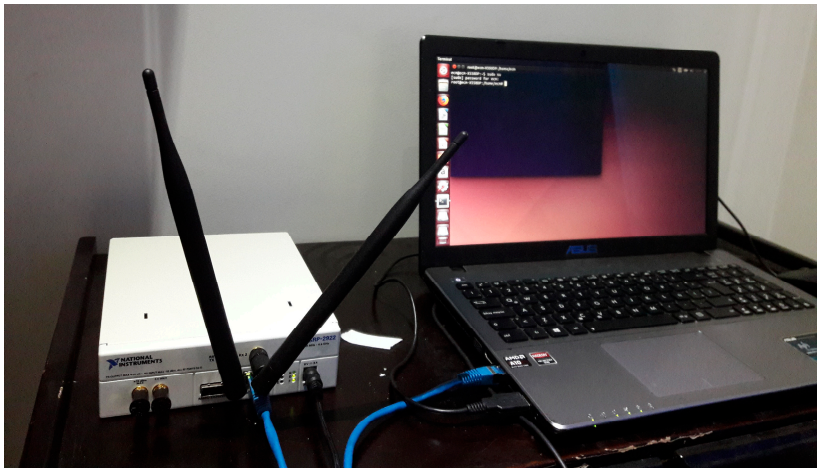

Figure 7. SDR test bed platform.

For moving the SDR, we implement a system that moves just the antenna with the help of a quadcopter, which does not affect the users' received signals, as seen in Figure 8.

For moving the SDR, we implement a system that moves only the antenna and not the SDR with the help of a quadcopter, which does not affect the users' received signals, as seen in Figure 8.

Reliance on energy detection and local databases for authenticating SUs can lead to false positives. Obstacles and fading can affect the detected signal strength of PUs. In the model, energy detection detects PUs or SUs and releases the channel, protecting the PUs. In contrast, the detection is complemented by an authentication process to distinguish legitimate from malicious SUs, obtaining detailed user information. The PU will always connect to the PBS and will not have connection problems; the SU can be affected temporarily while the PUE attack is confirmed or denied.

Creating and sharing a database with attacker data can enhance the network's ability to recognize patterns associated with PUE attacks. This could be particularly useful in distinguishing between known threats and legitimate behavior.

Figure 9 shows the receiver operating characteristics (ROC) results for different SNR values with the parameters of Table 1. These curves serve as a parameter to study the performance of the sensing scheme.

We measure the power signal in each point of the grid in case of the presence and absence of the PU or PUE. According to [25], the IEEE 802.22 standard recommends PFA <0.1 for spectrum sensing. Analyzing the simulation results, we expect an 88% detection for this PFA. The PU power level was measured to find an optimal threshold for the experiment,

and 10,000 samples were taken in the lowest part of the grid close to MCRN. With these samples, the confidence level is 95%, and the margin of error is 5%.

Figure 8. Mobile SDR device.

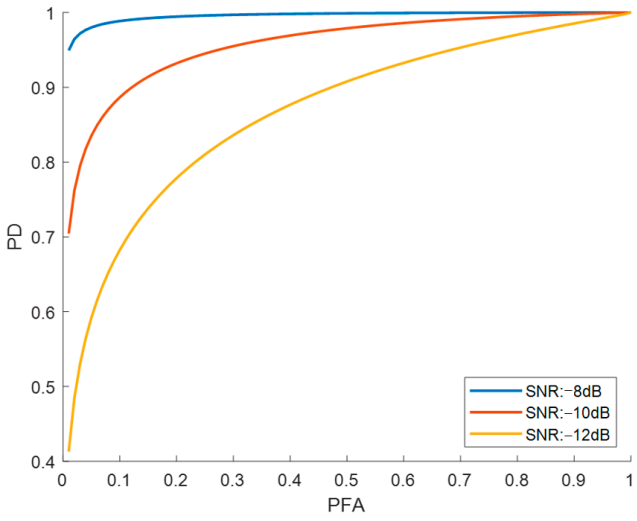

Figure 9. Probability of detection vs. probability of false alarm results for AWGN channel (source: own).

This study delineates the hypothesis-testing framework for downlink and uplink signals. Downlink communication aims to distinguish the desired signal from noise by applying a defined threshold. Conversely, in uplink scenarios, the absence of a signal indi-

cates an unauthorized user. In contrast, the presence of a signal is attributed to a legitimate primary user, contingent upon prior validation of a downlink signal.

Specifically, in the downlink context, a signal may be detected by either a legitimate primary user or a user attempting to mimic primary user behavior. In uplink communication, a detected signal unequivocally indicates a primary user, whereas its absence signifies the presence of an emulating user. Consequently, the overall probability of detecting a malicious user is determined by the product of the detection probabilities associated with both downlink and uplink transmissions. The Figure 10 shows the results of Montecarlo simulations in a CSS scheme for 1, 3, and 10 SUs with an SNR = −12 dB expressed as ROC curves.

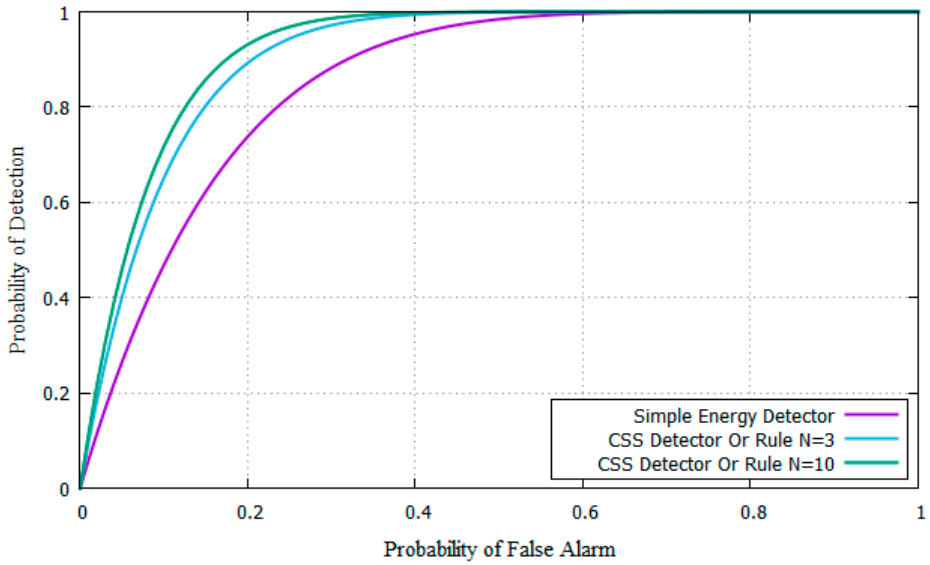

Figure 10. Probability of detection vs. probability of false alarm for CSS for SNR = −10 dB (source: own).

As expected from the literature, the CSS has better results than individual detection. Increasing the number of SUs transmitting the detector results to the FC also increases the PD for the assigned PFA = 0.01.

The results of the downlink power measurement can be seen in Figure 11, where the spectrum measurement without power and with power is shown.

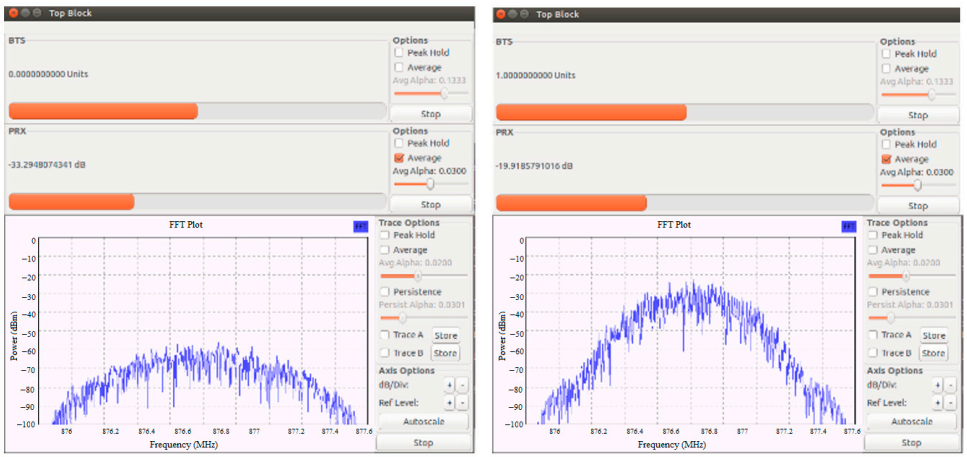

Figure 11. Downlink signal without and with active signal (source: own).

The results of the uplink power measurement can be seen in Figure 12, where the spectrum measurement without power and with power in the uplink channel is shown.

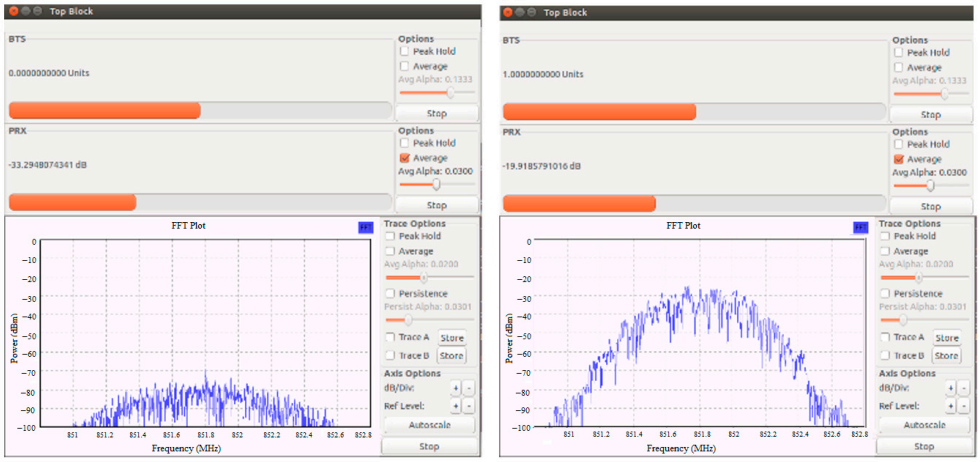

Figure 12. Uplink signal without and with active signal (source: own).

The results show that by selecting a practical threshold for downlink at −50 dBm, the percentage of malicious PUE detection with fixed location is 92% (9188/10,000) with one PUE. When there are four PUEs, the detection is (8720/10,000).

5.2. Application Data Results

We identified data from telecommunication operators in Colombia by analyzing site survey measurements and experimental configuration settings about PUs and PUEs. The relevant information includes mobile country code (MCC) 732 and mobile network codes (MNCs): 101 for Claro, 102 for Movistar, 111 for Tigo, 142 for UNE, and 154 for Virgin.

Subsequently, the application transmits key information—such as the network's short name, MCC, and MNC—to the central base station (CBS). The CBS then cross-references these data with verified operator information to determine whether the signal is associated with a legitimate primary user or an emulator. Additionally, the SUs undergo authentication within the network. The results are also transmitted to the FC for final decision-making.

In scenarios where the system encounters a short name associated with an alternative operator, the sensors will attempt to connect and gather information concurrently with the actions executed by the motion detection algorithm. During the experiments, the default identifier for the primary user emulator (PUEA) is set to "01-001" or referred to as the "range network," depending on the Software Defined Radio (SDR) in use. When an emulator attempts to mimic a legitimate primary user, it may present itself similarly to a genuine operator (e.g., Operator_).

The testing process starts with a PUE designed to simulate a PU. However, if the system detects an invalid short name or authentication fails, the device is subsequently classified as a PUE, as illustrated in Figure 13.

If the PUE mimics the operator's real short name, the algorithm detects a duplicate name and tries to authenticate both to correctly identify the PU and the PUE. The results are saved as binary (0—authentication OK; 1—no authentication).

Each SU performs one authentication process in the detection process. The detection is valid for (905/10,000), obtaining a PD of 90% for the authentication process for one PUE. For four PUEs, there is a PD of (862/1000), 86%.

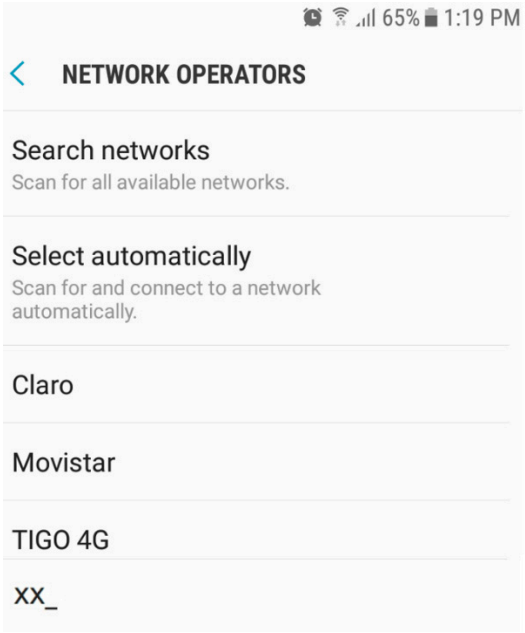

Figure 13. Available networks with PUE screen in the mobile phone (source: own).

5.3. DNN Results

The DNN system takes the data from the two detection processes, energy, and authentication; some libraries, such as Sci-kit and Keras, were used. The results are obtained when a PUE and four simultaneous PUE attacks occur. The SNR is estimated with each measure as additional data for the results. With this experimental dataset, the DNN is calculated, and on each SU, it takes the final decision of the PUE attack detection and is sent to the FC for final decision.

The DNN network architecture has the parameters defined in Table 3.

Table 3. DNN parameters for PUE detection.

Parameter	Value
Hidden Layers	2
Neurons	32
Activation Function	ReLU and Softmax
Epochs	50
Bath Size	32
Total Parameters	642

SNR levels range from −25 dB to 0 dB. The algorithm achieves a detection probability and accuracy of 90% at −9 dB SNR. The confusion matrix of the trained algorithm uses 10,000 samples, as seen in Figure 14.

We experiment with some epoch values to obtain precision and optimize the model, as seen in Figure 15.

According to this experiment, 50 epochs are required to obtain good precision results, but with more, it shows higher precision results but involves more calculations.

The number of neurons was initially 16, but we increased it to 32, keeping good precision results and balancing the computational complexity of the model. Using 64 neurons increases the precision results of the model by 2%.

The prediction probability increases using the DNN; individual energy and authentication models maintain an 86%, while more than 95% are obtained with the DNN model.

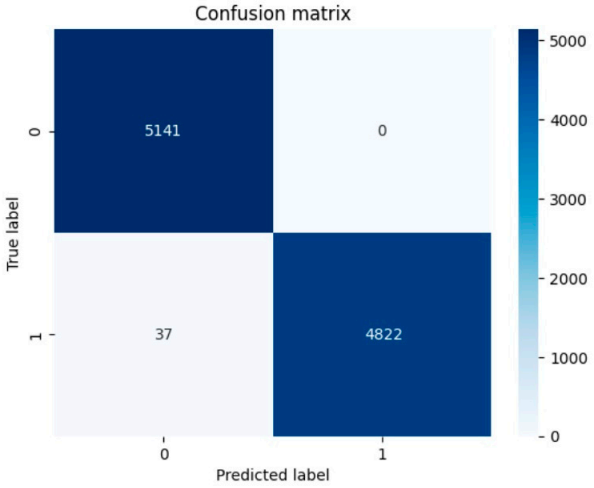

Figure 14. Confusion matrix -10 dB (source: author).

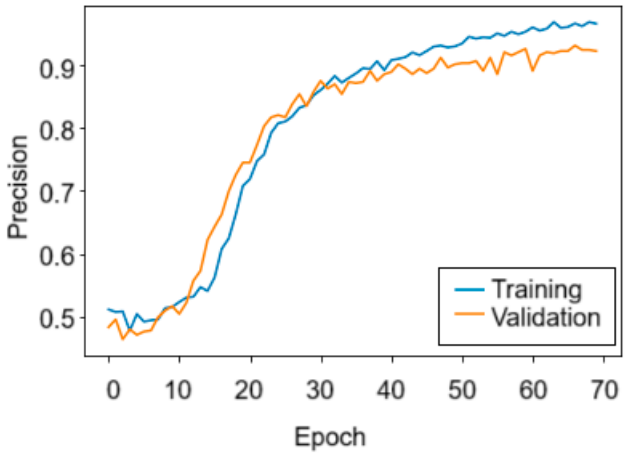

Figure 15. DNN results depend on the epoch size (source: own).

The DNN code is implemented with Keras in Python; part of the code can be seen in Figure 16.

```
model = Sequential()
model.add(Dense(32, input_dim=1, activation='relu'))
model.add(Dense(2, activation='softmax'))
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_test))
loss, accuracy = model.evaluate(X_test, y_test)
print('Loss: {:.4f}, Accuracy: {:.4f}'.format(loss, accuracy))
```

Figure 16. DNN code in Keras and Python (source: own).

An OR logic mechanism estimates the PUE attack in the FC; if any SU identifies a PUE, an alert is broadcast from the FC to alert all SUs within the network.

The probability of detecting a PUE attack is calculated as the number of correct PUE detections to the total instances in which a PUE is present. Experiments are carried out with one and four PUE attacks active on an SDR simultaneously at different work frequency ranges. The samples are taken from each dB in increments of −25 dB to 0 dB, with the results illustrated in Figure 17. The receiver operation curves (ROC) for the probability of detection for a CSS energy detector based on [26] are compared with the DNN experimental results for a single and four PUE attacks.

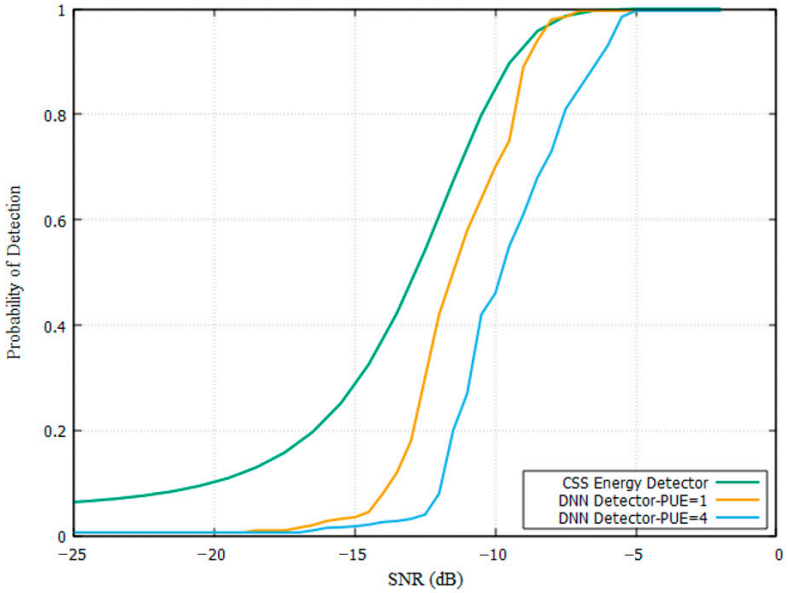

Figure 17. Probability of detection of a PUE attack (source: own).

The results show that the SDR experiment achieves a probability of detection above 90% in -9 dB when there is a single PUE attack and −7 dB with multiple PUE attacks. The PFA for DNN results is 5% for a single PUE attack and 10% for a multiple PUE attack. The results are compared with those of existing work with techniques like KNN and SVM [27] for −7 dB of SNR. Table 4 shows the probability of detection for one or multiple PUE attacks and the measured false alarm rate.

Table 4. DNN PUE detection comparison with other techniques.

Technique	Detection Probability One PUE	Detection Probability Four PUEs	False Alarm Rate
DNN	95%	90%	10%—Multiple PUEs
SVM	98%	0%	5%—Single PUE
KNN	97%	0%	5%—Single PUE

The SVM and KNN show the highest detection values, but they work only with a single PUE attack; other detectors for multiple PUEs do not work correctly, but the DNN works with 90% of detection. The computational complexity is similar in the three cases; it is medium [28]. In the DNN model, the measured false positive rate is close to 10% (1042/10,000 samples), while the false negative is close to 5% (514/10,000 samples).

The Table 5 describes the techniques' results individually and the DNN's.

Table 5. Energy, authentication, and DNN PUE detection comparison.

Technique	Detection Probability One PUE	Detection Probability Four PUE	False Alarm Rate
Energy	(9188/10,000) 92%	(8720/10,000) 87%	5%—Single PUE
Authentication	(905/1000) 90%	(862/1000) 86%	5%—Single PUE
DNN	95%	90%	10%—Multiple PUEs

The individual energy detection shows a PD of 92% for one PUE and 87% for four PUE attacks. The authentication process obtains a PD of 90% with one PUE and 86% for multiple (four) PUE attacks. The DNN model uses the detection results of the two techniques. After

the learning process, the PD results increased to 95% with one PUE and 90% with the four PUE attacks, showing an increment of 5% for authentication and 3% for energy with one PUE attack and 4% in authentication for multiple PUEs.

The false alarm rate is 13% for energy detection and 14% for authentication detection with multiple PUEs. For DNN, it decreases to 10% with multiple PUE attacks.

The DNN creates a new interaction between the input variables, allowing for higher detection results with multiple PUE attacks while keeping lower values for false alarm results. The time for calculation after the learning process is lower than that of the other techniques, allowing for an average detection time of 5 s, faster than the 20 s that the authentication process needs in a cooperative environment.

6. Discussion

The CSS single energy detector working with an authenticator detector can detect a PUE attack with a probability of detection above 90% for an SNR of -10 dB in simulations. In the SDR experiments with the DNN, this value is achieved in -9 dB, very close to simulations, and works with a mobile PUEA and PU. In the presence of four PUE attackers in the environment, the SU starts releasing frequencies and detecting the possible PUE, obtaining a 90% detection in -7 dB. The effects of multiple PUEs are an increasing value of SNR, and the time detection increases by 2.36 s, the average time the DNN algorithm takes to detect the attack. The DNN algorithm works faster with the energy detector to release the channel and detect the malicious PUE attackers; authentication detection takes an average time of 20 s, obtains 90% of detection, and is also used to save the data from the attacker and update the database from PU and PUE attackers. The DNN algorithm detects multiple PUE attackers even if they are simultaneous in 5 s, but it releases the channel in 200 ms.

The cognitive protocol implemented in this project is a CSS with an FC that keeps all the network data; it has been probed with users in movement in the MCRN in a controlled environment. To adapt it to a real-world scenario, continuous monitoring of the spectrum environment and user mobility is needed to adjust parameters like frequency allocation and power control dynamically. Given the potential for various interference sources, the protocol must be designed to maintain robustness. Adaptive modulation, error correction, and interference mitigation strategies can help sustain communication quality under diverse conditions.

As part of future work, the protocol should be tested in scenarios that simulate large-scale deployments with increasing user density. This includes implementing resource management strategies that can efficiently handle the demands of multiple users while minimizing conflicts and optimizing spectrum usage. Implementing cooperative strategies among SUs can enhance detection and response to PUE attacks. A distributed learning approach, where devices share information about detected attacks, can improve system awareness and responsiveness.

Integrating deep neural networks (DNNs) can help detect PUE attacks and predict network conditions. Establishing feedback loops where SUs can report their experiences with interference and PUE attacks will create a learning environment that continuously improves the protocol's performance. This approach enhances the protocol's robustness and adaptability and supports the collaborative nature of Mobile Cognitive Radio Networks (MCRNs).

As the SUs increase, the FC may struggle to process incoming data efficiently. In future work, a hybrid solution that combines centralized and decentralized elements will be explored to leverage the strengths of both architectures.

Using the DNN improves the prediction rates, decreases the false alarm rates, and speeds up the detection process in the presence of multiple or single PUE attacks. It can process several amounts of data in the learning process and extract the features or patterns of the inputs, simplifying the modeling process and adapting to one or multiple PUE attacks.

7. Conclusions

Using the energy detection and authentication processes provides a framework for distinguishing between legitimate PUs and malicious PUE attacks. The energy detection mechanism identifies potential PU/PUE signals by comparing the received signal's energy against a predefined threshold. This is followed by an authentication process that helps to verify whether the detected signal is from a legitimate PU or an impersonating PUE. The combined use of these methods allows for a high probability of accurate PUE detection, as demonstrated by experimental results.

The implemented DNN for PUE detection increases the system's ability to classify and detect multiple PUE attacks. The DNN was trained with features derived from both energy detection and authentication results. Experimental results showed that the DNN achieved a detection accuracy of 90% at an SNR of -9 dB. The DNN's capability to process and analyze large datasets improves the system's adaptability to multiple and moving network users and attackers.

The proposed methodology was tested in a controlled SDR-based environment, validating its performance in real-world settings. The experiments involved single and multiple PUE attacks, with the DNN achieving a detection probability of over 90%, even in challenging conditions. The results demonstrated that the detection system can effectively identify and manage multiple simultaneous PUE attacks, maintaining high performance across different SNR conditions.

Author Contributions: The design, experiments, and environment proposed in this paper were conceived by E.C.M.; SDR implementation and test, E.C.M., G.C.P. and A.A.M.; results' analysis, discussion, and editing of this paper, E.C.M., G.C.P. and A.A.M. All authors participated in proofreading, and read and approved the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Universidad Santo Tomas, Proyecto FODEIN 2023.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Acknowledgments: We thank Universidad Santo Tomas for funding this project.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vishnu, J.B.; Bhagyaveni, M. Opportunistic transmission using hybrid sensing for Cognitive Radio Sensor Network in the presence of smart Primary User Emulation Attack. *Int. J. Electron.* **2020**, *108*, 1183–1197. [CrossRef]
2. Sekar, S.; Jeyalakshmi, S.; Ravikumar, S.; Kavitha, D. Modified light GBM based classification of malicious users in cooperative cognitive radio networks. *Cyber-Phys. Syst.* **2022**, *10*, 104–122. [CrossRef]
3. Olaleru, G.; Ohize, H.; Mohammed, A.S.; Dauda, U.S. Optimal Detection Technique for Primary User Emulator in Cognitive Radio Network. In Proceedings of the 2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS), Abuja, Nigeria, 15–16 July 2021; IEEE: Piscataway Township, NJ, USA; pp. 1–6.
4. Shrivastava, S.; Chen, B.; Wang, H. DQN Learning Based Defense Against Smart Primary User Emulation Attacks in Cooperative Sensing Systems. *IEEE Access* **2021**, *9*, 163791–163814. [CrossRef]
5. Awan, M.N.; Haq, S.U.; Anwar, S. An Improved Cooperative Spectrum Sensing Scheme for Emulation Attacker Detection in Cognitive Radio Network. In Proceedings of the 2023 International Conference on Robotics and Automation in Industry (ICRAI), Peshawar, Pakistan, 3–5 March 2023; IEEE: Piscataway Township, NJ, USA; pp. 1–5.
6. Aygul, M.A.; Furqan, H.M.; Nazzal, M.; Arslan, H. Deep Learning-Assisted Detection of PUE and Jamming Attacks in Cognitive Radio Systems. In Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Virtual, 18 November–16 December 2020; IEEE: Piscataway Township, NJ, USA; pp. 1–5.
7. Chatterjee, P.S. A Systematic Survey for Detecting and Counteracting PUE Attacks in CWSNs. In Proceedings of the 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 1–3 October 2021; IEEE: Piscataway Township, NJ, USA; pp. 1–6.
8. Inamdar, M.A.; Kumaraswamy, H.V. Accurate Primary User Emulation Attack (PUEA) Detection in Cognitive Radio Network using KNN and ANN Classifier. In Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 15–17 June 2020; IEEE: Piscataway Township, NJ, USA; pp. 490–495.

9. Xu, M.; Zhao, Y.; Zhang, R.; Yin, Z.; Wu, Z. A Reliable Spectrum Sensing Method Based on Deep Learning for Primary User Emulation Attack Detection in Cognitive Radio Network. *IEEE Commun. Lett.* **2024**, *28*, 547–551. [CrossRef]
10. Camana, M.R.; Garcia, C.E.; Koo, I.; Shakhov, V. Machine Learning Based Primary User Emulation Attack Detection. In Proceedings of the 2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Sofia, Bulgaria, 6–9 June 2022; IEEE: Piscataway Township, NJ, USA; pp. 244–248.
11. Wang, Q.; Sun, H.; Hu, R.Q.; Bhuyan, A. When Machine Learning Meets Spectrum Sharing Security: Methodologies and Challenges. *IEEE Open J. Commun. Soc.* **2022**, *3*, 176–208. [CrossRef]
12. Elghamrawy, S.M. Security in Cognitive Radio Network: Defense against Primary User Emulation attacks using Genetic Artificial Bee Colony (GABC) algorithm. *Futur. Gener. Comput. Syst.* **2018**, *109*, 479–487. [CrossRef]
13. Reddy, A.A.; Battula, R.B.; Gopalani, D.; Sharma, A. Location based detection mechanism for PUEA on CR enabled 5G-IoT network. In Proceedings of the 2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Hyderabad, India, 13–16 December 2021; IEEE: Piscataway Township, NJ, USA; pp. 331–336.
14. Muñoz, E.C.; Pedraza, G.C.; Cubillos-Sánchez, R.; Aponte-Moreno, A.; Buitrago, M.E. PUE Attack Detection by Using DNN and Entropy in Cooperative Mobile Cognitive Radio Networks. *Futur. Internet* **2023**, *15*, 202. [CrossRef]
15. Jin, F.; Varadharajan, V.; Tupakula, U. Improved detection of primary user emulation attacks in cognitive radio networks. In Proceedings of the 2015 International Telecommunication Networks and Applications Conference (ITNAC), Sydney, Australia, 18–20 November 2015; IEEE: Piscataway Township, NJ, USA; pp. 274–279.
16. Yousef, E.M.; Soliman, H.Y.; Ghuniem, A.M. Sensing-Throughput tradeoff with primary user traffic and cooperative sensing in cognitive radio. In Proceedings of the 2017 2nd International Conference on Computer and Communication Systems (ICCCS), Krakow, Poland, 11–14 July 2017; IEEE: Piscataway Township, NJ, USA; pp. 121–127.
17. Chen, C.; Chen, Y.; Qian, J.; Xu, J. Triple-Threshold Cooperative Spectrum Sensing Algorithm Based on Energy Detection. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018; IEEE: Piscataway Township, NJ, USA; pp. 791–795.
18. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
19. Goodfellow, I. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
20. Doherey, A.; Singh, A.; Kumar, A. Intrusion Detection Using Dense Neural Network in Network System. In Proceedings of the 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), Malang, Indonesia, 16–18 June 2022; IEEE: Piscataway Township, NJ, USA; pp. 484–488.
21. Dufera, T.T. Deep neural network for system of ordinary differential equations: Vectorized algorithm and simulation. *Mach. Learn. Appl.* **2021**, *5*, 100058. [CrossRef]
22. Bengio, Y.; Lecun, Y.; Hinton, G. Deep learning for AI. *Commun. ACM* **2021**, *64*, 58–65. [CrossRef]
23. Higham, C.F.; Higham, D.J. Deep Learning: An Introduction for Applied Mathematicians. *SIAM Rev.* **2019**, *61*, 860–891. [CrossRef]
24. Ettus, C. Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux. 2019. [En línea]. Available online: https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux (accessed on 5 September 2024).
25. Pandya, P.; Durvesh, A.; Parekh, N. Energy detection based spectrum sensing for cognitive radio network. In Proceedings of the 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 4–6 April 2015; IEEE: Piscataway Township, NJ, USA; pp. 201–206.
26. Munoz, C.; Ernesto; Martinez, P.; Fernando, L.; Parra, P.; Patricia, I. Cooperative Energy Spectrum Sensing for Mobile Cognitive Radio Networks using SDR. In Proceedings of the 2020 IEEE Colombian Conference on Communications and Computing (COLCOM), Cali, Colombia, 7–8 August 2020; IEEE: Piscataway Township, NJ, USA; pp. 1–6.
27. Muñoz, E.C.; Pedraza, L.F.; Hernández, C.A. Machine Learning Techniques Based on Primary User Emulation Detection in Mobile Cognitive Radio Networks. *Sensors* **2022**, *22*, 4659. [CrossRef] [PubMed]
28. Liu, X.; Evans, B.G.; Moessner, K. Comparison of reliability, delay and complexity for standalone cognitive radio spectrum sensing schemes. *IET Commun.* **2013**, *7*, 799–807. [CrossRef]