

Article

Not peer-reviewed version

Predicting Noise and User Distances from Spectrum Sensing Signals Using Transformer and Regression Models

[Myke Valadão](#)*, [Diego Amoedo](#), [André Costa](#), [Celso Carvalho](#), [Waldir Sabino](#)

Posted Date: 12 February 2025

doi: 10.20944/preprints202502.0910.v1

Keywords: Spectrum Sensing; Machine Learning; Regression Models; Transformer



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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.

Article

Predicting Noise and User Distances from Spectrum Sensing Signals Using Transformer and Regression Models

Myke Valadão^{1,2,*,†}, Diego Amoedo^{1,3}, André Costa⁴, Celso Carvalho¹
and Waldir Sabino¹

¹ Center for R&D in Electronic and Information Technology (CETELI), Department of Electronics and Computing (DTEC), Federal University of Amazonas (UFAM), Manaus 69067005, Brazil; diegoalves@anatel.gov.br (D.A.); ccarvalho_@ufam.edu.br (C.C.); waldirjr@ufam.edu.br (W.S.)

² Sidi - Institute for Research, Development, and Innovation, Manaus 69037000, Brazil; m.douglas@sidi.org.br

³ National Telecommunications Agency (ANATEL), Manaus 69057070, Brazil

⁴ Electronic and Telecommunications Engineering, Faculty of Electrical Engineering (FEELT), Federal University of Uberlândia (UFU), Uberlândia 38408288, Brazil; alacosta@ufu.br

* Correspondence: mykedouglas@ufam.edu.br

† Current address: Avenida General Rodrigo Otávio Jordão Ramos, 1200, Coroado I, Manaus - AM, 69067-005, Brazil.

Abstract: The frequency spectrum allocation has been a subject of dispute in recent years. Cognitive Radio dynamically allocates users to spectrum holes using various sensing techniques. Noise levels and distances between users can significantly impact the efficiency of cognitive radio systems. Designing robust communication systems requires accurate knowledge of these factors. This paper proposes a method for predicting noise levels and distances based on spectrum sensing signals using regression machine learning models. The proposed methods achieved correlation coefficients of over 0.98 and 0.82 for noise and distance prediction, respectively. Accurately estimating these parameters enables adaptive resource allocation, interference mitigation, and improved spectrum efficiency, ultimately enhancing the performance and reliability of cognitive radio networks.

Keywords: spectrum sensing; machine learning; regression models; transformer

1. Introduction

The frequency spectrum, essential for wireless communications, is a limited resource that has become increasingly congested [1–3]. Cognitive radio (CR) dynamically allocates communication for secondary users (SUs) in parts of the spectrum, known as spectrum holes, where primary users (PUs) are absent [4–6]. Several techniques exist for spectrum sensing, including energy detection, feature detection, Nyquist and sub-Nyquist methods, and multi-bit and one-bit compressive sensing. More recently, machine and deep learning models have been employed to detect the presence of PUs [7,8]. Across these methods, noise can adversely affect the accurate detection of PUs, impacting the efficient use of the spectrum. In multi-user systems, the distance between users significantly influences noise levels. Thus, predicting the noise level in the received signal and understanding user distances is fundamental for optimizing spectrum efficiency.

In communication systems, noise refers to any unwanted or random interference that compromises the quality of a transmitted signal. Noise can distort the original information being transmitted, leading to errors, diminished signal clarity, and reduced communication performance [9]. Almost all communication channels and systems encounter noise, which can originate from various sources. The noise level can be influenced by several factors, including transmitted power, path loss, shadow fading, multipath fading, and the distance between users. In addition to these factors, additive white Gaussian noise is a type of noise that exists across all frequencies [10]. This noise is characterized by its randomness and uniform energy distribution throughout the frequency spectrum. White Gaussian noise is a

common model for representing random background noise in communication systems. By predicting noise levels, one can more effectively optimize system parameters and frequency allocation [11].

The distance between users is an important factor that influences signal quality and, consequently, spectrum efficiency. In spectrum sensing, the strength of the received signal depends on the distance between the transmitter and receiver. Predicting these distances helps adjust transmission power levels: users closer to the transmitter require lower power for reliable communication, while users farther away need higher power [12]. This power control optimizes energy efficiency and minimizes interference. Especially in systems with limited resources, such as bandwidth, predicting distances is fundamental for effective resource allocation. Users closer to the transmitter can be assigned greater bandwidth and consequently higher data rates, while those farther away might be allocated smaller bandwidths to improve signal quality at the same transmission power level. Additionally, distance prediction is essential for providing location-based services. The distance between users can also impact the performance of artificial intelligence models used to increase spectrum efficiency. By estimating distances, services like navigation, location-based advertisements, and emergency services can be offered.

To predict noise and distances between users, regression models and deep learning architectures are recommended [24]. Regression models are a class of machine learning algorithms designed to predict continuous numerical values based on input data. These models play a crucial role in various fields, including economics, finance, healthcare, and natural sciences. They allow for the analysis and forecasting of trends, relationships, and outcomes by learning from historical data. Among the most traditional machine learning regression approaches are support vector regression (SVR) (an extension of support vector machines to regression), decision trees and random forests [13], linear regression (one of the simplest yet widely used regression techniques), and ridge and lasso regression (variants of linear regression). In more robust regression approaches, convolutional neural networks (CNNs), initially designed for image analysis, can be adapted for regression tasks [14]. More recently, the Transformer, designed for forecasting, has shown promise for regression activities [36].

In this article, we propose the use of regression models to predict noise levels and distances based on spectrum sensing signals. During our study, we generated a dataset that considers important parameters, including a wide range of noise power densities, an extensive sensing area, and power leakage from the PU. We compared both traditional and deep learning models for prediction purposes. Furthermore, we evaluated the results using various metrics. Our proposed method has shown promising results, with a correlation coefficient exceeding 0.98 for noise and over 0.82 for distances. Additional metrics also indicate that our method is effective in predicting noise levels and distances between users. Such predictive capability can be important in designing communication systems to enhance spectrum efficiency.

This article is organized as follows: in Section 2, related works about the prediction of noise levels and distances between users in a communication system are presented; in Section 3, the proposed methodology for data generation, training, and evaluation is described; in Section 4, the experiments conducted and the results achieved are presented; the discussions related to the experiments are presented in Section 5; lastly, in Section 6, the conclusions are presented.

1.1. Contributions

The main contributions of this paper can be summarized as follows: (1) the use of the Transformer model, one of the newest and most robust networks, which achieved good results for distance prediction even with a limited architecture; (2) for noise prediction, classical regression methods presented excellent results, achieving a correlation coefficient greater than 0.98, demonstrating a high level of similarity between predictions and real values; (3) since current models used for spectrum sensing are very sensitive to signal quality, the proposed approach can help design better communication systems based on artificial intelligence models by providing prior information.

2. Related Works

In this section, the related works about noise and distances between users prediction in a spectral sensing scheme are described in detail.

2.1. Noise Prediction

The authors proposed a signal-to-noise ratio (SNR) estimation method based on the sounding reference signal and a deep learning network in [15]. The proposed deep learning network, called DINet, combines a denoising convolutional neural network (DnCNN) and an image restoration convolutional neural network (IRCNN) in parallel. The method was compared to other algorithms, and the results demonstrated superior performance in SNR estimation. The evaluation metric used was the normalized mean square error (NMSE) calculated over 200 test samples, yielding a NMSE value of 0.0012. This result was significantly better than the performance achieved by other algorithms.

In [16], the authors presented a method for estimating SNR in LTE systems and 5G. They employed a combination of a CNN and long short-term memory (LSTM), known as a CNN-LSTM neural network. The CNN was utilized to extract spatial features, while the LSTM was used to extract temporal features from the input signal. Data was generated using MATLAB LTE and 5G toolboxes, taking into consideration modulation types, path delays, and Doppler shifts. The evaluation metric used was NMSE. The NMSE achieved a value of zero in the time-domain for SNR ranging from -4 to 32 dB, demonstrating very low latency. However, in the frequency-domain, the proposed method exhibited lower performance.

In [17], the authors proposed NDR-Net, a novel neural network for channel estimation in the presence of unknown noise levels. NDR-Net comprises a noise level estimation subnet, a DnCNN, and a residual learning cascade. The noise level estimation subnet determines the noise interval, followed by the DnCNN, which processes the pure noise image. Subsequently, residual learning is applied to extract the noiseless channel image. The evaluation metric used for assessing the model's performance was the mean square error (MSE). The experiments conducted across different channel models (TDL-A, TDL-B, TDL-C) consistently demonstrated low MSE values. However, it's worth noting that the proposed model's performance was evaluated within a SNR range of 0 to 35. This limited range does not provide a comprehensive understanding of the model's robustness, particularly in scenarios with high levels of noise.

In Table 1, some techniques for noise prediction are summarized and compared. The methodology proposed in this article considers several variables that influence signal quality for data generation. The study incorporates noise power density across a wide range of values and explores a spectrum sensing environment where multiple users are in motion at fixed speeds over time. Several regression models are compared using various metrics to highlight the robustness of the proposed method.

Table 1. Related Works on noise prediction.

Reference	Research Direction	Contribution	Limitation
[15]	SNR estimation method based on the sounding reference signal and a deep learning network	Proposed a DICNN for SNR estimation, which is a DnCNN and IRCNN in parallel	The number of testing samples is small
[16]	Method for estimating SNR in LTE systems and 5G	They used a CNN-LSTM neural network to extract spatial and temporal features	The proposed method exhibited lower performance in the frequency-domain

Table 1. Cont.

Reference	Research Direction	Contribution	Limitation
[17]	Novel neural network for channel estimation in the presence of unknown noise levels	Proposed an NDR-Net for channel estimation, which comprises a noise level estimation subnet, a DnCNN, and a residual learning cascade	Limited to a small range of noise level
Proposed	Prediction of noise power density in spectrum sensing signals	Proposed several regression algorithms for predicting noise power density in signals considering several other variables that influence the quality of the signal	Computing power was a limitation for training with more data and robust architectures

2.2. Distances Prediction

In [18], the authors proposed a deep learning approach for user equipment positioning in non-line-of-sight scenarios. The impact of variables such as the type of radio data, the number of base stations, the size of the training dataset, and the generalization of the trained models on 3GPP indoor factory scenarios was analyzed. The model trained consisted of a customized residual neural network (ResNet) for the path gain dataset and a ResNet-18 for the channel impulse response dataset. The metric used was the 90% quantile of the cumulative distribution function of the horizontal positioning error. The authors obtained the best performance with a large number of samples for training and tuning the models.

In [19], the authors proposed a machine learning algorithm for indoor fingerprint positioning based on measured 5G signals. The dataset was created by collecting 5G signals in the positioning area and processing them to form fingerprint data. A CNN was trained to locate a 5G device in an indoor environment. The metrics used were the root mean square error (RMSE) and the circular error probable. The experiments were conducted in a real field and demonstrated a positioning accuracy of 96% for the proposed method.

Finally, in [20], the authors introduced a location-aware predictive beamforming approach utilizing deep learning techniques for tracking unmanned aerial vehicle communication beams in dynamic scenarios. Specifically, they designed a recurrent neural network called LRNet, based on LSTM, to accurately predict unmanned aerial vehicle locations. Using the predicted location, it was possible to determine the angle between the unmanned aerial vehicle and the base station for efficient and rapid beam alignment in the subsequent time slot. This ensures reliable communication between the unmanned aerial vehicle and the base station. Simulation results demonstrate that the proposed approach achieves a highly satisfactory unmanned aerial vehicle-to-base station communication rate, approaching the upper limit attained by a perfect genie-aided alignment scheme.

In Table 2, some techniques for distances prediction are summarised and compared. The methodology proposed in this article considers several variables that influence signal quality for data generation. The study incorporates Euclidean distances between users across a wide range of values and explores a spectrum sensing environment where multiple users are in motion at fixed speeds over time. Several regression models are compared using various metrics to highlight the robustness of the proposed method.

Table 2. Related Works on distances prediction.

Reference	Research Direction	Contribution	Limitation
[18]	User equipment positioning in non-line-of-sight scenarios	Proposed customized ResNet for the path gain dataset and a ResNet-18 for the channel impulse response dataset	Without finetune and with large number of samples the models exhibited lower performance
[19]	Indoor fingerprint positioning based on measured 5G signals	A CNN was trained to locate a 5G device in an indoor environment. The experiments were conducted in a real field and demonstrated a positioning accuracy of 96% for the proposed method	The proposed method is not compared with other deep learning models
[20]	Location-aware predictive beamforming approach utilizing deep learning techniques for tracking unmanned aerial vehicle communication beams in dynamic scenarios	Designed a recurrent neural network called LRNet, based on LSTM, to accurately predict unmanned aerial vehicle locations. Using the predicted location, it was possible to determine the angle between the unmanned aerial vehicle and the base station for efficient and rapid beam alignment in the subsequent time slot	Limited only to unmanned aerial vehicle-to-base station communication
Proposed	Prediction of initial and final distances between users during spectrum sensing	Proposed several regression algorithms for predicting distances between users considering several other variables that influence the quality of the signal	Computing power was a limitation for training with more data and robust architectures

3. Methods

This section presents the proposed methods for data generation, noise and distance predict.

3.1. System Model

The use of generated data simulating PU signals is proposed in spectrum sensing to predict the noise level of the sensed signal and the initial and final distances between users. For this, the methodology is divided as follows: (1) database generation, where the signals that represent PUs are generated [1,2]; (2) training the proposed regression models [21,22]; and (3) evaluation of the trained models for predicting noise level and distance [23,24]. At the end of the process, it is expected that the models will be able to predict the level of noise and the initial and final distance between the PU and SU during the sensing period.

The dataset utilized for training the proposed regression models comprises signal data, noise levels, and distances between the users. Initially, the SU and PU are positioned within the area $X \times Y$, moving randomly for a duration of Δt . During this time, signals are sensed, and data on noise levels (N_0) as well as initial (D_i) and final (D_f) distances are collected. This collected data is then used to train and test the proposed regression models. To validate the effectiveness of the method in predicting noise levels and distances, various metrics are calculated.

The Figure 1 illustrates the representation of the environment just described. The Initial State (left) is an example of how users can be positioned within the area, while the Final State (right) demonstrates the change in position after Δt . The PU, marked by a red diamond, and the SUs, represented by black triangles, are randomly positioned within the area. In the Initial State, the heatmap represents the distance of each SU from the PU, with warmer (red) areas indicating closer proximity and cooler (blue) areas showing greater distances. In the Final State, we can observe that both the PU and SUs have changed positions, and the distances between the SUs may also vary during the observation period.

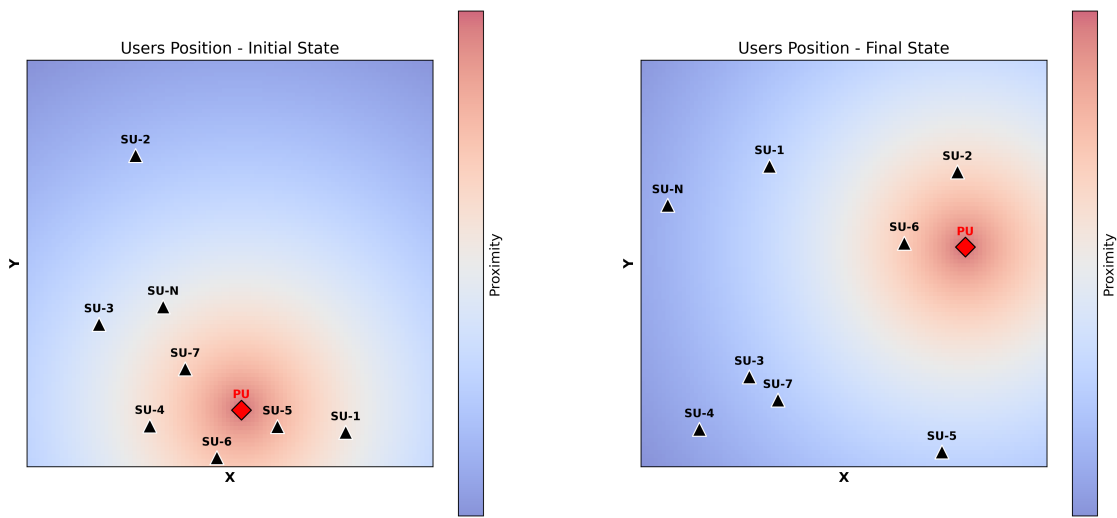


Figure 1. Users position over time (Initial State on the left, Final State on the right).

The output of step (1) is expected to be the signals that represent the PU, the associated noise level for each signal, and the initial and final distances during the sensing. Using this information, two regression models are trained: one for noise level prediction and another for initial and final distance predictions. In step (2), some classical machine learning regression models are employed, including Random Forest, Decision Tree, Extra Trees, XGBosst, LightGBM, SVR and Transformer. The output of this step consists of predicted values for noise level and distances given a test dataset. Lastly, in step (3), several metrics are used to evaluate the best models for these tasks.

3.2. Data Generation

In the spectrum sensing process, the decision on the channel condition is binary, involving two hypotheses: H_1 and H_0 [25]. Here, H_1 represents the hypothesis in which the PU is present, while H_0 represents the absence of the PU [2]. For the purposes of this paper, we will only consider hypotheses involving the presence of the PU. We assume that N_{SU} SU and a single PU are moving at a speed v , with their starting positions randomly chosen within a given area. As a result, the users' locations change over a time interval of Δt . Additionally, we are considering a multi-channel system with N_B bands, each having a bandwidth of B_W . Furthermore, we assume that the PU can utilize N_{B_P} consecutive bands [1]. Therefore, the received signal of the i -th SU on the j -th band at time n can be described as

$$y_i^j(n) = \begin{cases} s_i^j(n) + w_i^j(n), & \text{for } H_1 \text{ and } j \in B_P \\ \sqrt{\eta} s_i^j(n) + w_i^j(n), & \text{for } H_0 \text{ and } j \in B_A \end{cases} \quad (1)$$

where $s_i^j(n) = \kappa_i(n)g_i^j(n)x(n)$ and $w_i^j(n)$ is the additive white Gaussian noise (AWGN) whose noise power density is N_0 , mean zero and standard deviation $\sigma = \sqrt{B_W 10^{\frac{N_0}{10}}}$. Being η the proportion of power leaked to adjacent bands, then B_P are the bands occupied by the PU and B_A are the bands affected by the leaked power of the PU.

In the expression $s_i^j(n)$, a simplified path loss model is utilized, which can be written as follows:

$$\kappa_i(n) = \sqrt{\frac{P}{\beta(d_i(n))^\alpha 10^{\frac{h_i(n)}{10}}}}, \quad (2)$$

where α and β denote the path-loss exponent and path-loss constant, respectively. Here, $d_i(n)$ represents the Euclidean distance between the PU and SU i at time n . The shadow fading of the channel, indicated by $h_i(n)$, between the PU and SU i at time n in decibels (dB) can be described by a normal distribution with a mean of zero and a variance of σ^2 . The term P denotes the power transmitted by the PU within a specified frequency band. Furthermore, the multipath fading factor, denoted as $g_i^j(n)$, is modeled as an independent zero-mean circularly symmetric complex Gaussian (CSCG) random variable. Moreover, the data transmitted at time n , represented by $x(n)$, has an expected value of one [1,2].

3.3. Regression Models

The machine learning regression models used to predict interference and distance between the PU and SUs are the Random Forest, Decision Tree, Extra Trees, XGBoosting, LightGBM, SVR and Transformer.

3.3.1. Random Forest

The Random Forest consists of a collection of trees denoted as $h(x; \theta_k), k = 1, \dots, K$. Here, x represents an input vector of length q , containing a correlated random vector X , while θ_k refers to independent and identically distributed random vectors. In the context of regression, assume that the observed data is drawn independently from the joint distribution of (X, Y) , where Y represents the numerical outcome. This dataset includes $n(q + 1)$ -tuples, namely $(x_1, y_1), \dots, (x_n, y_n)$ [26]. The prediction of the Random Forest regression is the unweighted average over the collection

$$h(x) = \frac{1}{K} \sum_{k=1}^K h(x; \theta_k) \quad (3)$$

3.3.2. Decision Tree

In the context of regression, the Decision Tree is based on recursively partitioning the input features space into regions and then assigning a constant value to each region. This constant value serves as the prediction for any data point that falls within that region. Assume that X is the input data, Y the target variable and θ represents the parameters that define the splits in the Decision Tree [27]. Let $h(X; \theta)$ be the predicted value for Y given input X and parameter θ . Given a set of n training samples (X_i, Y_i) , where $i = 1, 2, \dots, n$, the decision tree regressor seeks to find optimal split θ that minimize the sum of square differences between the predicted value and the actual target value. The prediction for a given X can be represented as

$$h(X; \theta) = \sum_{i=1}^N c_i I(X \in R_i) \quad (4)$$

where N is the number of leaf nodes (regions) in the tree, c_i is the constant value associated with the leaf node R_i and $I(X \in R_i)$ is an indicator function equals 1 if X falls within the region R_i and 0 otherwise.

3.3.3. Extra Trees

The extra trees follows the same step-by-step as Random Forest, using a random subset of features to train each base estimator [26]. Although, the best feature and the corresponding value for splitting the node are randomly selects [28]. Random Forest uses a bootstrap replica to train the model, while the extra trees the whole training dataset to train each regression tree [29].

3.3.4. XGBoosting

XGBoosting (XGB) is a highly optimized distributed gradient boosting library. It employs a recursive binary splitting strategy to identify the optimal split at each stage, leading to the construction of the best possible model [30]. Due to its tree-based structure, XGB is robust to outliers and, like many boosting methods, is effective in countering overfitting, making model selection more manageable. The regularized objective of the XGB model during the t^{th} training step [31] is illustrate in Equation 5. Here, $l(y_{\text{pred}}^{(t)}, y_{\text{truth}})$ represents the loss, which quantifies the disparity between the prediction of the imputed missing value $y_{\text{pred}}^{(t)}$ and the corresponding ground truth y_{truth} .

$$L^{(t)} = \sum_i l(y_{\text{pred}}^{(t)}, y_{\text{truth}}) + \sum_k \Omega(f_k) \quad (5)$$

where $\Omega(f_k)$ is the regularizer representing the complexity of the k^{th} tree.

3.3.5. LightGBM

The LightGBM (LGBM) is the gradient boosting decision tree (GBDT) algorithm with gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB). The GOSS technique is employed within the context of gradient boosting, utilizing a training set consisting of n instances $\{x_1, \dots, x_n\}$, where each instance x_i represents a s -dimensional vector in space χ^s . In every iteration of gradient boosting, we compute the negative gradients of the loss function relative to the model's output, resulting in $\{g_1, \dots, g_n\}$. These training instances are then arranged in descending order, based on the absolute values of their gradients, and we select the top- $a \times 100\%$ instances with the largest gradient magnitudes to constitute a subset A [32]. For the complementing set A^c , comprising $(1 - a) \times 100\%$ of instances characterized by smaller gradients, a random subset B is extracted, sized at $b \times |A^c|$. The division of instances is subsequently determined by the estimated variance gain concerning vector $V_j(d)$ over the combined subset $A \cup B$, where

$$V_j(d) = \frac{1}{n} \left(\frac{(\sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i)^2}{n_l^j(d)} + \frac{(\sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i)^2}{n_r^j(d)} \right) \quad (6)$$

where $A_l = \{x_i \in A : x_{ij} \leq d\}$, $A_r = \{x_i \in A : x_{ij} > d\}$, $B_l = \{x_i \in B : x_{ij} \leq d\}$, $B_r = \{x_i \in B : x_{ij} > d\}$, and $\frac{1-a}{b}$ is the coefficient used to normalize the the sum of the gradients over B back to the size of A^c .

3.3.6. SVR

Given a n training data (X_i, Y_i) , where $i = 1, 2, \dots, n \subset \chi \times \mathbb{R}$, being χ the space of the input patterns [33]. The goal of the ε -SVR is to find a function which exhibits a maximum deviation of ε or less from the target values y_i obtained during training, while also maintaining a minimal degree of fluctuation or variability. This function can be described as

$$f(x) = \langle w, x \rangle + b, \text{ with } w \in \chi, b \in \mathbb{R} \quad (7)$$

where $\langle \cdot, \cdot \rangle$ denotes dot product in χ .

3.3.7. Transformer

A Transformer model consists of an encoder and a decoder, each composed of multiple layers of self-attention and feed-forward neural networks. The base structure of the Transformer is the self-attention mechanism. Given an input sequence, the self-attention computes a weighted sum of the values. Multi-head attention is used to capture different aspects of relationships. Let the input of a Transformer layer be $X \in \mathbb{R}^{n \times d}$, where n is the number of tokens and d is the dimension of each token. Then, one block layer can be a function $f_{\theta}(X) =: Z$ defined by [37]:

$$A = \frac{1}{\sqrt{d}} XQ(XK)^T \quad (8)$$

$$\hat{X} = \text{softmax}(A)(XV) \quad (9)$$

$$M = \text{LayerNorm}_1(\hat{X}O + X) \quad (10)$$

$$F = W_2 * (\sigma(M * W_1 + b_1)) + b_2 \quad (11)$$

$$Z = \text{LayerNorm}_2(M + F) \quad (12)$$

where Equations 8, 9 and 10 referred to attention computation, and Equations 11 and 12 referred to the feed forward network layer. $\text{softmax}(\cdot)$ is the row-wise softmax function, $\text{LayerNorm}(\cdot)$ is the layer normalization function, and σ to activation function. Q, K, V and $O \in \mathbb{R}^{d \times d}$, $W_1 \in \mathbb{R}^{d \times d_f}$, $b_1 \in \mathbb{R}^{d_f}$, $W_2 \in \mathbb{R}^{d_f \times d}$, $b_2 \in \mathbb{R}^d$ are the training parameters in the layer [37]. In Figure 2 is shown the architecture of the proposed Transformer.

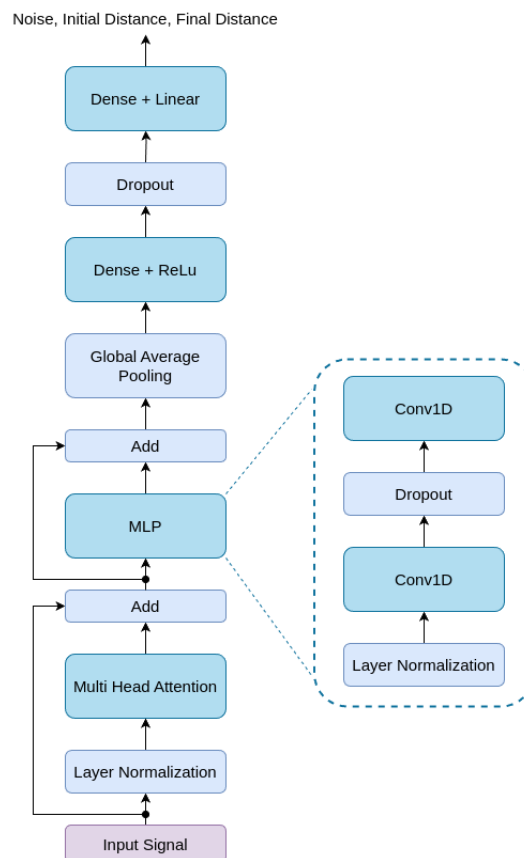


Figure 2. Architecture of the proposed Transformer.

3.4. Evaluation Metrics

The metrics used for evaluation of the regression models are the mean square error, mean absolute error, root mean square error, mean absolute percentage error, R-square and correlation coefficient [34].

- Mean square error (MSE) [34]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (13)$$

where n is the number of data points in the dataset, y_i represents the actual target value of the i -th data point, and \hat{y}_i represents the predicted value of the i -th data point.

- Mean absolute error (MAE) [34]:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (14)$$

where $|\cdot|$ denotes the absolute value.

- Root mean square error (RMSE) [34]:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (15)$$

- Mean absolute percentage error (MAPE) [34]:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i|} \times 100 \quad (16)$$

- R-square (R^2) [23]:

$$R^2 = 1 - \frac{SSE}{SST} \quad (17)$$

where SSE represents the sum of squared differences between the the actual target values and the predicted values. The SST represents the total sum of squares, which is the sum of squared differences between the actual target values and their mean.

- Correlation coefficient [35]:

The correlation coefficient is given by the Pearson correlation coefficient:

$$r = \frac{\sum_{i=1}^n (y_i - y_{mean})(\hat{y}_i - \hat{y}_{mean})}{\sqrt{\sum_{i=1}^n (y_i - y_{mean})^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - \hat{y}_{mean})^2}} \quad (18)$$

where y_{mean} is the mean of the actual target values and \hat{y}_{mean} is the mean of the predicted values.

4. Experiments and Results

This section presents the experiments conducted and the results achieved for data generation, noise and distance predict.

4.1. Data Generation

The first stage of the experiments involves signal generation. It is assumed that multiple SU and a single PU are moving at a velocity of $v = 3$ km/h, and their initial positions are randomly chosen within an area of 250 meters \times 250 meters. As a result, the users' positions change over a time period of $\Delta t = 5$ seconds. Each occupied band N_B has bandwidth B_W of 10 MHz, and the PU can simultaneously use 1 to 3 bands. Additionally, $P = 23$ dBm, $\beta = 10^{3.453}$, $\alpha = 3.8$, $\sigma = 7.9$ dB, and N_0 is randomly chosen between -114 and -174 dBm/Hz. The ratio of leaked power to adjacent bands, η , is 10 dBm, resulting in leaked power to adjacent bands being half of the PU signal power. For the

experiments, 42,000 instances were generated, divided into 80% for training and 20% for testing. For the Transformer method, the training dataset was divided into 80% for training and 20% for validation. A total of 1,024 samples per second of the signal were generated. In Table 3 is presented the parameters and their respective values.

Table 3. Parameters and values for data generation.

Parameter	Value	Mean
v	3 km/h	Velocity
Δt	5 seconds	Time period
B_W	10 MHz	Bandwidth of each N_B band
N_{Bp}	1 to 3	Number of consecutive bands that the PU can used
P	23 dBm	Power transmitted by the PU
α	3.8	Path-loss exponent
β	$10^{3.453}$	Path-loss constant
σ	7.9 dB	Standard deviation
N_0	-114 to -174 dBm/Hz	Noise power density
η	10 dBm	Leaked power to adjacent bands

4.2. Model Parameters

The Random Forest, Decision Tree, Extra Trees, and SVR models were parameterized following the default settings of the *scikit-learn* library. In Table 4, the parameters of the XGB, and LGBM models are presented. And, in the Table 5, the parameters of the Transformer model is presented.

Table 4. XGB and LGBM parameters.

Parameter	XGB	LGBM
Estimators	100	100
Max depth	6	-
Learning rate	0.1	0.05
Subsample	0.8	-
Column sample by tree	0.8	-
Random state	42	42
Boosting type	-	GBDT
Number of leaves	-	31

Table 5. Transformer parameters.

Parameter	Transformer
Head size	32
Number of heads	4
Filter dimension	32
Transformer block	1
Dense layer	32
Drop out	0.25
Batch size	32
Optimizer	Adam
Learning rate	0.001
Loss	MSE

4.3. Noise Predict

Figure 3 (left) presents the graph of the MSE with the variation of the N_0 in the proposed regression methods. It is noticeable that the SVR had the worst performance, only achieving good results at an N_0 of -144 dBm/Hz. The Random Forest, Extra Trees, XGB, LGBM, Transformer, and Decision Tree

showed similar performances, as shown in the graph. However, the Random Forest, XGB, LGBM and Extra Trees exhibited better performance, especially at lower levels of N_0 . The XGB was the best model, achieving the lowest level of MSE, achieving 15.53, which indicates that the predicted values are close to the actual ones.

Figure 3 (right) presents the graph of the MAE with the variation of the N_0 in the proposed regression methods. It is noticeable, also, that the SVR had the worst performance, only achieving good results at an N_0 of -144 dBm/Hz. The Random Forest, Extra Trees, XGB, LGBM, Transformer, and Decision Tree showed similar performances, as shown in the graph. However, the Random Forest, XGB, LGBM and Extra Trees exhibited better performance, especially at levels of -164 dBm/Hz and -154 dBm/Hz. The Random Forest was the best model, achieving the lowest level of MAE, achieving 1.9473, which indicates that the predicted values are close to the actual ones.

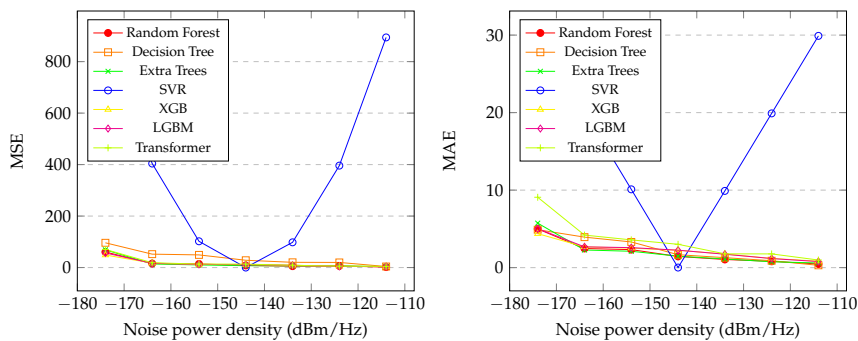


Figure 3. Graphic of the MSE (left) and MAE (right) of proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

Figure 4 (left) presents the graph of the RMSE with the variation of the N_0 in the proposed regression methods. It is noticeable, also, that the SVR had the worst performance, only achieving good results at an N_0 of -144 dBm/Hz. The Random Forest, XGB, LGBM, Transformer and Extra Trees showed similar performances, as shown in the graph. The XGB was the best model, achieving the lowest level of RMSE, achieving 3.94, which indicates that the predicted values are close to the actual ones.

Figure 4 (right) presents the graph of the MAPE with the variation of the N_0 in the proposed regression methods. It is noticeable, also, that the SVR had the worst performance, only achieving good results at an N_0 of -144 dBm/Hz. The Random Forest, XGB, LGBM Extra Trees, Transformer, and Decision Tree showed similar performances, as shown in the graph. However, the Random Forest and Extra Trees exhibited better performance. The Random Forest was the best model, achieving the lowest level of MAPE, achieving 1.2534, which indicates that the predicted values are close to the actual ones.

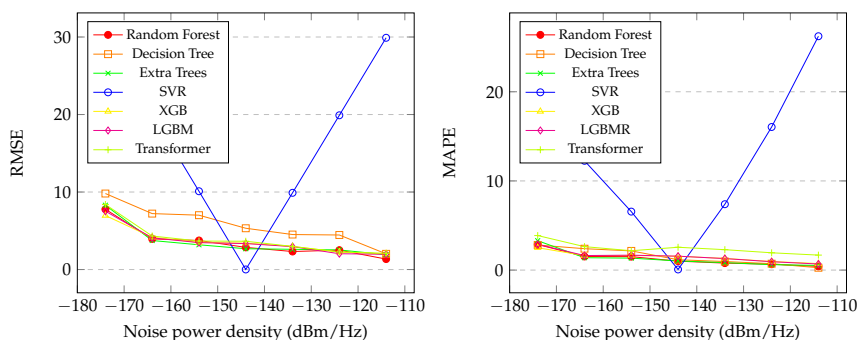


Figure 4. Graphic of the RMSE (left) and MAPE (right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

In Table 6, the general metrics for all proposed models are presented for noise prediction. It is worth noting that, in terms of the correlation coefficient, Random Forest, XGB, LGBM, Transformer

and Extra Trees exhibited the highest and similar performances. However, Random Forest performed slightly better on the MAE and MAPE metrics, while XGB performed better on the correlation coefficient, MSE, RMSE, and R^2 . The SVR presented the worst performance in the general metrics.

Table 6. Noise regression comparison metrics of the proposed methods.

Metrics	R.F.	D.T.	E.T.	SVR	XGB	LGBM	Transformer
C.C.	0.9801	0.9521	0.9794	0.0079	0.9806	0.979	0.9697
MSE	16.084	38.6547	16.8942	404.64335	15.53	16.35	19.2086
MAE	1.9473	2.32738	2.0186	17.2736	2.23	2.32	3.4854
RMSE	4.0104	6.2172	4.11026	20.11575	3.94	4.04	4.3827
MAPE	1.2534	1.49406	1.2971	12.3579	1.48	1.52	2.4489
R^2	0.96025	0.9044	0.9582	$-4.0934e^{-06}$	0.9616	0.959	0.9365

4.4. Distance Predict

Figure 5 presents the graphics of the MSE of the initial and final distance between the SU and PU during the spectrum sensing with the variation of N_0 in the proposed regression methods. It is noticeable that the SVR and Decision Tree had the worst performances at all levels of noise. The Random Forest, XGB, Transformer, LGBM and Extra Trees showed similar performances, as shown in the graph. However, the LGBM exhibited slightly better performance than the others, especially at the lowest level of noise, achieving 100.67. In the final distance prediction, the Extra Trees model exhibited slightly better performance, achieving 107.77.

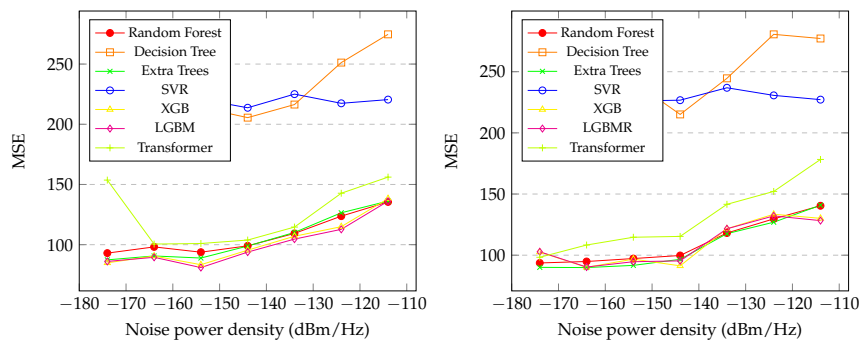


Figure 5. Graphic of the MSE (initial distance at the left and final distance at the right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

Figure 6 presents the graphics of the MAE of the initial and final distance between the SU and PU during the spectrum sensing with the variation of N_0 in the proposed regression methods. It is also noticeable that the SVR and Decision Tree had the worst performances at all levels of noise. The Random Forest, XGB, LGBM, Transformer and Extra Trees showed similar performances, as shown in the graph. However, the XGB exhibited slightly better performance than the others, achieving 7.19. In the final distance prediction, the Extra Trees model exhibited slightly better performance, achieving 7.33.

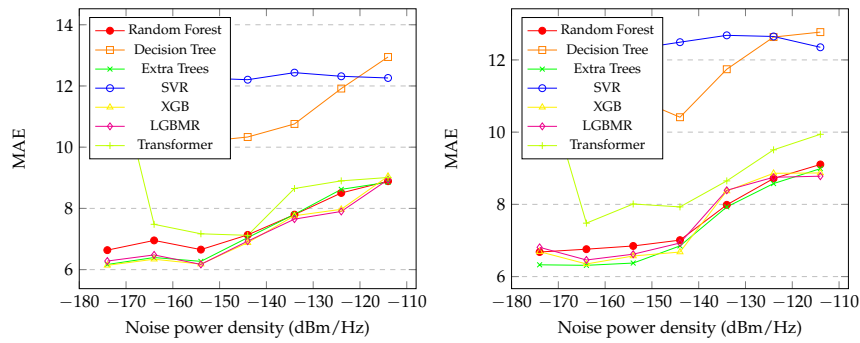


Figure 6. Graphic of the MAE (initial distance at the left and final distance at the right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

Figure 7 presents the graphics of the RMSE of the initial and final distance between the SU and PU during the spectrum sensing with the variation of N_0 in the proposed regression methods. It is also noticeable that the SVR and Decision Tree had the worst performances at all levels of noise. The Random Forest, XGB, LGBM, Transformer and Extra Trees showed similar performances, as shown in the graph. However, the LGBM exhibited slightly better performance than the others, achieving 10.03. In the final distance prediction, the Extra Trees model exhibited slightly better performance, achieving 10.38.

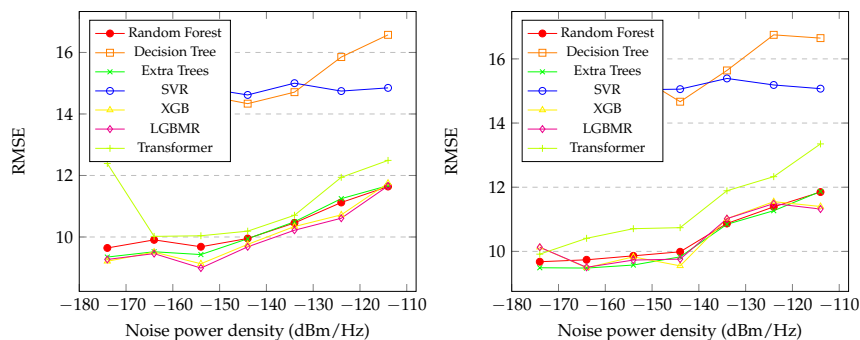


Figure 7. Graphic of the RMSE (initial distance at the left and final distance at the right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

Figure 8 presents the graphics of the MAPE of the initial and final distance between the SU and PU during spectrum sensing with variations in N_0 using the proposed regression methods. It is also noticeable that SVR and Decision Tree had the worst performance levels across all levels of noise. Random Forest and Extra Trees showed similar performance, as shown in the graph on the left. However, Transformer exhibited slightly better performance than Random Forest, especially at -174 dBm/Hz, -164 dBm/Hz, and -155 dBm/Hz, achieving 30.23. In the graph on the right, Random Forest, XGB, LGBM, and Extra Trees showed similar performance. However, Transformer exhibited slightly better performance than the others, achieving 31.25.

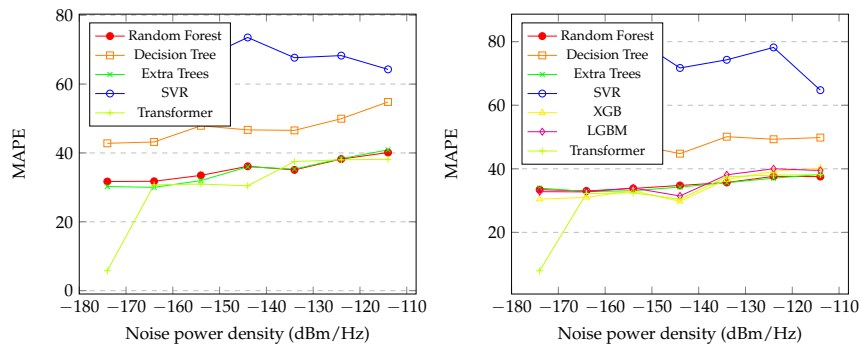


Figure 8. Graphic of the MAPE (initial distance at the left and final distance at the right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

In Table 7, general metrics for all proposed regression models are presented for initial distance prediction. It is worth noting that, in all metrics, Random Forest, XGB, LGBM, Transformer and Extra Trees exhibited the highest and similar performances. Transformer performed better in correlation coefficient, MAPE, and R^2 , demonstrating a high level of correlation between the predictions and the actual values of distances. In terms of MSE and RMSE, LGBM achieved the lowest values, indicating that the differences between predictions and actual values are small. The lowest MAE values were exhibited by XGB.

Table 7. Initial distance regression comparison metrics of the proposed methods.

Metrics	R.F.	D.T.	E.T.	SVR	XGB	LGBM	Transformer
C.C.	0.7222	0.4876	0.7290	0.08103	0.718	0.725	0.841
MSE	107.56	226.2186	105.5776	218.8497	102.17	100.67	124.63
MAE	7.5115	10.9095	7.31	12.2795	7.19	7.20	8.87
RMSE	10.3713	15.0405	10.275	14.7935	10.10	10.03	11.16
MAPE	35.2191	47.4401	34.69	68.5288	inf	inf	30.23
R^2	0.51807	-0.0135	0.5269	0.00399	0.516	0.523	0.58

In Table 8, general metrics for all proposed regression models are presented for final distance prediction. The Extra Trees exhibited the best performance in all metrics except for MAPE, R^2 and correlation coefficient, where Transformer exhibited the best performance.

Table 8. Final distance regression comparison metrics of the proposed methods.

Metrics	R.F.	D.T.	E.T.	SVR	XGB	LGBM	Transformer
C.C.	0.7225	0.47	0.7321	0.0923	0.712	0.714	0.821
MSE	110.61	245.51	107.77	228.4979	109.51	109.36	129.805
MAE	7.58	11.405	7.33	12.45	7.49	7.54	9.06
RMSE	10.51	15.66	10.38	15.1161	10.46	10.45	11.39
MAPE	35.13	47.78	35.01	74.74	34.36	35.52	31.25
R^2	0.5182	-0.0069	0.5306	0.004	0.5081	0.5083	0.573

5. Discussion

In Section 4.3, the results for noise predict were presented, and several important aspects need to be highlighted. In Figure 3 (left), it is noticeable that the models achieved similar performances, except for Decision Tree, and SVR. Interestingly, the best performance occurred at the highest levels of noise, except for SVR, which specialized in a single noise level, -144 dBm/Hz. Figure 3 (right), Figure 4 (left), and Figure 4 (right) exhibit similar characteristics to Figure 3 (left). The overall results for Random Forest, XGB, LGBM, Transformer and Extra Trees are similar across all metrics, with Random Forest and XGB performing particularly well, as demonstrated in Table 6. In contrast, SVR

exhibited poor performance overall. Additionally, the correlation coefficients reveal that Random Forest, XGB, LGBM and Extra Trees achieved strong correlations between predicted and real noise values (Table 6). Due to the limited computational resources and the large amount of generated data, it was not possible to enhance the robustness of the Transformer architectures. Furthermore, a search for optimal hyperparameters values for all models could be a proposed enhancement for the work.

In Section 4.4, the results for initial and final distance predict were presented, and several notable observations arise. In Figure 5, it is evident that Extra Trees and LGBM achieved the highest performances. Unlike the noise predict, the best performances for both distance predicts occurred at the lowest levels of noise for all models, which was expected. The behavior of predict the initial distance mirrors that of predict the final distance, as seen in Figures 5–8. The general results for Random Forest, XGB, LGBM, Transformer and Extra Trees exhibit similarity across all metrics, with Extra Trees, Transformer and LGBM slightly outperforming in almost all metrics for both initial and final distance, as shown in Tables 7 and 8, respectively. In contrast, Decision Tree and SVR showed poor performance overall. Moreover, the correlation coefficients reveal that the predicted distances exhibited indices greater than 0.82 in relation to the real values in both distances, initial and final. Interesting factor is that in the initial distance prediction, the MAPE values for the XGB and LGBM models returned 'inf'. More studies are needed to understand what happened.

6. Conclusions

In this paper, noise and distance prediction based on spectrum sensing signals using regression models was proposed. The conducted experiments have shown that the proposed methods hold promise for predict noise levels, as well as the initial and final distances between the PU and SU. The correlation coefficient value for XGB is the highest and closest to one (Table 6), indicating a strong correlation between predicted and actual noise values in the test database. As a result, the proposed methods can greatly benefit various applications, especially in telecommunication and networking, enabling the design of communication systems that meet appropriate requirements for ensuring reliable and efficient data transfer.

Additionally, the predicts for the initial and final distances between PU and SU are presented as results. The conducted experiments have demonstrated that the proposed methods show promise in predicting distances between users. The correlation coefficient values for Transformer are the highest, exceeding 0.82 (Tables 7 and 8), which implies a good level of correlation between the predicted distances and the actual distances in the test database. It's important to highlight that the number of possible noise levels is limited to 7 levels (-114 to -174 dBm/Hz), while the number of possible distances is unknown since the distance was chosen randomly within a certain range of the area. Therefore, there may be a difference in performance between the two approaches. Hence, the proposed methods hold the potential to benefit numerous applications, including signal attenuation and path loss, interference and frequency reuse, fading and multipath effects, localization and tracking, and power control.

Author Contributions: Conceptualization, Myke Valadão; methodology, Myke Valadão; software, Myke Valadão; validation, Myke Valadão; formal analysis, Myke Valadão; investigation, Myke Valadão; resources, Myke Valadão; data curation, Myke Valadão; writing—original draft preparation, Myke Valadão and Diego Amoedo; writing—review and editing, Myke Valadão, Diego Amoedo, André Costa, Celso Carvalho and Waldir Sabino; visualization, Myke Valadão, Diego Amoedo, André Costa, Celso Carvalho and Waldir Sabino; supervision, Myke Valadão, André Costa, Celso Carvalho and Waldir Sabino; project administration, Myke Valadão; funding acquisition, Myke Valadão, Celso Carvalho and Waldir Sabino. All authors have read and agreed to the published version of the manuscript.

References

1. Valadão, M. D., Amoedo, D., Costa, A., Carvalho, C., & Sabino, W. (2021). Deep cooperative spectrum sensing based on residual neural network using feature extraction and random forest classifier. *Sensors*, *21*(21), 7146. <https://doi.org/10.3390/s21217146>

2. Lee, W., Kim, M., & Cho, D. H. (2019). Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks. *IEEE Transactions on Vehicular Technology*, 68(3), 3005-3009. 10.1109/TVT.2019.2891291
3. Valadão, M. D., Carvalho, C. B., & Júnior, W. S. (2020). Trends and challenges for the spectrum sensing in the next generation of communication systems. In *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)* (pp. 1-2). IEEE. 10.1109/ICCE-Taiwan49838.2020.9258205
4. Valadão, M. D., Amoedo, D. A., Pereira, A. M., Tavares, S. A., Furtado, R. S., Carvalho, C. B., ... & Júnior, W. S. (2022). Cooperative spectrum sensing system using residual convolutional neural network. In *2022 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1-5). IEEE. 10.1109/ICCE53296.2022.9730218
5. Valadão, M. D., Júnior, W. S., & Carvalho, C. B. (2021). Trends and Challenges for the Spectrum Efficiency in NOMA and MIMO based Cognitive Radio in 5G Networks. In *2021 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1-4). IEEE. 10.1109/ICCE50685.2021.9427695
6. Valadão, M., Silva, L., Serrão, M., Guerreiro, W., Furtado, V., Freire, N., ... & Craveiro, C. (2023). MobileNetV3-based Automatic Modulation Recognition for Low-Latency Spectrum Sensing. In *2023 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1-5). IEEE. 10.1109/ICCE56470.2023.10043380
7. Gupta, M. S., & Kumar, K. (2019). Progression on spectrum sensing for cognitive radio networks: A survey, classification, challenges and future research issues. *Journal of Network and Computer Applications*, 143, 47-76. <https://doi.org/10.1016/j.jnca.2019.06.005>
8. Arjoune, Y., & Kaabouch, N. (2019). A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions. *Sensors*, 19(1), 126. <https://doi.org/10.3390/s19010126>
9. Shen, X., Shi, D., Peksi, S., & Gan, W. S. (2022). A multi-channel wireless active noise control headphone with coherence-based weight determination algorithm. *Journal of Signal Processing Systems*, 94(8), 811-819. <https://doi.org/10.1007/s11265-022-01749-4>
10. Cheon, B. W., & Kim, N. H. (2022). AWGN Removal Using Modified Steering Kernel and Image Matching. *Applied Sciences*, 12(22), 11588. <https://doi.org/10.3390/app122211588>
11. Ngo, T., Kelley, B., & Rad, P. (2020). Deep learning based prediction of signal-to-noise ratio (SNR) for LTE and 5G systems. In *2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM)* (pp. 1-6). IEEE. 10.1109/WINCOM50532.2020.9272470
12. Nagah Amr, Mohammed, Hussein M. ELAttar, Mohamed H. Abd El Azeem, and Hesham El Badawy. (2021). "An Enhanced Indoor Positioning Technique Based on a Novel Received Signal Strength Indicator Distance Prediction and Correction Model" *Sensors* 21, no. 3: 719. <https://doi.org/10.3390/s21030719>
13. Jain, P., Choudhury, A., Dutta, P., Kalita, K., & Barsocchi, P. (2021). Random forest regression-based machine learning model for accurate estimation of fluid flow in curved pipes. *Processes*, 9(11), 2095. <https://doi.org/10.3390/pr9112095>
14. Wang, X., Zhao, Y., & Pourpanah, F. (2020). Recent advances in deep learning. *International Journal of Machine Learning and Cybernetics*, 11, 747-750. <https://doi.org/10.1007/s13042-020-01096-5>
15. Yao, G., & Hu, Z. (2023, June). SNR Estimation Method based on SRS and DInet. In *Proceedings of the 2023 15th International Conference on Computer Modeling and Simulation* (pp. 218-224). <https://doi.org/10.1145/3608251.3608262>
16. Ngo, T., Kelley, B., & Rad, P. (2020, October). Deep learning based prediction of signal-to-noise ratio (SNR) for LTE and 5G systems. In *2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM)* (pp. 1-6). IEEE. 10.1109/WINCOM50532.2020.9272470
17. Li, Y., Bian, X., & Li, M. (2023). Denoising Generalization Performance of Channel Estimation in Multipath Time-Varying OFDM Systems. *Sensors*, 23(6), 3102. <https://doi.org/10.3390/s23063102>
18. Chatelier, B., Corlay, V., Ciochina, C., Coly, F., & Guillet, J. (2023). Influence of Dataset Parameters on the Performance of Direct UE Positioning via Deep Learning. *arXiv preprint arXiv:2304.02308*. <https://doi.org/10.48550/arXiv.2304.02308>
19. Wang, C., Xi, J., Xia, C., Xu, C., & Duan, Y. (2023, January). Indoor fingerprint positioning method based on real 5G signals. In *Proceedings of the 2023 7th International Conference on Machine Learning and Soft Computing* (pp. 205-210). <https://doi.org/10.1145/3583788.3583819>
20. Conti, A., Morselli, F., Liu, Z., Bartoletti, S., Mazuelas, S., Lindsey, W. C., & Win, M. Z. (2021). Location awareness in beyond 5G networks. *IEEE Communications Magazine*, 59(11), 22-27. 10.1109/LWC.2020.3045150
21. Yang, H., Xie, X., & Kadoch, M. (2020). Machine learning techniques and a case study for intelligent wireless networks. *IEEE Network*, 34(3), 208-215. 10.1109/MNET.001.1900351

22. Arjoune, Y., & Kaabouch, N. (2019). On spectrum sensing, a machine learning method for cognitive radio systems. *In 2019 IEEE International Conference on Electro Information Technology (EIT)* (pp. 333-338). IEEE. 10.1109/EIT.2019.8834099
23. Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, e623. <https://doi.org/10.7717/peerj-cs.623>
24. Spüler, M., Sarasola-Sanz, A., Birbaumer, N., Rosenstiel, W., & Ramos-Murguialday, A. (2015). Comparing metrics to evaluate performance of regression methods for decoding of neural signals. *In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 1083-1086). IEEE. 10.1109/EMBC.2015.7318553
25. Cichoń, K., Kliks, A., & Bogucka, H. (2016). Energy-efficient cooperative spectrum sensing: A survey. *IEEE Communications Surveys & Tutorials*, 18(3), 1861-1886. 10.1109/COMST.2016.2553178
26. Segal, M. R. (2004). Machine Learning Benchmarks and Random Forest Regression. *UCSF: Center for Bioinformatics and Molecular Biostatistics*. Retrieved from <https://escholarship.org/uc/item/35x3v9t4>
27. Luo, H., Cheng, F., Yu, H., & Yi, Y. (2021). SDTR: Soft decision tree regressor for tabular data. *IEEE Access*, 9, 55999-56011. 10.1109/ACCESS.2021.3070575
28. Ahmad, M. W., Reynolds, J., & Rezgui, Y. (2018). Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *Journal of cleaner production*, 203, 810-821. <https://doi.org/10.1016/j.jclepro.2018.08.207>
29. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63, 3-42. <https://doi.org/10.1007/s10994-006-6226-1>
30. Zhang, X., Yan, C., Gao, C., Malin, B. A., & Chen, Y. (2020). Predicting missing values in medical data via XGBoost regression. *Journal of healthcare informatics research*, 4, 383-394. <https://doi.org/10.1007/s41666-020-00077-1>
31. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). <https://doi.org/10.1145/2939672.2939785>
32. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30. <https://dl.acm.org/doi/10.5555/3294996.3295074>
33. Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14, 199-222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>
34. Botchkarev, A. (2018). Evaluating performance of regression machine learning models using multiple error metrics in azure machine learning studio. *Available at SSRN 3177507*. <http://dx.doi.org/10.2139/ssrn.3177507>
35. Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., ... & Cohen, I. (2009). Pearson correlation coefficient. *Noise reduction in speech processing*, 1-4. https://doi.org/10.1007/978-3-642-00296-0_5
36. Su, X., Li, J., & Hua, Z. (2022). Transformer-based regression network for pansharpening remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1-23. <https://doi.org/10.1109/TGRS.2022.3152425>
37. Min, E., Chen, R., Bian, Y., Xu, T., Zhao, K., Huang, W., ... & Rong, Y. (2022). Transformer for graphs: An overview from architecture perspective. *arXiv preprint arXiv:2202.08455*. <https://doi.org/10.48550/arXiv.2202.08455>

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.