# Prediction of Digital Terrestrial Television Coverage Using Machine Learning Regression

Carla E. Garcia Moreta, Mario R. Camana Acosta, and Insoo Koo, *Member, IEEE*

***Abstract*—Appropriate coverage prediction is a fundamental task for an operator during the dimensioning process and planning of a digital terrestrial television (DTT) system because it allows offering a satisfactory quality of service to end users. Accordingly, several prediction methods based on propagation path loss estimation and traditional statistical models have been proposed. However, the choice of model depends on many factors, such as the presence of obstacles (buildings, trees, and so on) and propagation paths. This fact leads to increasing the error gap between the predicted and real value, which varies from one propagation model to the next. Therefore, novel techniques are required to achieve a high accuracy in the prediction of the signal strength based on few local measurements over the zone of interest. A machine learning regression algorithm is a novel approach that improves the accuracy of DTT coverage prediction regardless of the aforementioned constraints. To this end, we propose an approach based on clustering and machine regression algorithms, such as random forest regression, AdaBoost regression, and *K*-nearest neighbors regression, where we choose the best algorithm for our approach. We use real measurements in terms of electric field strength corresponding to eight DTT channels operating in the city of Quito, Ecuador. Furthermore, we display the coverage results in Google Maps. We perform extensively simulation analysis based on the tenfold cross validation method to evaluate the performance of the machine learning regressor algorithms and compare the results in three error metrics with support vector regression, lasso regression, multilayer perceptron regression, and ordinary kriging technique. Satisfactorily, the results using random forest regression depict a considerable improvement in the accuracy of coverage prediction under a low computational load.**

***Index Terms*—Digital terrestrial television, random forest regression, AdaBoost regressor, K-nearest neighbors (KNN) regression.**

## I. Introduction

THE IMPACT of digital terrestrial television (DTT) represents the digital change in television technology that has led to the process of transition from analog television to digital terrestrial television due to its benefits and technological improvements, such as managing the radioelectric spectrum more efficiently.

One of the most important requirements to ensure that the end user receives a quality service is network coverage planning, whose planning tool needs to manage a large number of transmitters, high-resolution databases for terrain elevation, accurate propagation models, and the propagation calculation of radio waves should be in an acceptable time. However, an acceptable computation time through the reduction of the required time and memory entail in a degradation of the prediction quality, affecting the accuracy of it [1]. In this sense, different approaches for new planning techniques are necessary to meet with this complex task. Unfortunately, several countries especially those in the developing world, have still failed to complete the process of analog shutdown, one of the main reasons for this fact is the lack of requirements for suitable dimensioning and network coverage planning of a digital terrestrial television system.

The accuracy of traditional statistical models, such as those from the International Telecommunications Union (ITU) [1] (ITU-R P.1546-5 [2] and ITU-R P-1812-4 [3]), from Okumura-Hata [4], and so on, depends on the similarities between the environment where the measurements are carried out and the environment to be surveyed. These models have been used for decades to predict signal quality in DTT systems, and many of them are based on the observation that the received signal is generally a mixture of delayed and attenuated copies of the same transmitted waveform. However, for the current generation of digital terrestrial television, the imperfections of propagation models become even more acute [5]. This fact raises the question of whether new or modified prediction methods are required to improve coverage prediction.

Mendes-Moreira *et al.* [6] described in details the process for ensemble based regression and summarized the principal approaches for regression. They discussed the advantages of combine several models in order to reduce the error in the estimation and increase the robustness when we compare with a single model. Finally, they refer to AdaBoost and random forest regression (RFR) as two of the main ensemble generation methods.

Our main objective is to propose a new and more accurate DTT coverage prediction method by using machine learning regression algorithms and real data corresponding to field strength measures for eight DTT channels, which were taken by utilizing global positioning system (GPS) coordinates in a zone located in Quito, Ecuador. We intend to overcome the

Manuscript received July 20, 2018; revised November 21, 2018; accepted February 5, 2019. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government [Ministry of Science and ICT (MSIT)] under Grant NRF-2018R1A2B6001714. *(Corresponding author: Insoo Koo.)*

The authors are with the Department of Electrical/Electronic and Computer Engineering, University of Ulsan, Ulsan 689-749, South Korea (e-mail: carli.garcia27@hotmail.com; mario_camana@hotmail.com; iskoo@ulsan.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2019.2901409

constraints presented by propagation models [7] that are based on the theory of electromagnetic waves and statistical evaluation, such as the dependence of uncertain path loss prediction between the transmitters and the coverage area due to obstacles and long trajectories. Our goal is to enhance the precision of DTT coverage prediction without increasing the computational load during the training procedure of each model.

Our main contributions are summarized as follows.

- A novel approach to machine learning regression algorithms is proposed to determine DTT coverage prediction. We consider electric field intensity measurements for eight DTT channels with their GPS coordiantes, which were taken in a specific area of Quito, Ecuador, by special equipment called the automatic radio spectrum control system (SACER) [8].
- Traditional statistical models and prediction methods based on propagation path loss estimation depend on many factors, such as the presence of obstacles and propagation paths. This fact affects the accuracy of DTT coverage prediction, which differs from one model to another. Therefore, we propose an approach based on clustering and a machine learning regressor algorithm to improve the accuracy of DTT coverage prediction regardless of the aforementioned constraints. Subsequently, the proposed scheme would be meaningful for planning a digital terrestrial television system, particularly in those countries that still have not accomplished the analog switch-off process.
- We study three candidate machine learning algorithms; random forest regression [9], AdaBoost regression [10] and K-nearest neighbor regression [11], which are evaluated by the 10-fold cross-validation technique to select the best regressor algorithm for our proposed approach by using the error metrics relative error, mean absolute error and root mean squared error (RMSE).
- Through numerical and graphical results, we evaluate and compare the performance of the studied machine learning algorithms and the ordinary kriging technique [12] for DTT coverage prediction. Furthermore, we compare the results with other regression algorithms such as support vector regression with radial basis function (RBF) kernel, lasso regression and neural network regression.
- We create a grid of $1000 \times 1000$ data geographic points, located in the same area of interest, to come up with results on the digital map tool Google Maps. In this way, the operators can observe clearly the coverage prediction and shadow areas in the digital terrestrial television coverage area. In this paper, we perform the coverage prediction considering eight DTT channels that currently operate in the city of Quito in Ecuador.

The paper is structured as follows. Related work is presented in Section II. In Section III, we give a brief description of digital terrestrial television based on the Brazilian−Japanese integrated services digital broadcasting−terrestrial Brazil (ISDB−Tb) standard, coverage areas, and measurement system specifications. The proposed methodology is outlined in Section IV, and the machine learning regression algorithms used in this paper are explained in Section V. We present the numerical and graphical results in Section VI. Finally, the conclusions are made in Section VII.

## II. Related Work

In the literature on DTT, there are several models for coverage prediction through propagation path loss estimation in outdoor environments. We first cover the main studies based on traditional statistical models. In that sense, García *et al.* [13] proposed optimizing the performance of the propagation model of ITU-R Recommendation P.1812-4 with the aim of improving its accuracy in propagation path loss predictions, where they used the bio-inspired particle swarm optimization (PSO) technique. This optimized model presented a significant increase in accuracy, compared with the original version and even other propagation models such as the Okumura-Hata model, and ITU-R Recommendation P.1546-5, and it nearly equaled the accuracy produced by the TDA-Venezuela-II model.

Silva *et al.* [7] compared the values predicted for the field strength measurements of a digital TV signal in five cities of Brazil by the following models: ITU-R Recommendation P. 525, Deygout-Assis Knife Edge, Deygout-Assis Rounded, Deygout-Assis Main Rounded, CRC-Predict, and ITU-R Recommendation P.1546. Their paper pointed out that, at large, ITU-R P.1546 is the model that best fit the field measurements, and had the best performance in the cities of São Paulo, Recife, and Brasilia, where values for RMSE were 11.8, 11.84, and 10.23, respectively. The authors in that paper concluded it is necessary to invest in more accurate elevation models with more resolution and that take the urban environment elements into consideration because the environment morphology is responsible for better adjustment to the models.

ITU BT.2137 [1] provides a brief outline of the results of comparisons between measured and predicted signal levels, which shows a wide divergence in terms of both mean error and standard deviation of error. This document describes coverage prediction methods applied in the United Kingdom, Japan, Canada, and Brazil. The basis of the United Kingdom Propagation Model is the prediction of received field strength at a specific location, considering external factors in the environment. In Japan, the field strength is predicted by computing free-space field strength and propagation loss. CRC-Predict is a propagation model developed in Canada where the predictions are based on detailed simulations of diffraction over the terrain and an estimate of additional local clutter attenuation. This model needs information such as antenna locations and heights, as well as a path profile extracted from a database, to perform the calculation. The method used in Brazil consists of the calculation of field strengths over diffraction paths described in ITU-R Recommendation P.526 (propagation by diffraction, associated with the Degout-Assis method): diffraction in obstacles considering the curvature of obstacles.

All the aforementioned studies are based on traditional statistical models and as we mentioned in the Introduction section, these type of models needs accurate and extensive information of the environment and the transmitter equipment.

In contrast, our paper is based on geographic coordinates and real measurements of the field strength, which implicates a reduction in the complexity of the algorithm and an increase in the accuracy of the prediction.

In the literature of coverage prediction based on geostatistics, ordinary kriging is presented as one of the main approaches. Kriging was introduced by Riihijärvi and Mähönen [12] and is a spatial interpolation technique inspired by geostatistics. The interpolation relies on the spatial correlation between the measured data to build a complete map over the geographic area of interest [14]. Kriging was applied in several studies to predict coverage [2], [3], highlighting that coverage map prediction based only on the interpolation of geo-located measurements gives a very good performance in terms of prediction accuracy. However, the main drawback is the computational complexity of the algorithm, which increases exponentially with the number of measurement points [14]. Ying *et al.* [15] presented results based on spatial interpolation techniques (kriging) of measurement data obtained in Seattle, WA. According to their results, empirical database administrator (DBA) models tend to overestimate received signal strengths by not explicitly accounting for local obstructions, while measurement-based kriging achieves consistently good performance. Specifically, they compared their results against ordinary kriging, the longley-rice model, and the Federal Communications Commission's Curves program. In our paper, we studied machine learning algorithms to improve the accuracy in the prediction compared with ordinary kriging, as well as, present an approach with low computational cost.

Recently, machine learning algorithms for analysis and data prediction have been proposed. A neural network method was applied in radio wave propagation prediction, where a radial basis function [16] is used to predict the propagation loss based on the variables of base antenna height, mobile antenna height, frequency, and the distance between transmitting and receiving antennas. However, this method is based on the distance from the transmitter to the receiver, where points at the same distance are estimated with similar values without taking into account the different real environments in each position. To tackle this issue, our paper is based on GPS coordinates instead of distances, which allows the regressor algorithm to learn about the specific environment of each training measurement.

Wolfle and Landstorfer [17] presented a model based on a feed-forward perceptron for the field strength prediction inside buildings. In this study, the main inputs parameters considered was the field strength at the centre point, the distance between transmitter and receiver, visibility, the shape of the room, the transmission loss of the walls, and the number of distance of walls. This method has the same drawback as the study in [16] since the authors considered the distance between transmitter and receiver as the metric for the prediction, and it needs a information in detail of the area in the analysis. Contrary to our paper, where we use GPS coordinates as the variables for our dataset.

Hou *et al.* [18] proposed a hybrid scheme based on the ray-tracing method and machine learning regression for radio wave propagation prediction. First, the authors predict the field strength in near field with RFR and then for the far field they convert the field strength to rays to apply the ray tracing algorithm. The training data come from simulations rather than measurements, and they used CST Studio Suite from CST Computer Simulation Technology GmbH to simulate them. The drawback of this method is the assumption of use simulation data to characterize a real environment, the authors used the triangular face as a basic unit to model buildings and terrains. Then, the variables for the machine learning algorithm are used to characterize the triangular face instead of variables to characterize the real environment. Furthermore, in the prediction of the far field, the authors used the distance from the observation point to the antenna as the metric for the prediction, which is the same problem as we discussed for Chang and Yang [16] and Wolfle and Landstorfer [17]. In our paper, we tackle these issues using real measurements with variables that depends on the specific location, which allows the regressor algorithm to learn the characteristic of the environment improving the accuracy in the prediction.

None of the research documents described previously have considered machine learning regression algorithms to assess the coverage prediction of DTT systems electric field intensity measurements. Consequently, to the best of our knowledge, our paper is the first work implementing machine learning regressor algorithms to predict the field strength measurements in DTT based on geographic coordinates. More specifically, the machine learning regression algorithms random forest regression, adaptive boosting (AdaBoost), and K-nearest neighbors (KNN) regression are studied to calculate the coverage of DTT. We made the tests and collected the field intensity measurements of eight channels that currently operate in the city of Quito, Ecuador. Furthermore, we compare and assess the performance of each model using 10-fold cross-validation with three error metrics: percentage of relative error, mean absolute error (MAE), and RMSE.

## III. Digital Terrestrial Television

Digital terrestrial television is a digital transmission system of open television signals where the content is encoded in the binary system. DTT is transmitted over radio frequencies through terrestrial space; it employs multiplex transmitters to allow reception of multiple channels in single frequency ranges known as subchannels. The DTT signal received is decoded via TV gateway, a digital set-top box or an integrated tuner included with television sets.

ISDB-Tb is one of the terrestrial digital television standards widely used in South and Central America. This standard has be employed in Ecuador, Peru, Argentina, Chile, Venezuela, Brasil, Costa Rica, Paraguay, Paraguay, Bolivia, Nicaragua, Uruguay, Honduras, and El Salvador.

### A. Coverage Areas and Reception Setup

The coverage area of a broadcasting station, or a group of broadcasting stations, is the area in which the wanted field strength is equal to or exceeds the minimum value of the field strength necessary to permit a desired reception quality

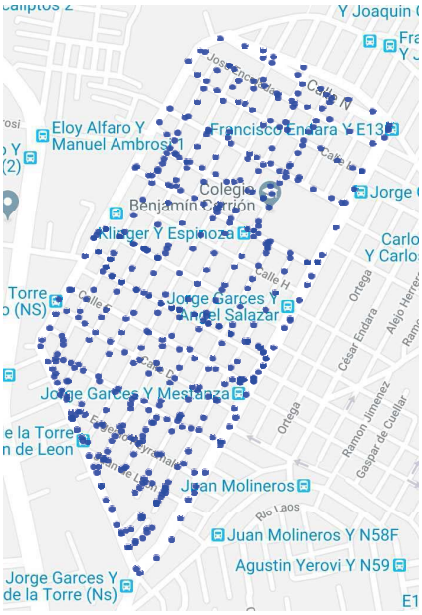


Fig. 1. Graphic area of interest with the location of the measuring points.

defined for specified reception conditions and for an envisaged percentage of the covered receiving locations [19].

Through several technical tests carried out in the city of Quito, the minimum field strength required to guarantee reception in the measurement area is 51 dBuV/m. According to the technical standards for the development of DTT in Ecuador, approved on April 30, 2015, the land area within the main coverage area authorized for a parent station and which does not comply with the level of field intensity of 51 dBuV/ m is considered a shadow zone [20].

In analog systems, the degradation of the signal as a function of the distance to the transmitter is smoother, and therefore, the signal can be received with distortions, whereas degradation of a DTT signal limits the coverage and causes the digital image to be completely lost.

We collected the measurements through the SACER Mobile monitoring vehicle [8]. This system is used by the Agency for Regulation and Control of Telecommunications (ARCOTEL), the main entity of telecommunications in Ecuador. The tests were made for eight DTT channels out of the nine DTT channels that currently operate in Quito, according to the information provided by ARCOTEL [21].

For each measured point, the exact coordinates were taken. These measurements correspond to the received signal strength, and are expressed as the decibel ratio referenced to microvolts per meter (dBuV/m). These observed measurements are stored in $S \times 1$ vector $E$.

$$\mathbf{E} = [\mathbf{E}(\mathbf{x}_1), \ldots, \mathbf{E}(\mathbf{x}_S)]^T \tag{1}$$

where $\mathbf{E}(\mathbf{x})$ is the received electric field strength at the location $\mathbf{x}$, $\mathbf{x} \in R^3$. $(.)^T$ denotes the transposition (by convention, the vectors are column-vectors). The measuring points shown in Fig. 1 are located in the city of Quito, this geographic area of interest has an area of 36,967.0758 $m^2$.

## IV. Proposed Methodology

### A. General Overview

We propose to separate the measurement points in groups based in their latitude, longitude and altitude such that the regression algorithm can achieve a better estimation of the area of interest. The technique of clustering, discussed in [22] appears as the best candidate scheme to separate in groups the measurement area. In this paper, we propose the K-Means algorithm which is a widely used technique for clustering points in Euclidean space and provides a rapid convergence.

K-Means is an unsupervised technique that permits to separate the samples into a number of disjoint groups, called clusters. A detailed description of the K-means algorithm is provided in [23, Sec. II]. In order to achieve a correct behavior of the K-means algorithm we transform the latitude and longitude coordinates (spherical coordinates) into a universal transverse mercator (UTM) coordinate system, which provides a projection of the GPS coordinates over a 2-dimensional Cartesian coordinate system for a correct operation of the K-means algorithm based on the Euclidean distance.

One important issue in clustering is to define the optimal number of clusters for our dataset. For this purpose, we apply the elbow method, which consists in to execute the K-means algorithm over the dataset for a range of K values, and then for each candidate number of clusters it evaluates the within cluster sum of squares (WCSS), which is the sum of the square distances of the samples to their closest cluster center. In a formal description, for a given set of clusters $C = (C_1, \ldots, C_n)$, whose centers are $(\phi_1, \ldots, \phi_n)$, the WCSS can be expressed as follows:

$$WCSS = \sum_{i=1}^{n} \sum_{x \in C_i} \|x - \phi_i\|^2 \tag{2}$$

Next, we plot the curve of the WCSS according to the number of clusters, where we achieve smaller WCSS values while we reduce the number of samples in each cluster. In this context, a value of zero in WCSS is reached when each cluster has only one sample. The objective is to have a low WCSS value with a reduced number of clusters. Then, the most appropriate value of the number of clusters can be consider as the location of a "knee" in the plot.

Once the number of clusters is selected, we separate the available measurements into disjointed clusters and train independent instances of the model for each cluster by applying machine learning regression algorithms (explained in Section V). The final objective is to obtain the predicted values of the electric field for any point in the interested area. In order to represent this objective, we build a grid of $1000 \times 1000$ points based on the geographic coordinates, so that the grid covers the entire sector of the dataset points. Finally, we assign each sample of the grid to their respective cluster based on the GPS coordinates and predict the electric field values by using the respectively trained instance of the regression model and represent this measurements in Google Maps using color mapping for the values of the electric field. Fig. 2 illustrates the aforementioned procedure with a block diagram.

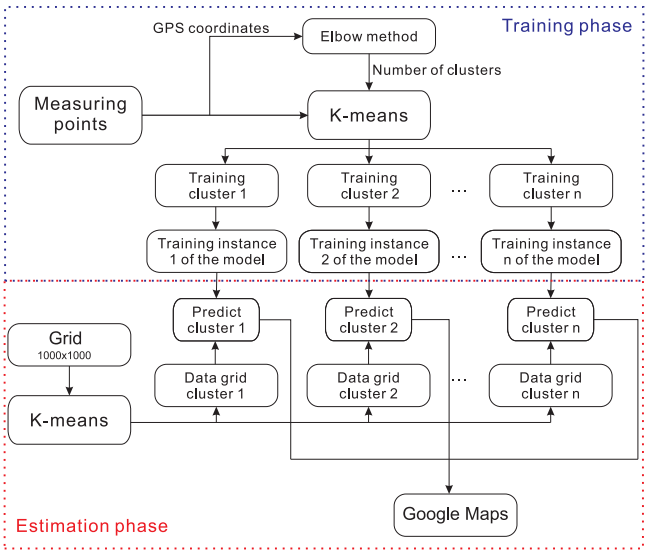


Fig. 2. Overall diagram of the proposed scheme for estimating DTT Coverage.

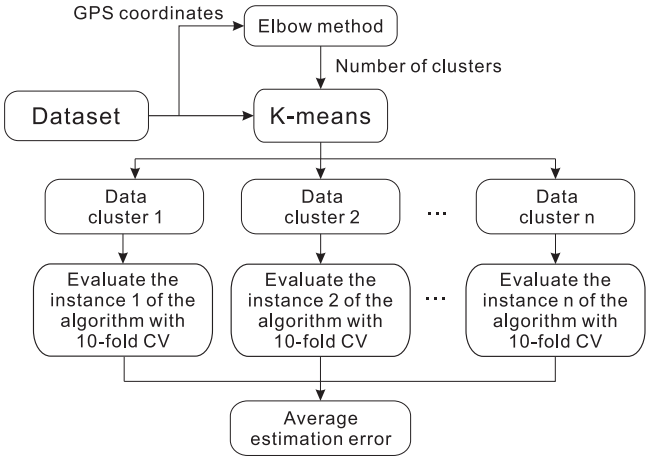


Fig. 3. Model evaluation method based on 10-fold cross-validation.

### B. Model Selection Procedure

In this paper we consider the Cross-validation (CV) technique [24] for the regression algorithm selection. Specifically, we apply the K-Fold cross-validation, which randomly splits the dataset into a K folds of approximately same size. Let us define the dataset as $D$, the K folds as $\{V_1, V_2, \ldots, V_K\}$ and the complement of each fold as $R_k = D \backslash V_k$, $k = \{1, \ldots, K\}$. Based on the aforementioned, we train the algorithm using the data of $R_k$ and evaluate the lost function by using the $V_k$, which guarantee that the evaluation of the model is considering unseen data. Then, we repeat the procedure K times with each of K folds used once time as evaluation data. In other words, if we consider L different algorithms represented by $A_i, i = \{1, \ldots, L\}$, the K-CV prediction error for the algorithm $A_i$ is the average of errors committed by the algorithm $A_i$ in their corresponding subsets $V_k$ as we can illustrate in the following equation:

$$CV(A_i) = \frac{1}{K} \sum_{k=1}^{K} \sum_{j \in V_k} \frac{\Psi(y_j, \hat{y}_j)}{|V_k|} \tag{3}$$

where $\Psi(y_j, \hat{y}_j)$ is the error in the estimation calculated by an error metric, $|V_k|$ is the number of samples in the fold $V_k$, $y_j$ is the true value of the sample $j$ in the fold $V_k$ and $\hat{y}_j$ is the predicted value of the sample $j$ by the algorithm $A_i$, which was trained using the data of $R_k$. The error metrics to be used in this article are described in Section VI.

With respect to the number of folds, we take into account the recommendation of Arlot and Celisse [25], which reported that when the goal of the model selection is the estimation, an optimal value of K is between 5 and 10. Furthermore, Rodriguez *et al.* [26] presented a study of the bias and variance in the error estimation by using K-fold cross-validation with the conclusion of applying repeated cross validation, which performs new random splits in each iteration. Based on the aforementioned recommendations, we propose a 10-Fold cross-validation scheme with 100 independent repetitions to fairly select the best regressor algorithm to predict the electric field values. First, we separate the dataset into clusters following the procedure detailed in the previous Section IV-A and for each cluster we perform 100 repetitions of the 10-Fold cross-validation procedure to finally calculate the average error of the $n$ clusters as the metric to select the best regressor algorithm. Fig. 3 explains the proposed method based on 10-fold cross-validation used to evaluate the proposed regressor algorithms.

### C. Analysis of the Number of Measurements

Taking field strength measurements in a big area implicates problems such as the time consuming, substantial human resources and measurement equipment, and some issues of private since we are not able to enter in privacy property to take the measurements of that geographic point. Then, we need a regressor algorithm which can deal with a small amount of measurement without compromises the accuracy in the prediction. With the above motivation, in this subsection, we present a methodology to select the best regressor algorithm that allows us to reduce the number of field strength measurements needed to properly characterize the area of in analysis.

In order to study the performance of the regressor algorithms vs the number of field strength measurements, we propose a methodology based on clustering using K-means. First, we divide the original measurements of the dataset into N disjointed groups taking into account the values of latitude and longitude (transformed into UTM coordinates) in such a way that the measurements which are close together are grouped in the same cluster. Second, we randomly select one measurement point in each cluster and we use these points for training and the remaining points for testing. Therefore, by selecting the number of clusters, we are able to select the number of measurements to training the regressor algorithm. We repeat this procedure for 100 independent random repetitions evaluating in each repetition the root mean square error, where we use the average of the error in the 100 random realizations to select the best algorithm. This methodology is inspired by the idea that we can define the number of measurement points by

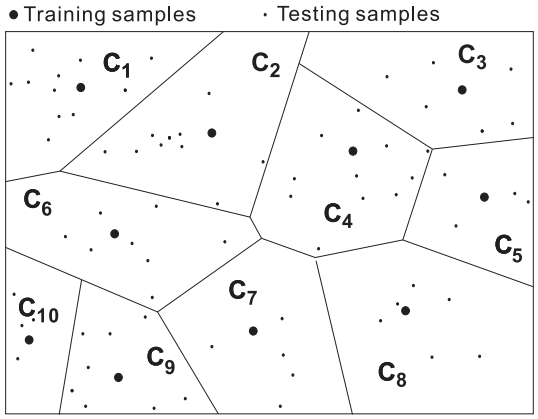


Fig. 4. Example of the proposed clustering scheme based on K-means to reduce the number of training points.

decreasing or increasing the distance between measurements in the area of interest.

Fig. 4 shows an example of one random realization using the proposed clustering scheme based on K-means, where we want to select only 10 points for training. Then, we divide the dataset into 10 clusters and randomly select one point in each cluster for training the algorithm and the remaining for evaluating the error.

## V. Machine Learning Regression Algorithms

In this section, we provide detailed descriptions of the machine learning regression algorithms used in this paper.

### A. Random Forest Regression

Random forest regression is an ensemble of multiple decision regression trees, each of them divides the original data into smaller subsets divided by split nodes. The algorithm starts dividing the dataset into n-tree bootstrap samples, which are randomly selected (with replacement), one for each regressor tree. Starting at the root node inside the regression tree, each split node performs a binary test that is applied to each subset to send it toward the left or the right child node. The test randomly selects a subset of features and looks for a value that carries the lowest mean square error (MSE) to group the training samples into clusters.

A sample point in the regression tree reaches a specific leaf node based on the values of its features, so the predicted value of the random tree regressor depends on the values of its features.

*1) Training Phase:* Given an original dataset $D = (\mathbf{x_i}, E_i)$ containing the vector $\mathbf{x_i} \in R^3$,$i \in \{1, 2, \ldots, q\}$, we consider $q$ to be the number of samples, and $\mathbf{x_i}$ is composed of the features of longitude, latitude and altitude. $E_i \in R$ is the real value of electric field strength.

Random forest divides original dataset $D$ into n-tree bootstraps, $D_s \subseteq D$, and each of them is entered the root node of the regression trees. Let $D_p$ be the subset of the bootstrap $D_s$ at split node $p$. A subgroup of features is randomly selected in each split node to develop the binary test, $t_{c,\tau}(\mathbf{x})$, where $\mathbf{x} \in D_p$. The binary test is defined as

$$t_{c,\tau}(\mathbf{x}) = \begin{cases} 1, & \text{if } x^c > \tau \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\tau$ is a threshold and $x^c$ indicates the value of sample $\mathbf{x}$ for feature $c \in \{1, 2, 3\}$. During the training process, in each split node, a pool of possible thresholds from the subset of randomly selected features is evaluated to select the test with the lowest MSE. The binary test divides the subset of training sample $D_p$ into two sets: $D_p^{right}$ containing those samples satisfying the test, and $D_p^{left}$ containing the rest of the samples, which do not satisfy the test [9]:

$$D_p^{right} = U\{\, (\mathbf{x}, E) \in D_p \mid t_{c,\tau}(\mathbf{x}) = 1 \,\} \tag{5}$$

$$D_p^{left} = U\{\, (\mathbf{x}, E) \in D_p \mid t_{c,\tau}(\mathbf{x}) = 0 \,\} \tag{6}$$

where $D_p^{right}$ and $D_p^{left}$ are assigned to the right child node and left child node, respectively. Each instance of the subset $D_p$ arriving at split node $p$ is evaluated for every possible test to minimize the MSE, which is defined as:

$$MSE(p) = \frac{1}{N_p^{right}} \Sigma_{i \in D_p^{right}} \left(E_i - E_p^{\hat{right}}\right)^2 + \frac{1}{N_p^{left}} \Sigma_{i \in D_p^{left}} \left(E_i - E_p^{\hat{left}}\right)^2 \tag{7}$$

where $N_p^{right}$ and $N_p^{left}$ are the number of training samples in $D_p^{right}$ and $D_p^{left}$, respectively. $E_i$ and $E_j$ are the true target values, and $E_p^{right}$ and $E_p^{left}$ are the predicted target values (sample mean) for the $D_p^{right}$ and $D_p^{left}$ subsets, respectively. The test with the lowest MSE is selected for the split, and it is held constant as the forest is growing. The regression trees are grown until a leaf node achieves a minimum number of training instances, or the minimum number of samples required to split an internal node is reached. Inside the random forest algorithm, we define a set of $K$ regression trees, $\{R_1(\mathbf{x}), R_2(\mathbf{x}), \ldots, R_k(\mathbf{x})\}$, where the output for a single sample point, $\mathbf{x}$, is defined as $\{\hat{E}_1 = R_1(\mathbf{x}), \hat{E}_2 = R_2(\mathbf{x}), \ldots, \hat{E}_k = R_k(\mathbf{x})\}$ . The final prediction of the random forest algorithm corresponds to the average of the k tree regressors by the following equation:

$$E_{predRF}(\mathbf{x}) = \frac{1}{k} \Sigma_{i=1}^{k} \hat{E}_i(\mathbf{x}). \tag{8}$$

*2) Testing Phase:* Via the trained regression forest, a test sample passes through each split node, where the stored binary test is applied to send the test instance to the right or left child until arriving at a leaf node. The predicted values for the test sample data in each regression tree are assigned according to the leaf node reached, and the final prediction of the random forest algorithm is calculated based on equation (8).

*3) Merits of Random Forest:* One of the main advantages of random forest is that it does not require of data preprocessing. In this context, random forest appears as a robust approach to outliers, which can be a problem in the dataset of field strength measurements. Note that the wireless channel is not uniform in the time, there can exist suddenly peaks of noise and interference that affects the real value of the measurement.

The aforementioned problems produce outlier measurements which can be easily managed by the random forest Algorithm. Inside the regressor trees, the outliers measurements are isolated in separate leaves in such a way they can not influence in the prediction of the other leaves.

Random forest is an algorithm easy to deploy since it is not too much sensitive to the specific parameters used, which makes random forest a versatile algorithm for any coverage prediction regardless of the type and the conditions of the area to analyze. The main adjustment to take into account is to select a large enough number of regressor trees to achieve the lowest error in the prediction.

The overfitting is a common problem in machine learning, which does not permit that our trained model generalize well to unseen data. Random forest avoids the overfitting if we have enough regressor trees in the forest. In general, the ensemble methods allow us to reduce the variance compared with an individual model, improving the accuracy in the prediction. Therefore, increasing the number of regressor trees in the forest will decrease the variance of the ensemble, reducing the possibility of overfitting.

### *B. AdaBoost Regressor*

AdaBoost is a boosting method to combine the prediction of other individual learners into a weighted sum representing the final output of the algorithm. In this paper, we select the regression tree as the individual learner to develop the AdaBoost regressor. The algorithm assigns weight $w_i$ to each training sample when learning the model, and this weight is used to indicate the relevance of a specific instance, and it is used to calculate the performance of the model. The method consists of $N$ iterations, each of which is used to train a different regressor tree. After each iteration, the weight assigned to each sample is recomputed and greater weight is given to those instances that have a higher loss in the prediction. On the other hand, the weight is reduced with smaller losses, decreasing probability $p_i$ (see Algorithm 1). Intuitively, in each iteration, AdaBoost varies weight $w_i$ to focus on the training samples with a higher error in the predicted value, and thus, each subsequent regressor tree has a redefined dataset to train on. In this paper, the AdaBoost.R2 [10] algorithm is adopted and further modified for DTT coverage. The proposed AdaBoost.R2 can be summarized in Algorithm 1. The algorithm expresses each sample error in relation to the largest error. Furthermore, preliminary experiments showed that AdaBoost with a linear loss function provides better performance than using square and exponential loss functions.

In the testing phase, test sample $\mathbf{x}$ is passed through each $k$ regressor tree, $R_k(\mathbf{x})$, and the prediction of the AdaBoost algorithm, $R_f(\mathbf{x})$, is the weighted median of $R_k(\mathbf{x})$.

**Algorithm 1** AdaBoost.R2 in DTT Coverage

INPUTS The original dataset $D = \{(\mathbf{x_i}, E_i)\}$ of size $q$ where $\mathbf{x_i} \in R^3$ (longitude, latitude, and altitude), the maximum number of iterations $N$, and an initial weight vector $\mathbf{w}^1$ in the first iteration, which can be $w_i^1 = 1$ for the samples $i = 1, 2, .., q$.

a) Initialize: k $= 0$ and $w_i^1 = 1, \forall i \in D$.

b) **Repeat**:

1: $k \leftarrow k + 1$;

2: Train the $k$ regressor tree with dataset $D$ using weight vector $w^k$, and get the model, $R_k(\mathbf{x})$.

3: Calculate loss $L_i^k$ (a linear function) for each training sample for model $R_k$ for which the algorithm uses the largest error, $M_k$, defined as: $M_k = sup \mid R_k(\mathbf{x_i}) - E_i \mid, i = 1, 2, ..., q$ then $L_i^k = \frac{|R_k(\mathbf{x_i}) - E_i|}{M_k}$

4: Calculate the average loss of $R_k(\mathbf{x})$ using the probability $p_i^k$, defined as: $p_i^k = \frac{w_i^t}{\sum_{i=1}^q w_i^t}$ Then $\bar{L_k} = \sum_{i=1}^q L_i^k p_i^k$

5: Evaluate $\beta_k = \frac{\bar{L_k}}{1-\bar{L_k}}$.

6: Update the weights: $w_i^{t+1} = w_i^t \beta_k^{1-L_i^k}$.

c) **Until**: average loss $\bar{L_k}$ greater than or equal to 0.5 or $k = N$.

The output of AdaBoost $R_f(\mathbf{x})$ is the weighted median of $R_k(\mathbf{x})$ for the $N$ iterations, using $\ln(1/\beta_k)$ as the weight for regressor tree $R_k(\mathbf{x})$.

## VI. Numerical Results

In this section, we evaluate the performance of the proposed machine learning regression algorithms by running the simulation in an urban scenario of 36,967.0758 $m^2$ located in the city of Quito, Ecuador, based on real field strength measurements related to channel 26 (545.14 MHz), channel 30 (569.14 MHz), channel 32 (581.14 MHz), channel 34 (593.14 MHz), channel 36 (605.14 MHz), channel 41 (635.14 MHz), channel 43 (647.14 MHz), and channel 45 (659.14 MHz). Furthermore, we compare our results with ordinary kriging, a method of interpolation applied to cellular coverage prediction [14]. For the ordinary kriging algorithm we use the module previously developed in Python called PyKrige [27]. In addition to ordinary kriging, we compare the results with other algorithms for regression such as support vector regression with radial basis function (RBF) kernel [28], lasso regression [29] and neural network regression. In the case of the neural network algorithm, we select the number of neurons based on the recommendation of Ke and Liu [30], where the number of neurons in the hidden layers are setting with the following equation

$$N = \frac{N_{in} + \sqrt{N_p}}{L} \tag{9}$$

where $N_{in}$ is the number of neurons in the input layer, $N_p$ is the total number of samples, and $L$ is the number of hidden layers. Applying the equation (9) to our dataset, we get 3 hidden layers with 8 neurons in each one. A deeply study as well as variants of the traditional neural network are beyond the scope of this article.

Following the procedure described in Section IV-A, we first apply the elbow method to select the number of clusters to execute the K-means algorithm. Fig. 5 shows the WCSS value for each possible value of the number of clusters for all the channels in the dataset. By applying the elbow method, we locate the bend or "knee" of the curve at 3 clusters, which will be

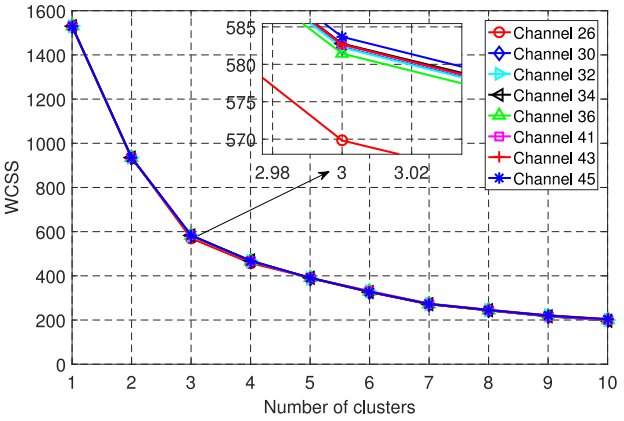


Fig. 5. Number of clusters versus WCSS.

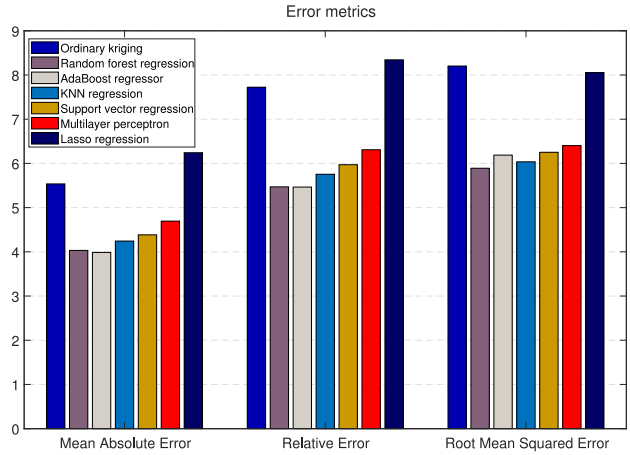


Fig. 6. Performance comparison among machine learning algorithms.

the number of clusters to be used in the rest of the simulation section. Note that by applying the elbow method for all the channels, we achieve the same result for the number of clusters. Next, we divide the measurements of the data set in 3 (value obtained with the elbow method) groups by using K-means and perform K-fold cross-validation with K=10 following the procedure detailed in Section IV-B. We selected three error metrics, which are explained below, to evaluate the estimation error in the considered algorithms.

- Relative error indicates the accuracy of the measurement, and is defined as absolute error $(\hat{y} - y)$ divided by the exact value, i.e., $(\hat{y} - y)/y$; in our case, y means the real electric field strength measurement, and $\hat{y}$ is the predicted value of electric field strength.
- Mean absolute error indicates the average error of the predicted values by calculating the absolute values of the differences between the predicted measure of electric field strength, $\hat{y_i}$, and the corresponding true value, $y_i$. This metric is defined by:

$$MAE = \frac{1}{n_{samples}} \Sigma_{i=0}^{n} \mid y_i - \hat{y_i} \mid \tag{10}$$

- Root mean square error measures the error rate through the square root of the average of squared differences between $\hat{y}$, the predicted value of electric field strength, and the actual observation given by the value $y$. This metric is defined by:

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{n} \left(y_i - \hat{y_i}\right)^2}{n}} \tag{11}$$

where $n$ is the number of samples. RMS tends to grow larger than MAE with $n^{(1/2)}$ since its lower limit is fixed at the MAE, and its upper limit increases with $n^{(1/2)}$.

### A. Analysis With K-Fold Cross Validation

In this subsection, we present the numerical results by applying the procedure described in Section IV-B. In Fig. 6, we compare regression performance using the error metrics mentioned above among ordinary kriging, random forest regressor, AdaBoost regressor, KNN regression, support vector machine regression, multilayer perceptron and lasso regression. The results show that random forest regression and AdaBoost regressor have the lower error measurements, while being a slight difference between them, in terms of, mean absolute error and relative error. However, if we compare both algorithms, in terms of, root mean squared error, the error difference is noticeable as AdaBoost regressor presents higher error measurement than random forest regressor. In this sense, we can appreciate that the performance of random forest regression outperforms the other machine learning regression algorithms and ordinary kriging. Furthermore, it is remarkable that the lasso regression algorithm exhibits the worst percentage of error measurements, followed by ordinary kriging.

Regarding the average evaluation of mean absolute error illustrated in Table I, the best response is given by the AdaBoost algorithm, with a mean value of 3.987, followed by a slight difference from random forest regression and KNN regression, with means of 4.032 and 4.244, respectively. We can see that the random forest regression, AdaBoost regression and KNN regression outperform the ordinary kriging method (with means of 5.537) and other regression algorithms, such as support vector regression, multilayer perceptron and lasso regression. The individual analysis for each channel shown in Table I supports the global analysis and reflects that the lowest MAE is given by AdaBoost, with 3.045 on Channel 32, slightly followed by random forest with 3.126 in the same channel, while the highest MAE is given by lasso regression, with 7.905 on Channel 30. Note that by using this error metric, all the individual error differences are weighted equally in the average.

While MAE gives the same weight to all errors, the RMSE penalizes variance, as it gives greater weight to errors with larger absolute values than errors with smaller absolute values. Random forest regression depicts the lowest mean RMSE over all the analyzed channels, as we can see in Table II, followed very closely by the AdaBoost regressor and KNN regression; however, lasso regression and ordinary kriging achieve the worst RMSE in all the channels. Analyzing the RMSE metric for each channel, we can see that the lowest error in all the

TABLE I
MEAN ABSOLUTE ERROR

| Ch | OK | RFR | AB | KNN | SVR | MP | LR |
|---|---|---|---|---|---|---|---|
| 26 | 7.104 | 5.232 | 5.454 | 5.299 | 5.5358 | 5.728 | 7.021 |
| 30 | 6.993 | 5.306 | 5.148 | 5.267 | 5.400 | 5.965 | 7.905 |
| 32 | 6.757 | 4.370 | 4.187 | 4.636 | 4.698 | 5.096 | 7.036 |
| 34 | 6.225 | 4.464 | 4.391 | 4.706 | 4.803 | 5.388 | 7.263 |
| 36 | 6.618 | 4.514 | 4.491 | 4.816 | 5.025 | 5.472 | 7.333 |
| 41 | 4.059 | 3.126 | 3.045 | 3.605 | 3.398 | 4.037 | 5.839 |
| 43 | 6.106 | 4.666 | 4.616 | 4.991 | 5.250 | 5.470 | 7.050 |
| 45 | 5.970 | 4.610 | 4.550 | 4.874 | 5.354 | 5.095 | 6.739 |
| **Mean** | 5.537 | 4.032 | 3.987 | 4.244 | 4.385 | 4.695 | 6.243 |

OK: Ordinary kriging, RFR: Random forest regression, AB: AdaBoost regression, KNN: K-nearest neighbor regression, SVR: Support vector regression, MP: Multilayer perceptron, LR: Lasso regression.

TABLE II
ROOT MEAN SQUARE ERROR

| Ch | OK | RFR | AB | KNN | SVR | MP | LR |
|---|---|---|---|---|---|---|---|
| 26 | 9.501 | 7.141 | 7.739 | 7.171 | 7.399 | 7.399 | 9.070 |
| 30 | 10.308 | 7.888 | 8.284 | 7.782 | 7.921 | 8.357 | 10.390 |
| 32 | 9.700 | 6.402 | 6.535 | 6.548 | 6.603 | 6.903 | 8.882 |
| 34 | 9.024 | 6.364 | 6.588 | 6.499 | 6.779 | 7.280 | 9.092 |
| 36 | 9.520 | 6.554 | 6.913 | 6.664 | 6.936 | 7.236 | 9.1232 |
| 41 | 8.601 | 6.024 | 6.300 | 6.429 | 6.292 | 6.685 | 8.519 |
| 43 | 8.652 | 6.359 | 6.737 | 6.652 | 7.011 | 7.102 | 8.852 |
| 45 | 8.520 | 6.285 | 6.595 | 6.587 | 7.326 | 6.657 | 8.562 |
| **Mean** | 8.203 | 5.891 | 6.188 | 6.037 | 6.252 | 6.402 | 8.055 |

OK: Ordinary kriging, RFR: Random forest regression, AB: AdaBoost regression, KNN: K-nearest neighbor regression, SVR: Support vector regression, MP: Multilayer perceptron, LR: Lasso regression.

channels is attained by the random forest regression algorithm, with 6.024 on Channel 41, and the highest error is attained by lasso regression, with 10.390 on Channel 30, following by ordinary kriging, with 10.308 on the same Channel 30.

Additionally, we assessed a suitable number of regressor trees used by random forest regression and the AdaBoost regressor algorithm because of the importance of improving the yield from the model. For this, we chose Channel 26 to analyze the training time and percentage of relative error while increasing the number of regressor trees. The training times were measured on a MacBook Pro 2.8 GHz Intel Core i7 with 16 GB of main memory. In that way, we appreciate that using more regressor trees means training time increases, as shown in Fig. 7. By contrast, Fig. 8 shows that the percentage of relative error, from 150 regressor trees, does not vary markedly when we utilize more regressor trees for random forest regression and AdaBoost regression. Hence, for our purposes, we use 200 regressor trees in random forest regression and in the AdaBoost regression.

### *B. Analysis Varying the Size of the Dataset*

We change the number of field strength measurements in order to evaluate the performance of each algorithm with fewer training points following the procedure described in Section IV-C. It can be seen from Fig. 9. that with 50 training points, the random forest regression is the algorithm with

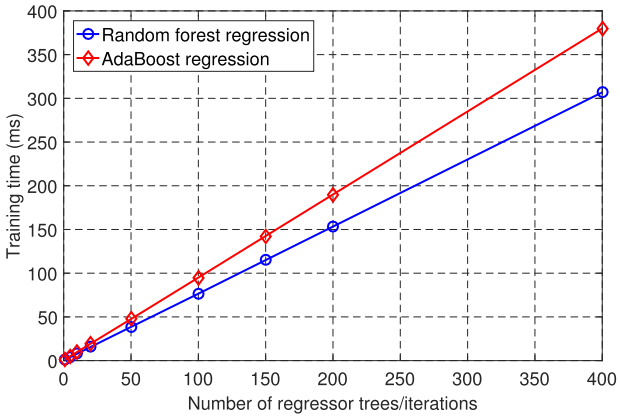


Fig. 7. Number of regressor trees vs. training time.

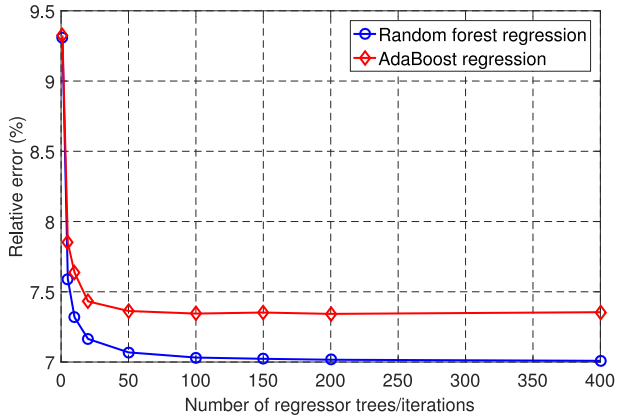


Fig. 8. Number of regressor trees vs. percentage of relative error.

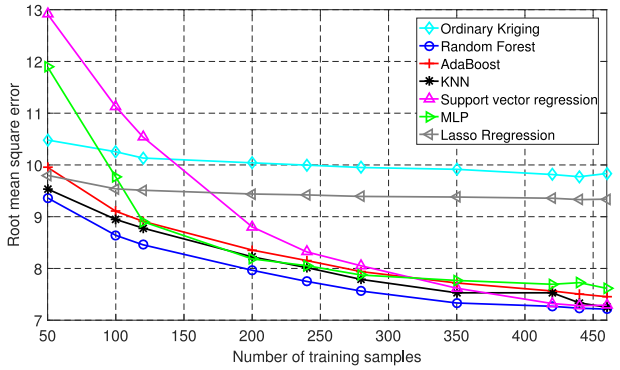


Fig. 9. Number of training points vs. root mean squared error.

the lowest root mean squared error, while the support vector regression has the highest error with 12.91, following by multilayer perceptron and ordinary kriging with 11.89 and 10.48, respectively. Besides, when we increase the number of training points, the mean squared error in the majority of the algorithms tends to decrease. However, even increasing the training points, ordinary kriging and lasso regression decrease the error slightly compared with the other algorithms. On the other hand, we can appreciate that the random forest regression achieves the lowest error as we increase the training points, which indicates that the random forest regression has a better performance than the other algorithms aforementioned for

TABLE III
RELATIVE ERROR WITH AND WITHOUT CLUSTERING

| Channel | Ordinary kriging | RFR | AdaBoost regressor | KNN regression |
|---|---|---|---|---|
| K-means | 9.53 | 7.02 | 7.34 | 7.16 |
| Whitout clustering | 10.12 | 7.03 | 7.13 | 7.19 |

our purpose, which is to make predictions of digital terrestrial television coverage.

Next, we compare the performance of clustering versus only one cluster (without using the K-means algorithm) by using the ordinary kriging and the machine learning regressor algorithms. In Table III we present the relative error in the channel 26 (channel with the lowest relative errors) by applying K-means with 3 clusters compared with the approach without clustering. We can observe that in both cases the relative error is similar, but using K-means we can obtain the lower relative error in ordinary kriging, random forest and KNN. In the case of AdaBoost, K-means causes an increase in the relative error since AdaBoost works better if more samples are available during the training phase as we analyzed in Fig. 9.

### C. Graphical Results of Coverage Map Prediction

In this subsection, we present a grid of $1000 \times 1000$ points based on geographic coordinates covering the entire area of the dataset with the field strength values predicted by the trained regressor algorithm following the procedure described in Section IV-A. For the representation in Google Maps, we use the Python module Bokeh [31], which provides models for displaying geographic points in Bokeh plots. We chose the channels 32 and 41 to present the DTT coverage prediction over Google Maps in the considered area of interest.

Fig. 10 illustrates the DTT coverage prediction for Channel 36, where the color bar chart located at the left of each graph indicates the level of the electric field in terms of the decibel ratio referenced to microvolts per meter (dBuV/m). We show the DTT coverage prediction by using ordinary kriging, commonly used in the literature, and by using random forest regression, AdaBoost regression and KNN regression, which achieved the lower error in the prediction as we described in the previous Section VI-A. We can observe areas with a coverage threshold that borders the acceptable limit of 51 dBuV/m. This fact indicates the possibility that the DTT signal reception quality presents shadow areas and degradation. Furthermore, we can notice that the coverage predictions using machine learning regression algorithms follow the same pattern, which is in concordance with the prediction of ordinary kriging.

Fig. 11 illustrates the DTT coverage prediction for Channel 41 by using ordinary kriging and random forest regression. The predictions over the channel 41 show more fluctuations in the signal level than Fig. 10, and in this graphic, we can see that even when the signal level changes remarkably, machine learning regression algorithms work effectively. The coverage pattern predicted by random forest is similar to the patter in ordinary kriging, however, there exist certain

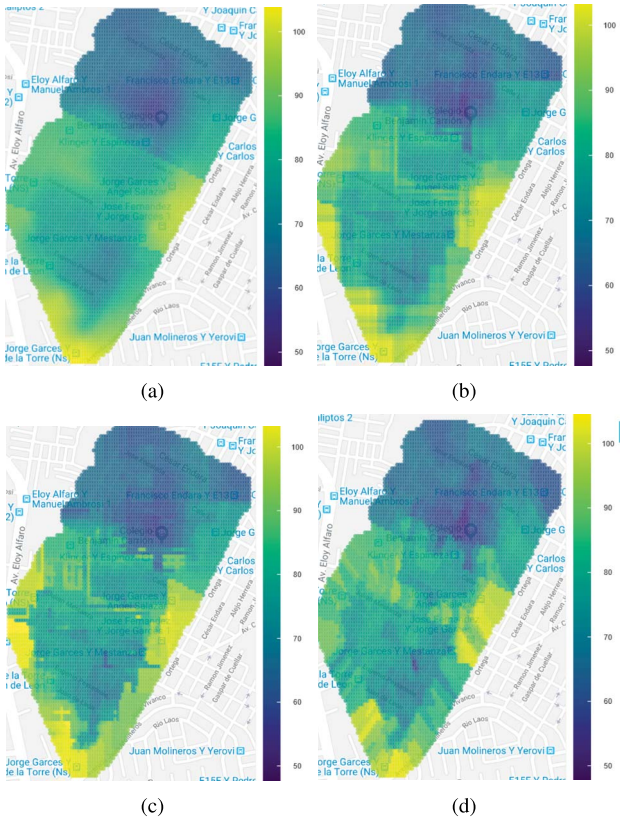


Fig. 10. Channel 36 - DTT coverage map by (a) ordinary kriging, (b) random forest regression, (c) AdaBoost regressor, and (d) KNN regression.

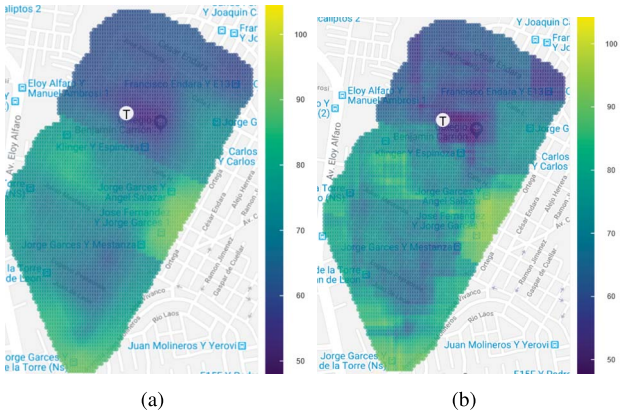


Fig. 11. Channel 41 - DTT coverage map by (a) ordinary kriging and (b) random forest regression.

areas where the prediction of ordinary kriging can not detect a shadow area. For instance, we can refer to sample point (T), which is predicted as a value of 49.377 (dBuV/m) in random forest (presence of shadow area) and as a value of 55.904 (dBuV/m) in ordinary kriging (no presence of shadow area).

## VII. CONCLUSION

In this paper, a new method of calculating DTT coverage prediction is proposed based on clustering and a machine regression algorithm. We use real field strength measurements and perform clustering with machine learning regression algorithms to obtain high accuracy in the prediction and low

computational load. It is noteworthy that the random forest regressor has the best performance, in comparison with Adaboost regression, K-nearest neighbors regression, and ordinary kriging. From the proposed method and the coverage views on Google Maps, it is shown that we can identify shadow areas and degradation in digital signal quality. Subsequently, the proposed scheme would be very useful for planning a digital terrestrial television system, especially in those countries that still have not carried out the analog switch-off process.

## REFERENCES


[1] "Coverage prediction methods and planning software for digital terrestrial television broadcasting (DTTB) networks," Int. Telecommun. Union, Rep. ITU-R BT.2137, Geneva, Switzerland, 2008. [Online]. Available: https://www.itu.int/pub/R-REP-BT.2137

[2] "Method for point-to-area predictions for terrestrial services in the frequency range 30 to 3000 MHz," Int. Telecommun. Union, Geneva, Switzerland, ITU Recommendation P.1546-5, 2013. [Online]. Available: https://www.itu.int/rec/R-REC-P.1546-5-201309-I/en

[3] "A path-specific propagation prediction method for point-to-area terrestrial services in the VHF and UHF bands," Int. Telecommun. Union, Geneva, Switzerland, ITU Recommendation P.1812-4, 2015. [Online]. Available: https://www.itu.int/rec/R-REC-P.1812-4-201507-I/en

[4] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Technol.*, vol. VT-29, no. 3, pp. 317–325, Aug. 1980.

[5] A. Achtzehn, J. Riihijärvi, G. Martínez, M. Petrova, and P. Möhönen, "Improving coverage prediction for primary multi-transmitter networks operating in the TV whitespaces," in *Proc. 9th Annu. IEEE Commun. Soc. Conf. Sensor*, Seoul, South Korea, 2012, pp. 623–631.

[6] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. Sousa, "Ensemble approaches for regression: A survey," *ACM Comput. Surveys*, vol. 45, no. 1, pp. 1–40, 2012.

[7] F. S. Silva, L. J. Matos, F. A. C. Peres, and G. L. Siqueira, "Coverage prediction models fitted to the signal measurements of digital TV in Brazilian cities," in *Proc. SBMO/IEEE MTT-S Int. Microw. Optoelectron. Conf. (IMOC)*, 2013, pp. 1–5.

[8] M. Fillián, *Sistema Automatico Para el Control del Espectro Radioelectrico (SACER)*, Superintendecia de Telecomunicaciones, Quito, Ecuador, 2013, pp. 18–26.

[9] H. Phan, M. Maaß, R. Mazu, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 1, pp. 20–31, Jan. 2015.

[10] H. Drucker, "Improving regressors using boosting techniques," in *Proc. 14th Int. Conf. Mach. Learn.*, 1997, pp. 107–115.

[11] X. D. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[12] J. Riihijärvi and P. Mähönen, "Estimating wireless network properties with spatial statistics and models," in *Proc. IEEE Conf. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Paderborn, Germany, 2012, pp. 331–336.

[13] N. A. P. García *et al.*, "Improved ITU-R model for digital terrestrial television propagation path loss prediction," *Electron. Lett.*, vol. 53, no. 13, pp. 832–834, Jun. 2017.

[14] H. Braham, S. Jemaa, G. Fort, E. Moulines, and B. Sayrac, "Spatial prediction under location uncertainty in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7633–7643, Nov. 2016.

[15] X. Ying, C. W. Kim, and S. Roy, "Revisiting TV coverage estimation with measurement based statistical interpolation," in *Proc. 7th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Bengaluru, India, 2015, pp. 1–8.

[16] P.-R. Chang and W.-H. Yang, "Environment-adaptation mobile radio propagation prediction using radial basis function neural networks," *IEEE Trans. Veh. Technol.*, vol. 46, no. 1, pp. 155–160, Feb. 1997.

[17] G. Wolfle and F. M. Landstorfer, "A recursive model for the field strength prediction with neural networks," in *Proc. 10th Int. Conf. Antennas Propagat.*, vol. 2, 1997, pp. 174–177.

[18] W. Hou, D. Shi, Y. Gao, and C. Yao, "A new method for radio wave propagation prediction based on finite integral method and machine learning," in *Proc. IEEE 5th Int. Symp. Electromagn. Compat. (EMC-Beijing)*, Oct. 2017, pp. 1–4.

[19] *Handbook on Digital Terrestrial Television Broadcasting Networks and Systems Implementation*, Int. Telecommun. Union, Geneva, Switzerland, 2016. [Online]. Available: http://handle.itu.int/11.1002/pub/80ec8c3c-en

[20] O. Myriam, *Norma Técnica de Radiodifusíon de Televisíon Digital Terrestre*, ARCOTEL, Quito, Ecuador, Sep. 2015. [Online]. Available: http://www.arcotel.gob.ec/wp-content/uploads/2016/06/norma-television-digital-terrestre.pdf

[21] *Estaciones Concesionadas de Televisíon Abierta y Digital Terrestre*, Agencia de Regulación y Control de las Telecomun., Quito, Ecuador, Mar. 2018. [Online]. Available: http://www.arcotel.gob.ec/radiodifusion-sonora-y-television-abierta-2/

[22] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.

[23] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Security Informat.*, Jinggangshan, China, Apr. 2010, pp. 63–67.

[24] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. IJCAI*, vol. 14, 1995, pp. 1137–1145.

[25] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Stat. Surveys*, vol. 4, pp. 40–79, Mar. 2010.

[26] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 569–575, Mar. 2010.

[27] B. Murphy. *Kriging Toolkit for Python, Version 1.3.2*. Accessed: Oct. 8, 2017. [Online]. Available: https://pypi.org/project/PyKrige/

[28] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[29] J. Freidman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1–22, Jan. 2010.

[30] J. Ke and X. Liu, "Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction," in *Proc. IEEE Pac.–Asia Workshop Comput. Intell. Ind. Appl.*, Wuhan, China, Dec. 2008, pp. 828–832.

[31] *Bokeh, Version 0.12.15*. Accessed: Mar. 2018. [Online]. Available: https://bokeh.pydata.org/en/latest/docs/reference/models/map_plots.html


**Carla E. Garcia Moreta** received the B.S. degree in electronics and telecommunications engineering from National Polytechnic School, Quito, in 2016. She is currently a graduate research student with the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. Her main research interests are machine learning, MIMO communications, NOMA, and optimizations.

**Mario R. Camana Acosta** received the B.E. degree in electronics and telecommunications engineering from National Polytechnic School, Quito, in 2016. He is currently a graduate research student with the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. His research interests include machine learning, optimizations, and MIMO communications.

**Insoo Koo** received the B.E. degree from Konkuk University, Seoul, South Korea, in 1996, and the M.Sc. and Ph.D. degrees from the Gwangju Institute of Science and Technology, Gwangju, South Korea, in 1998 and 2002, respectively, where he was with the Ultrafast Fiber-Optic Networks Research Center, as a Research Professor, from 2002 to 2004. In 2003, he was a Visiting Scholar with the Royal Institute of Science and Technology, Stockholm, Sweden. In 2005, he joined the University of Ulsan, Ulsan, South Korea, where he is currently a Full Professor. His current research interests include spectrum sensing issues for CRNs, channel and power allocation for cognitive radios and military networks, SWIPT MIMO issues for CRs, MAC and routing protocol design for UW-ASNs, and relay selection issues in CCRNs.