IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 May 2023, accepted 11 June 2023, date of publication 16 June 2023, date of current version 3 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3287103

RESEARCH ARTICLE

Extremely Randomized Trees Regressor Scheme for Mobile Network Coverage Prediction and REM Construction

CARLA E. GARCÍA, (Graduate Student Member, IEEE), AND INSOO KOO
Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 680-749, South Korea

Corresponding author: Insoo Koo (iskoo@ulsan.ac.kr)

This work was supported in part by the Information and Communication Technologies (ICT) Research and Development Innovation voucher conducted, in 2022, with the support of the Institute of Information and Communications Technology Planning and Evaluation (IITP), funded by the Government's Ministry of Science and ICT, development of wired/wireless communication technology for installation safety communication network for workers in shaded areas of building ships, under Grant 2022-0-00805; and in part by the "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) under Grant 2021RIS-003.

ABSTRACT In mobile communications network planning (and designing any radio system), coverage prediction helps network operators optimize cellular networks to improve customer experience. Accordingly, several path-loss models have been proposed that depend on many conditions, such as suitable selection of the terrain for each model, the height of the receiver and transmitter above ground and the distance between them, and the presence of obstacles. This may increase the prediction error between actual and estimated values, which change according to the propagation model selected. To overcome these problems, we propose a novel approach to mobile coverage prediction based on an extremely randomized trees regressor (ERTR) algorithm. In addition, we construct a radio environment map (REM) over a Google Earth digital map to improve visualization of the results and to easily detect coverage holes and traffic hotspots. For this purpose, we utilize a dataset with real measurements collected from Victoria Island and Ikoyi in Lagos, Nigeria. For performance evaluation, we use k-fold cross-validation based on four error metrics: relative error, root mean squared error, mean absolute error, and R^2 score. The proposed ERTR scheme achieves the best performance in terms of accuracy and computational load in predicting the reference signal received power and the received signal strength indicator value. We prove this with extensive simulation analysis and by comparing the error metrics of the proposed ERTR approach with an existing method widely used to perform coverage prediction, called ordinary kriging. We also compared seven machine learning regression algorithms, namely, random forest, a bagging regressor, support vector regression, k-nearest neighbors, a deep neural network, Gaussian process regression, and the decision tree.

INDEX TERMS Coverage prediction, radio environment map (REM), extremely randomized trees, machine learning, reference signal received power.

I. INTRODUCTION

Currently, mobile communications (MC) provides a flexible infrastructure subject to the challenges of increasing demand for mobile data. For instance, fifth-generation (5G) technology is capable of accessing and sharing information in scenarios with high data rates and extremely low latency, in which the transmission environment effects increase the vulnerabilities of the signal itself, especially in 5G millimeter wave networks [1], [2]. As a result, more antennas must be installed closer to user nodes, exceeding the number of antennas needed [1], [3], [4], [5]. Accordingly, coverage prediction plays a key role in the resource management of MC, which entails better network planning, design, and implementation, plus optimization improvements. In addition, a radio environment map (REM) is considered by regulatory agencies as a

The associate editor coordinating the review of this manuscript and approving it for publication was Aasia Khanum.

helpful tool for informed decision-making, and by network operators to ease coverage hole detection and traffic hotspots.

Overall, several path-loss models have been proposed that depend on many conditions, such as suitable terrain selection for each model, the height of the receiver and transmitter above the ground and the distance between them, the presence of obstacles, and so on [6]. These factors may increase the prediction error between actual and estimated values, which varies depending on the propagation model selected [7]. For instance, in [8], the authors proposed a propagation model called COST-231-Walfisch -Ikegami that utilized a geographic information system tool for field strength prediction in cellular mobile communications. Although the authors highlighted the benefits of geographic information system tools to deal with spatial databases analysis, and generating essential spatial parameters for field strength prediction, the proposed COST-231-Walfisch -Ikegami model is mainly useful for isotropic antennas. In the literature on REM construction, ordinary kriging (OK) has been widely used as a spatial interpolation technique based on geostatistics [1], [2], [9], [10]. OK estimates unknown data points according to the spatial correlation between measured data and the relative positional relationships between all sample points [2]. For instance, in [1], the authors constructed a REM for an indoor propagation environment based on interpolation methods. The results showed that OK outperformed baseline schemes such as inverse distance weight [3] and k-nearest neighbors (KNN) in terms of root mean squared error (RMSE) and correlation coefficients. Although OK can achieve high accuracy, its main disadvantage is computational cost, which rises exponentially with the number of measurement points [3], [10]. Moreover, a heuristic-based approach has been proposed in [11] for coverage prediction for indoor environments based on the indoor dominant path model. Since it still relies on a path loss model, its extension for outdoor scenarios can be difficult to implement.

Although the MC wireless transmission environment is complex, conventional mobile network planning techniques based on propagation models are inflexible and are subject to specifications such as antenna height, frequency, and environmental conditions [6]. Therefore, in recent breakthroughs, machine learning (ML)-based schemes have emerged as innovative prediction techniques capable of dealing with mobile network operational complexities, and they can provide high accuracy [12], [13]. For instance, in [14], the authors made path loss predictions in an urban environment in Beijing, China, by applying an artificial neural network (ANN), support vector regression (SVR), and random forest (RF) models. The performance evaluated in terms of RMSE achieved results between 4 dB and 5 dB. Similarly, in [15], the authors utilized SVR and RF to predict the path loss of a 5G network in Lisbon, Portugal. RMSE was evaluated using 10-fold cross-validation, and the obtained results varied between 6 dB and 7 dB. Moreover, an ANN and Gaussian process regression (GPR) were applied in suburban environments in South Korea, giving RMSE values between 8 dB and 9dB [16]. On the other hand, ML models based on an ANN, RF, and SVR were applied in rural environments in Greece to make path-loss-based predictions for an RMSE average of 4.2 dB [17]. In [18], the authors compared the coverage prediction performance between ANN schemes, multi-layer perceptron (MLP) with two hidden layers, and KNN for cellular networks based on the signal to interference ratio metric. The results showed that ANN with Gaussian kernels and the MLP technique obtained the best performance. To the best of our knowledge, none of the research described above considered an extremely randomized trees regressor (ERTR) for coverage prediction or REM construction.

In recent research, reference signal received power (RSRP) was considered the target label in MC since the RSRP parameter represents the network signal level at the user node location [19] in fourth-generation (4G) Long Term Evolution (LTE) and 5G New Radio (NR) networks. In [20], the authors applied an RF model to predict RSRP in multiple environments located in China. The results obtained 6.11 dB of RMSE by applying 10-fold cross-validation. Meanwhile, in [21], the authors analyzed several ML models as linear regression (LR), the ANN, SVR, GPR, regression trees (RT), and RF. The authors stated that according to 10-fold cross-validation, GPR achieved the best performance at 5.64 dB, followed by RF at 6.18 dB. For this purpose, the authors used 18,048 samples from 4G LTE collected in Putrajaya, Malaysia.

Motivated by the benefits provided by the ensemble learning techniques to obtain high accuracy for indoor [22] and outdoor [7], [21] environments regardless of propagation models. In this paper, we propose a novel ML regression approach based on an ensemble learning algorithm (namely ERTR) to perform coverage prediction and design the REM for MC. Our goal is to predict RSRP and the received signal strength indicator (RSSI) values in an urban dense area located on Victoria Island, Lagos, Nigeria [23]. In addition, we utilized 5-fold cross-validation to evaluate the performance of the proposed ERTR approach, the baseline ML models, and OK, by comparing different error metrics. This paper opens the door to constructing ERTR-based REM designs for MC environments that can be extended for coverage analysis in various outdoor and indoor propagation scenarios. It is worth highlighting that this is the first work that investigates ERTR for coverage prediction in MC according to RSRP and RSSI values. The main contributions of this paper can be summarized as follows.

- First, a novel, ensemble learning approach is proposed, called ERTR, for coverage prediction of MC systems by utilizing RSSI values, RSRP, and global positioning system (GPS) coordinates. For this purpose, we utilize a dataset of actual measurements collected from Victoria Island and Ikoyi in Lagos, Nigeria [23].
- Second, we construct the REM by using MATLAB to improve the visualization of coverage prediction. For this purpose, we created a 100 $\times$ 100 grid of data

geographic points in the area of interest to plot the results over a 2D map and a Google Earth digital map.
- Third, in addition to the proposed scheme, we assess the performance of seven ML regression algorithms: RF, a bagging regressor, SVR, KNN, a deep neural network (DNN), GPR, and the decision tree (DT). Additionally, we include a widely used benchmark algorithm called OK for coverage prediction. To compare the proposed ERTR algorithm with the baseline schemes, a 5-fold cross-validation technique is employed, measuring the relative error, mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R^2 score). Through extensive simulations, we validate that the proposed ERTR algorithm outperforms the baseline schemes, offering the highest accuracy while maintaining a low computational load.
- Fourth, to validate the superiority of the proposed ERTR in terms of complexity, we provide a computational complexity analysis between the proposed ERTR, and the baseline algorithms: RF, Bagging, and OK.

The rest of the paper is structured as follows. The dataset is described in Section II. In Section III, we present the coverage prediction methodology, including an overview, a model evaluation, and the ERTR scheme. In Section IV, we provide the numerical results, the computational complexity analysis, and the graphical results. Finally, conclusions are described in Section V.

II. DATASET DESCRIPTION

In this paper, we utilize the publicly available dataset described in [23] composed of key performance indicator parameters such as RSRP, RSSI, logging time, and GPS coordinates. The dataset contained 42,498 instances of each parameter. The measurement campaign was carried out in dense urban environments around Victoria Island and Ikoyi in Lagos, Nigeria, as shown in Figure 1. The dataset was collected with a 4G LTE test modem mounted on a computer housed in a test vehicle driving at 30 km/h. Note that user equipment periodically measures RSRP to perform cell selection/reselection and handover processes in 4G LTE, as well as in 5G NR networks [21]. Therefore, the proposed segmental-approach-based prediction model can easily be adapted to 5G NR network parameters in the future. Formally, we denote the features and labels of the dataset with $D = (m_i, r_i)$ where $m_i \in R^2$, $i \in \{1, 2, \ldots, M\}$, in which M is the number of instances, and m_i is longitude and latitude coordinates. Meanwhile, $r_i \in R$ represents the target label given by the RSRP value, expressed in decibel milliwatts. In this paper, we also consider the analysis of the RSSI as a target value. Similar to RSRP, RSSI is the signal strength received by the user equipment, but RSSI measurements include the main signals, co-channel non-serving signals, adjacent-channel interference, and thermal noise on the specified frequency band [24]. Therefore, the features of the dataset correspond to the longitude and latitude coordinates, and the target values are the RSRP and RSSI values. By measuring the RSRP and RSSI in several positions determined by longitude and latitude coordinates, it is possible to estimate the signal strength and gain insights into the signal's propagation characteristics. REMs are constructed by measuring the RSRP and RSSI at various points in a given area, which is then used as a dataset to build the proposed ERTR model. This model can estimate the RSRP and RSSI at any location within the coverage area.

III. COVERAGE PREDICTION METHODOLOGY

A. OVERVIEW

Our objective is to construct an ML regression framework to predict the outdoor propagation coverage of MC networks by entering data measurement points. First, to design the deployment model, we start by developing the training stage, which is programming in Python software. In this sense, the input dataset is divided into two subsets: the training dataset and the validation (or testing) dataset for hyperparameter tuning of the model. Hence, we adjusted the parameters according to the best results obtained in the evaluation procedure based on 5-fold cross-validation of relative error, MAE, RMSE, and R^2 score metrics. Accordingly, the trained model was ready to be used in the deployment stage. Next, we meshed the target area by creating a grid 100×100 points based on the minimum and maximum geographic coordinates of the dataset so that the grid covered the entire area of interest. After that, we performed feature normalization based on Z-score and then defined the ML regression-based method to be applied to the prediction task. Consequently, the coverage prediction given by the r_i values was obtained for each of the points on the grid by utilizing the model previously trained with the points of the dataset. Afterward, the predicted values of the ML-based framework, along with the longitude and latitude coordinate points of the grid, were exported to a MATLAB file. Then, we loaded that file into MATLAB to build the coverage map. For this purpose, we converted the GPS location measurements into the Universal Transverse Mercator (UTM) coordinate system to build a mesh where the new points were predicted. As a result, the REM with the predicted data points was obtained as a pseudocolor plot, which is drawn as a 2D map by applying the pcolor function. To further improve the visualization, the REM was plotted over a map from Google Earth by using the ge_imagesc function. Finally, we included a bar graph to identify the relationship between our data and the colors displayed in every chart. Figure 2 illustrates the aforementioned procedure.

1) MODEL EVALUATION

Figure 3 explains one iteration of the 5-fold cross-validation [25] used to evaluate the ML-based model. The dataset was divided into 80% for training and 20% for testing. The values predicted by the model were compared with real values from the test data to calculate the relative error, MAE, RMSE, and R^2 score, as defined in Section IV.

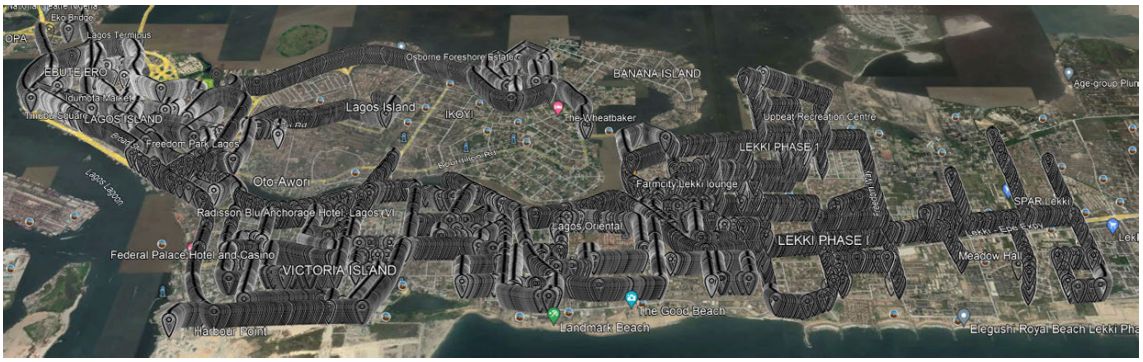


FIGURE 1. Area of interest with the location of the measured points.

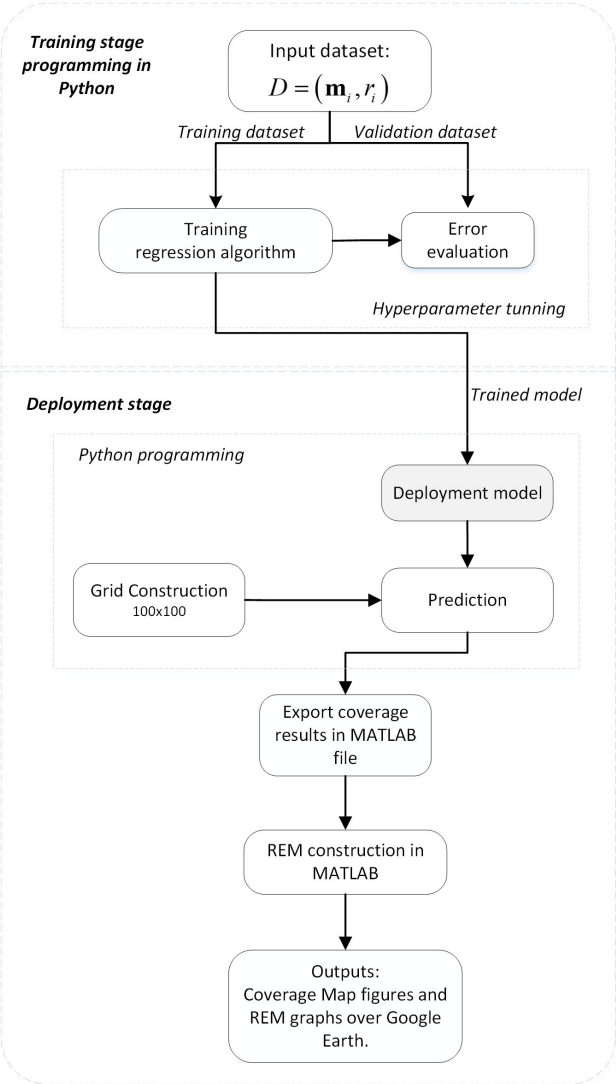


FIGURE 2. Diagram representing the process to build the REM.

B. ERTR-BASED FRAMEWORK FOR COVERAGE PREDICTION AND REM CONSTRUCTION

In this paper, we investigate an ERTR scheme to predict the coverage of outdoor propagation for 4G MC given by the

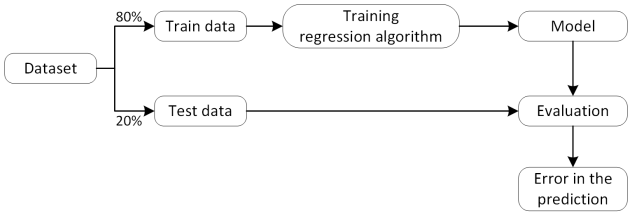


FIGURE 3. Diagram of the model evaluation method based on 5-fold cross validation.

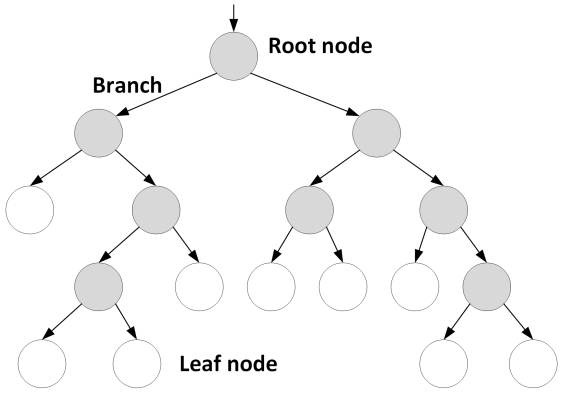


FIGURE 4. The decision tree scheme.

numerical values of r_i. The ERTR algorithm is a supervised ensemble learning model [12] that combines the prediction of various individual trees, in which the whole training dataset is used to create each DT. Figure 4 illustrates the structure of a DT where new instances perform top-down learning to make predictions. For example, every new instance begins in the root node, moves along the branches, and goes through child nodes until it reaches a leaf node [26]. Two rules differentiate this ERTR algorithm from similar ensemble techniques like RF. First, ERTR chooses a random subset of features for each tree from all available features. Second, the split procedure of ERTR relies on a random selection of a splitting value for each of the selected features.

In detail, given the features of the training dataset, $M = \{m_1, m_2, \ldots, m_M\}$, where M is the number of instances in

the training dataset; the samples, $m_i = \{x_1, x_2, \ldots, x_N\}$, constitute an N-dimensional vector, and x_j denotes the feature, in which $j \in \{1, 2, \ldots, N\}$. In each DT created by the ERTR algorithm, S_c represents the subset of instances in the training dataset at child node c. Therefore, at each node c, the best split is based on S_c and a random subgroup of features from Algorithm 1. Next, S_c at c is divided into two subsets: S_c^{right} includes samples that satisfy the two rules of the extra-tree algorithm, and S_c^{left} includes the rest of the training instances. Furthermore, we use mean square error (MSE) [7] to evaluate the quality of a split, i.e., the best division is selected in accordance with the lowest MSE. The process is repeated in each child node until reaching the minimum number of samples for the split, $v_{\min}$. On the other hand, during the testing procedure, a test sample goes through each DT and traverses each child node. During the process, the test sample uses the best split to go to the right or left child node until reaching a leaf node. The prediction for the test sample is given by the leaf node for each DT, and the final prediction of the ERTR algorithm is defined as the average of the F decision trees.

Algorithm 1 Splitting Algorithm of the ERTR-Based Framework

inputs: Training subset $S_c = \{c_1, c_2, \ldots, c_{Q_s}\}$, where the sample $c_i = \{x_1, x_2, \ldots, x_N\}$ is a N-dimensional vector. R is the number of randomly selected features, and the minimum number of samples required to split a node, $v_{\min}$.
If $Q_s < v_{\min}$
    Stop splitting and set the node as leaf node.
else
    Choose a random subgroup of R features $\{x_1, x_2, \ldots, x_R\}$ among the initial N features.
    For each feature r in the subgroup **do:**
        Calculate the maximum value, $x_r^{\max}$, and the minimum value, $x_r^{\min}$, of the feature r in the subset S_c.
        Get a random split value, x_r^c, uniformly taken in the range $\left[x_r^{\min}, x_r^{\max}\right]$.
        Choose $\left[x_r < x_r^c\right]$ as a candidate split.
    End for
    Obtain a split $\left[x_* < x_*^c\right]$ such that $MSE(x_*^c) = \min_{r=1,\ldots,R} MSE(x_R^c)$
Output: best split rule $\left[x_* < x_*^c\right]$ at the child node c.

IV. NUMERICAL RESULTS

In this section, we present simulation results from MC coverage prediction based on RSRP and RSSI, as well as REM construction over dense urban environments around Victoria Island and Ikoyi in Lagos, Nigeria [23]. First, we present the performance evaluation from 5-fold cross-validation of the proposed ERTR algorithm and the additional baseline ML algorithms: RF [7], the bagging regressor [27], [28], [29], KNN, the DNN, GPR [28], the DT, and SVR with a radial basis function (RBF) kernel [30]. Moreover, in our comparative approaches, we include OK, an interpolation technique for cellular coverage prediction [2], [31]. For the OK algorithm, we used the module previously developed in Python, named PyKrige [32]. Second, we present graphic results from REM construction on a 2D map and the Google Earth digital map.

A. EVALUATION WITH 5-FOLD CROSS VALIDATION

In this subsection, we assess the performance of the ML regression schemes and OK by applying 5-fold cross-validation [25] to obtain the following error metrics: relative error, MAE, RMSE, and the R^2 score. This procedure was described in Section III-A, and the results are the average of several repetitions of 5-fold cross-validation.

Specifically, the relative error is the absolute error between the predicted value $\widehat{r_i}$ and the real measure, r_i, divided by the real measure. Therefore, it provides insight into how well the model performs across a range of values, and a lower relative error indicates better performance. Regarding MAE, it measures the average absolute difference between the predicted and actual values, indicating how close the predictions are to the actual values [33]. Meanwhile, RMSE is the square root of the average of the squares of the differences between the actual value and the estimated value. The equations for relative error, MAE, and RMSE, are expressed in (1), (2), and (3), respectively:

$$Er_{Relative} = \frac{|\widehat{r_i} - r_i|}{r_i}, \tag{1}$$

$$Er_{MAE} = \frac{\sum_{i=1}^{m} |r_i - \widehat{r_i}|}{m}, \tag{2}$$

where m is the number of samples.

$$Er_{RMSE} = \sqrt{\frac{\sum_{i=1}^{m} (r_i - \widehat{r_i})^2}{m}}. \tag{3}$$

Note that the lower the value of the aforementioned metrics, the better the performance, unlike R^2 score where a higher value is better. The upper bound is 1, which indicates a perfectly accurate prediction. R^2 score can be expressed as follows:

$$Er_{R2} = 1 - \frac{\sum_{i=1}^{m} (r_i - \widehat{r_i})^2}{\sum_{i=1}^{m} (r_i - \overline{r_i})^2}, \tag{4}$$

where the numerator of the second term is the mean error given by the summation of squares of the residual prediction errors, while the denominator represents the variance, where $\overline{r_i}$ is the average target value [34], [35]. The main idea of the R^2 score is to measure the proportion of the variance in the dependent variable (i.e., the target variable being predicted)

that is predictable from the independent variables (i.e., the features used for prediction) in a regression model. The score has an upper bound of 1 which represents a perfectly accurate prediction, but there is no lower bound, implying that predictions can be extremely inaccurate. If the score is around 0, it can be considered as good as randomly guessing around the mean, $\bar{r}_i$.

In summary, the MAE measures the average absolute difference between predicted and actual values, while the relative error measures the error as a percentage of the actual value. Thus, MAE is scale-dependent, while relative error provides a scale-independent measure of accuracy. Regarding the R^2 score, it measures the amount of variability in the target variable that is explained by the model. Including all these metrics in the analysis provides a more comprehensive evaluation of the model's performance, as each metric captures different aspects of the model's accuracy and fits the data.

Accordingly, Figure 5 and Figure 6 show the number of RSRP training samples versus relative error and RMSE, respectively. From Figure 5 and Figure 6, we observe that as the number of training samples increased, the performance of the investigated algorithms improved. Therefore, we used 36,000 samples for the training procedure of the models. Moreover, we observe that worse performance was obtained from the DNN, SVR, and GPR. On the other hand, we can see that lower relative error values and lower RMSE were achieved by the ERTR ensemble learning algorithm, followed by RF and the bagging regressor. Therefore, in Figure 7 and Figure 8, we analyze the performance of these ensemble learning algorithms in terms of training time and RMSE, respectively, when varying the number of trees. In addition, from Figure 5 and Figure 6, we can see that OK achieved performance close to RF and the bagging regressor. Thus, the proposed ERTR performed best compared to the benchmark schemes. This is because ERTR improves the reduction of bias and variance by utilizing two main strategies. First, it samples the entire dataset and randomizes the selection of the node split, which differs from RF, which uses Bootstrap with replicas and selects the optimum split. The bagging regressor trains each regressor model on random subsets with the replacement of the original training set and then aggregates the individual predictions by averaging them to give a final prediction. Note that the values in the parameters of each algorithm in the simulation results are set based on the best results through hyperparameter tuning and several experiments. For instance, with the OK algorithm, we used the number of points closest to 2, and we selected power, loop, and bool as the variogram model, backend, and mask, respectively.

For the DT, we set the maximum depth to 50 and the minimum samples split parameter to 2. To set the hyperparameters for the DNN model, we evaluated the suitable number of neurons, hidden layers, and learning rate through extensive simulation results and hyperparameter tuning performed on the training data. In this sense, utilizing too many neurons and hidden layers can lead to overfitting, where the network

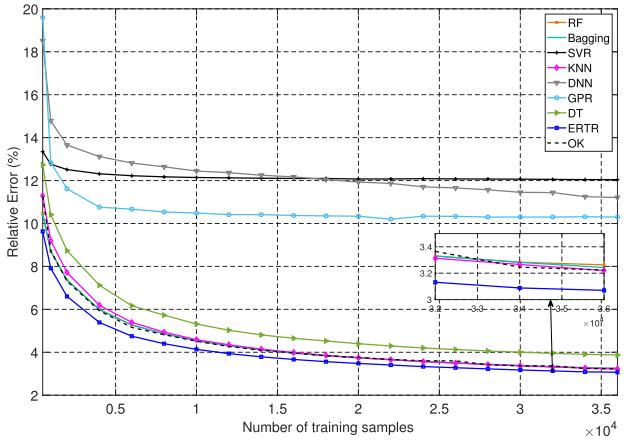


FIGURE 5. Number of RSRP training samples versus percentage of relative error.

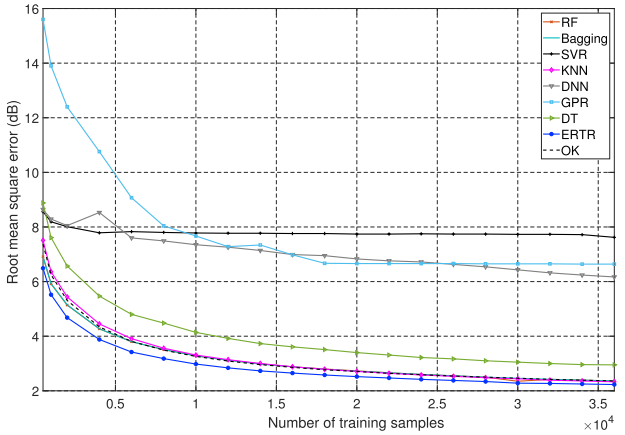


FIGURE 6. Number of RSRP training samples versus RMSE.

becomes overly specialized to the training data and performs poorly on new, unseen data. On the other hand, using too few neurons and hidden layers may result in underfitting, where the network fails to capture remarkable patterns in the data. Similarly, the learning rate determines the step size at which the network adjusts its weights during the training process. A larger learning rate allows for faster convergence but can lead to overshooting the optimal weights and potentially unstable training. By contrast, a smaller learning rate can improve stability but might result in slower convergence behavior or getting trapped in local optimal. Therefore, we selected an appropriate learning rate through experimentation to find a balance between convergence speed and stability. Accordingly, for the DNN, we used four hidden layers. The number of neurons per hidden layer was 100, 50, 100, and 50. We used the ReLU activation function and Adam as the solver, with the learning rate set to 0.0001 and the maximum number of iterations at 300. To establish the best values for the ERTR, RF, and bagging regressor parameters, we analyzed the results obtained by these algorithms with different numbers of regressor trees, samples, and required training times. In this sense, Figure 7 and Figure 8 show the

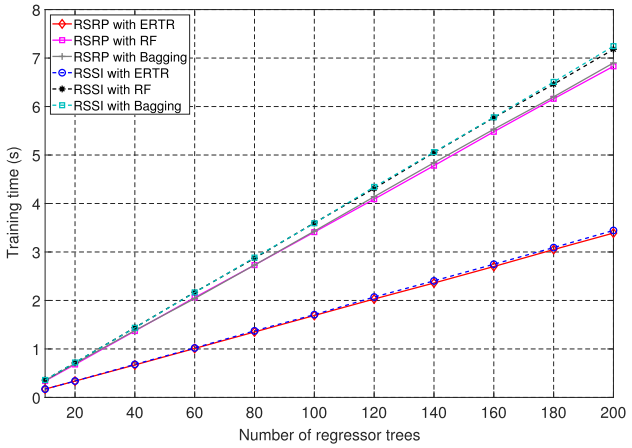


FIGURE 7. Number of trees versus training time.

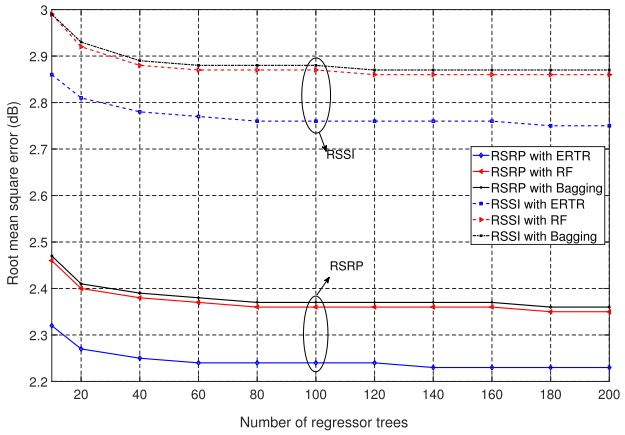


FIGURE 8. Number of trees versus RMSE.

performance of these algorithms in terms of training time and RMSE based on the number of regressor trees. Specifically, Figure 7 shows the number of trees utilized by ERTR, RF, and the bagging regressor versus the training time to obtain the two target values (RSRP and RSSI). Here, we appreciate that the training time increases by using more regressor trees. By contrast, Figure 8 shows that from 180 regressor trees, the RMSE for both RSRP and RSSI did not vary remarkably. Therefore, for our purposes, we utilized 200 regressor trees for ERTR, RF, and the bagging regressor. Moreover, we set the maximum tree depth for ERTR and RF equal to 50. In addition, from Figure 7 and Figure 8, it is noteworthy that ERTR outperformed RF and the bagging regressor in both training time and RMSE. These results validate the efficiency of the proposed ERTR algorithm, which achieved the lowest error with less computational time. Figure 9 shows the training time versus the number of RSRP samples. In Figure 9, we include the performance by ERTR, RF, the bagging regressor, and OK. It is worth noting that the OK algorithm has been widely utilized for coverage prediction in MC environments [2], [14]. Overall, from Figure 9, we can see that as the number of samples increased, the training time increased. However, we can also see from Figure 9 that the ensemble learning methods required less computational time, whereas OK required the longest training time. Consequently, from Figure 7 and Figure 9, we verify that ERTR needed the shortest training time, which results in a computational load reduction. We used a PC with an AMD Ryzed 9 5900X CPU and 48GB of main memory.

Figure 10 and Figure 11 show the number of RSSI training samples versus relative error and RMSE, respectively. Similar to Figure 5 and Figure 6, from Figure 10 and Figure 11, we can observe that as the number of training samples increased, the performance of the investigated algorithms was enhanced. Moreover, the DNN, SVR, and GPR schemes had a higher relative error and RMSE than the OK method and the ensemble learning algorithms (RF, bagging regressor, and ERTR). However, it is remarkable from Figure 10 and Figure 11 that the proposed ERTR outperformed the baseline

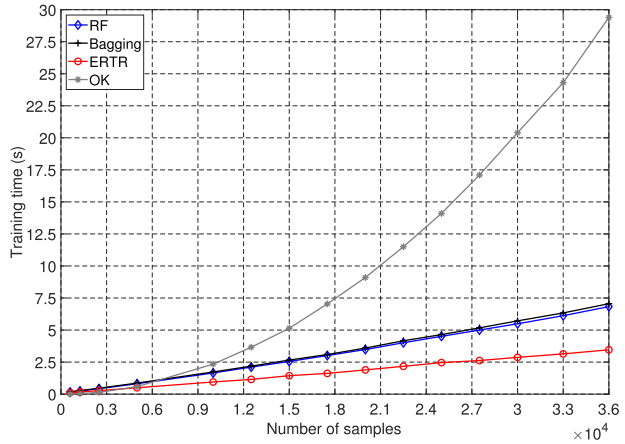


FIGURE 9. Computation time versus the number of RSRP samples.

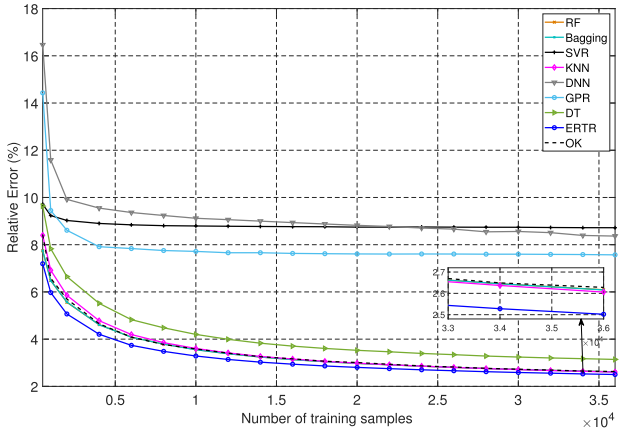


FIGURE 10. Number of RSSI training samples versus relative error.

schemes by achieving the lowest relative error and RMSE, respectively.

Table 1 and Table 2 compare regression performance using RMSE, MAE, and R^2 score for RF, the bagging regressor, SVR, KNN, GPR, the DNN, DT, ERTR, and OK. Recall that

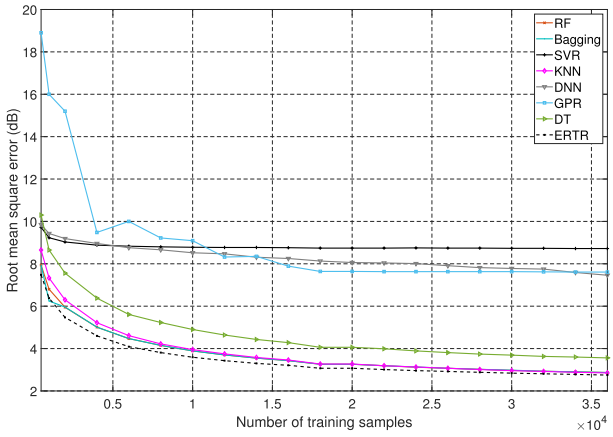


FIGURE 11. Number of RSSI training samples versus RMSE.

TABLE 1. Comparison of RSRP target values from the ERTR algorithm and the benchmark schemes.

Algorithm	RMSE (dB)	MAE (dB)	R^2 score
ERTR	2.23	1.58	0.94
RF	2.36	1.67	0.93
Bagging	2.36	1.67	0.93
SVR	7.62	6.03	0.25
KNN	2.33	1.66	0.93
GPR	6.64	5.26	0.44
DNN	6.17	4.85	0.52
DT	2.95	2	0.89
OK	2.36	1.66	0.93

TABLE 2. Performance comparison between the proposed ERTR algorithm and benchmark schemes according to RSSI target values.

Algorithm	RMSE (dB)	MAE (dB)	R^2 score
ERTR	2.75	2.02	0.92
RF	2.78	2.10	0.92
Bagging	2.87	2.11	0.92
SVR	8.72	6.90	0.24
KNN	2.85	2.10	0.92
GPR	7.61	6.04	0.42
DNN	7.46	5.94	0.44
DT	3.56	2.02	0.92
OK	2.90	2.11	0.92

the parameters of each algorithm in the simulation results are set according to the best results through hyperparameter tuning and several experiments. The results in Table 1 are based on RSRP target values, whereas those in Table 2 are from RSSI target values. From Table 1 and Table 2, we can see that ERTR obtained the lowest error measurements. Thus, we can appreciate that ERTR outperformed the other ML techniques and OK. Moreover, it is remarkable that SVR and GPR presented worse error measurement percentages, followed by DNN.

B. COMPUTATIONAL COMPLEXITY ANALYSIS

In this subsection, we analyze the computational complexity of the proposed ERTR method and the comparative schemes: RF, Bagging, and OK. Accordingly, the computational complexity of the proposed ERTR depends on the number of regression trees, the number of features, the number of samples, and the maximum depth of trees. Specifically, the computational complexity of ERTR can be approximated by $\mathcal{O}(F \cdot N \cdot M \cdot t_d)$, where F is the number of trees, N is the number of features, M is the number of training samples, and t_d is the maximum tree depth. In our simulations, we set the maximum tree depth to $t_d = 50$ and included all available features when selecting the best split, i.e., $R = N = 2$. As shown in Fig. 7, the training time exhibits linear growth since the computational complexity is directly proportional to the number of trees, F, while the number of samples, M, and the number of features, N, remain fixed.

Similarly, in the case of RF, the computational complexity is given by $\mathcal{O}(F \cdot N \cdot M \cdot t_d)$ [36]. However, ERTR achieves lower computational time compared to RF because it makes use of a random threshold to split the data at each node, without searching for the best possible threshold like in RF. In the case of Bagging, the computational complexity is given by $\mathcal{O}(E_B \cdot L_B)$ [36], where E_B is the number of base regressors, and L_B is the computational complexity of training a single base regressor. In our simulations, we use the decision tree regressor as the base estimator, which has a complexity of $L_B = \mathcal{O}(N \cdot M \cdot t_d)$. On the other hand, the computational complexity of training for OK is given by $\mathcal{O}(M^3)$ [37], which leads to the cubic growth of the training time as observed in Figure 9.

C. GRAPHICAL RESULTS OF REM CONSTRUCTION

In this subsection, we present a grid of 100×100 points covering the area of interest from the dataset [23] with RSSI and RSRP predicted values from the trained regressor algorithm processed as described in Section III-A. The threshold values for RSSI are defined as follows: values higher than -70 dBm are considered excellent signal strength reception; values from -70 dBm to -85 dBm are considered good reception; -90 dBm to -100 dBm is considered fair reception, and less than -100 dBm is poor. Meanwhile, the threshold values for RSRP are defined as follows: values higher than -80 dBm are considered excellent signal strength reception; values from -80 dBm to -90 dBm are considered good reception; -90 dBm to -100 dBm is considered fair reception, and values less than -100 dBm are poor.

Figure 12 and Figure 13, respectively, show the MC coverage map predictions for RSRP and RSSI target values on a 2D map by applying the proposed ERTR algorithm and the RF and bagging regressor baseline schemes. From Figure 12 and Figure 13, we observe that RF and the bagging regressor presented abrupt changes of color on the 2D map, which makes it difficult to identify critical points where the signal is decreasing. This makes coverage prediction unreliable.

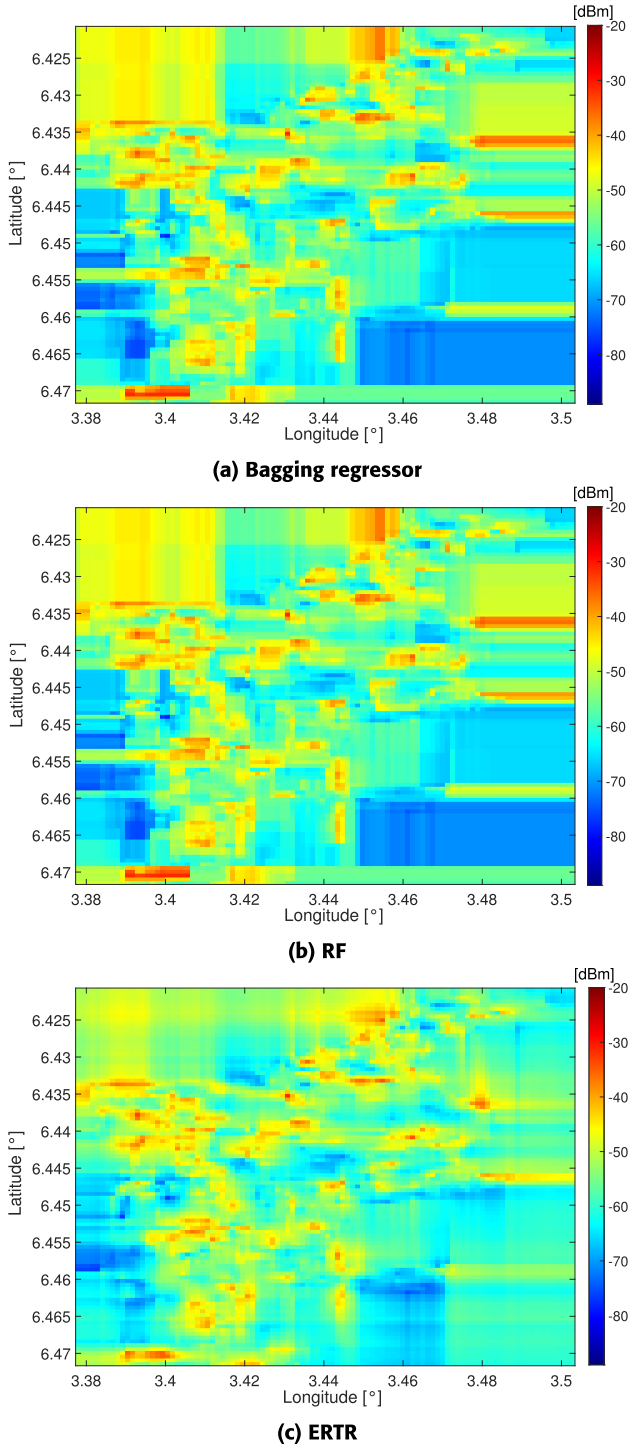


FIGURE 12. REM visualization on 2D maps for RSRP target values from (a) the bagging regressor, (b) RF, and (c) ERTR.

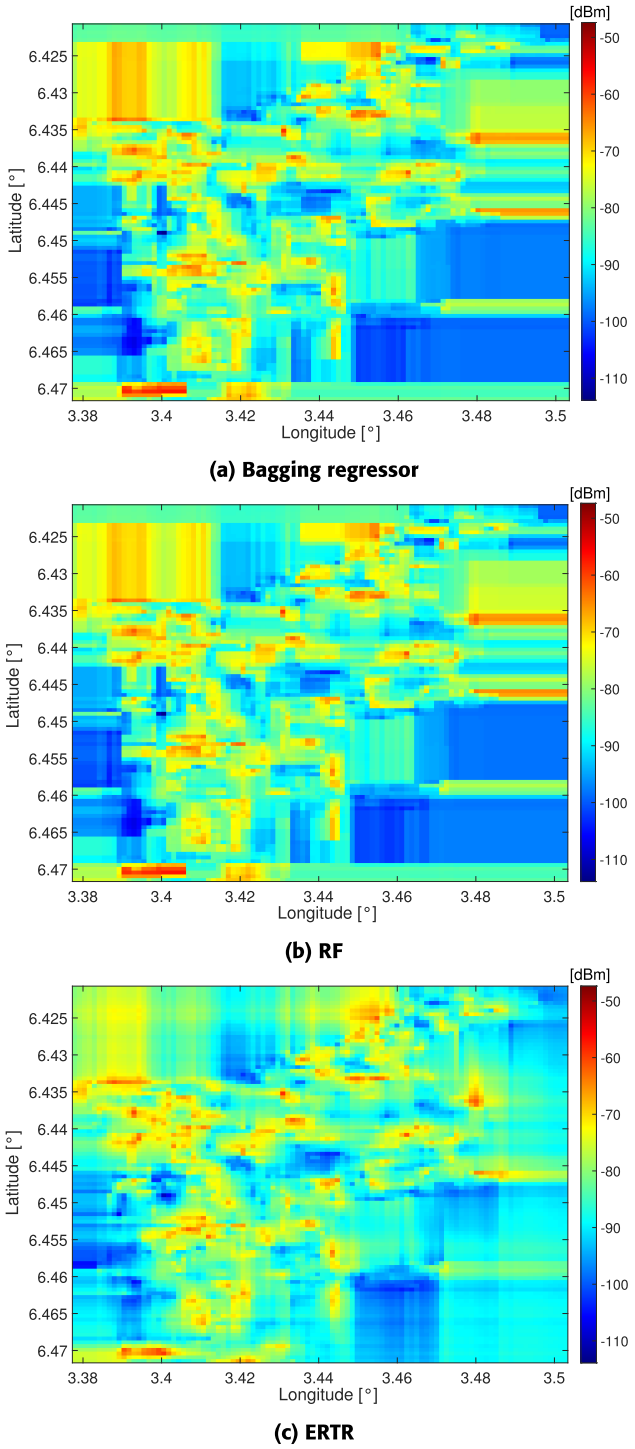
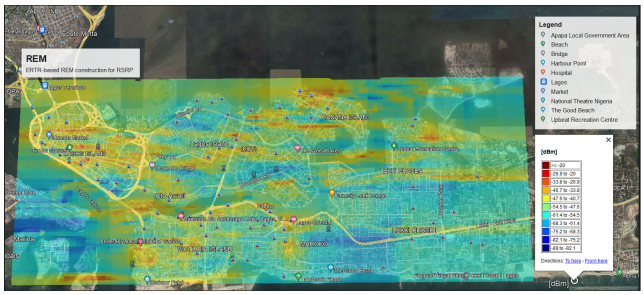
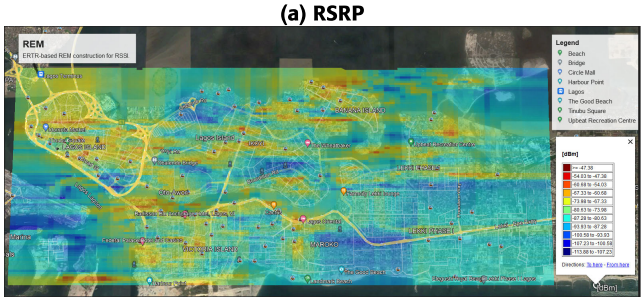


FIGURE 13. REM visualization on 2D maps for RSSI target values from (a) the bagging regressor, (b) RF, and (c) ERTR.

By contrast, the 2D coverage maps obtained by ERTR tend to better generalize the prediction points, since we can appreciate how the signal strength is degrading without abrupt changes. In this sense, ERTR can better detect the quality of signal reception, where we can see areas with good reception and those with shadow areas. Note that the quality of REMs constructed using machine learning techniques depends on several factors, including the quality and quantity of the data used for training, the chosen algorithm for constructing the maps, and the complexity of the environment being mapped. As a result, it is essential to carefully collect and preprocess data, as well as thoroughly test and validate the radio maps to



(a) RSRP



(b) RSSI

FIGURE 14. ERTR-based REM construction on a Google Earth map for (a) RSRP and (b) RSSI.

ensure their quality and reliability. In this regard, the quality of the REMs for ensemble methods based on decision tree regressors primarily depends on how the algorithm selects the split rule in each child node. In the case of RF, the set of possible split points at the child node c is chosen from the feature values of the samples in the subset of the training dataset, S_c. Consequently, only specific values are available for choosing the threshold of the best-split rule at each node. Conversely, in the case of ERTR, the split value is randomly selected within a range based on the subset of the training dataset at node c, as described in lines 7 to 9 of Algorithm 1. This randomness leads to a smoother representation of the final REMs.

Figure 14 shows graphical results for RSSI and RSRP values on a Google Earth map after following the procedure described in Section III-A by applying the ERTR algorithm. From Figure 14, we can identify coverage predictions according to the geographic coordinates that allow us to improve the signal quality reception by installing relay nodes in those points where quality is poor or by adjusting transmission parameters such as antenna height and tilt angle [21]. This mechanism may help operators to improve network planning in MC systems or any radio system.

V. CONCLUSION

In this paper, we proposed a novel MC coverage prediction approach based on a supervised ML regression algorithm. In particular, we designed an ERTR-based scheme to predict coverage through RSRP and RSSI values in an outdoor-to-outdoor propagation environment. For this purpose, we used real measurements carried out using the 4G LTE frequency band in dense urban environments around Victoria Island and Ikoyi in Lagos, Nigeria. Furthermore, we investigated the performance of the OK method, which has been utilized for coverage prediction tasks. In addition, the following ML regression techniques were considered as comparative approaches: RF, the bagging regressor, SVR, KNN, a DNN, GPR, and the DT. It is noteworthy that the ensemble learning techniques (ERTR, RF, and the bagging regressor) achieved higher performance than the other ML schemes. Moreover, it is worth highlighting that OK obtained error metrics closer to RF and the bagging regressor. However, OK incurs high computational costs that tend to get worse when increasing the number of samples. Through numerical results, we showed that the proposed ERTR outperformed the benchmark schemes in terms of computational cost, relative error, RMSE, MAE, and R^2 score. Furthermore, we constructed a REM 2D map on a Google Earth map that provided better visualization of signal quality according to the geographic coordinates, which can help network operators to improve the planning of an MC network.

REFERENCES

[1] P. Maiti and D. Mitra, "Ordinary Kriging interpolation for indoor 3D REM," *J. Ambient Intell. Hum. Comput.*, vol. 2022, pp. 1–15, Mar. 2022, doi: 10.1007/s12652-022-03784-2.

[2] Z. Han, J. Liao, Q. Qi, H. Sun, and J. Wang, "Radio environment map construction by Kriging algorithm based on mobile crowd sensing," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–12, Feb. 2019, doi: 10.1155/2019/4064201.

[3] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proc. 23rd ACM Nat. Conf.*, 1968, pp. 517–524, doi: 10.1145/800186.810616.

[4] U. Ali, G. Caso, L. De Nardis, K. Kousias, M. Rajiullah, Ö. Alay, M. Neri, A. Brunstrom, and M.-G. Di Benedetto, "Data-driven analysis of outdoor-to-indoor propagation for 5G mid-band operational networks," *Future Internet*, vol. 14, no. 8, p. 239, Aug. 2022, doi: 10.3390/fi14080239.

[5] M. E. Diago-Mosquera, A. Aragón-Zavala, and M. Rodriguez, "Testing a 5G communication system: Kriging-aided O2I path loss modeling based on 3.5 GHz measurement analysis," *Sensors*, vol. 21, no. 20, p. 6716, Oct. 2021, doi: 10.3390/s21206716.

[6] O. O. Erunkulu, A. M. Zungeru, C. K. Lebekwe, and J. M. Chuma, "Cellular communications coverage prediction techniques: A survey and comparison," *IEEE Access*, vol. 8, pp. 113052–113077, 2020, doi: 10.1109/ACCESS.2020.3003247.

[7] C. E. G. Moreta, M. R. C. Acosta, and I. Koo, "Prediction of digital terrestrial television coverage using machine learning regression," *IEEE Trans. Broadcast.*, vol. 65, no. 4, pp. 702–712, Dec. 2019, doi: 10.1109/TBC.2019.2901409.

[8] X. Chen, H. Wu, and T. M. Tri, "Field strength prediction of mobile communication network based on GIS," *Geo-Spatial Inf. Sci.*, vol. 15, no. 3, pp. 199–206, Sep. 2012.

[9] P. Maiti and D. Mitra, "Complexity reduction of ordinary Kriging algorithm for 3D REM design," *Phys. Commun.*, vol. 55, Dec. 2022, Art. no. 101912, doi: 10.1016/j.phycom.2022.101912.

[10] M. Pesko, T. Javornik, A. Košir, M. Štular, and M. Mohorčič, "Radio environment maps: The survey of construction methods," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 11, pp. 1–21, Nov. 2014, doi: 10.3837/tiis.2014.11.008.

[11] D. Plets, W. Joseph, K. Vanhecke, E. Tanghe, and L. Martens, "Coverage prediction and optimization algorithms for indoor environments," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, p. 123, Mar. 2012.

[12] C. E. Garcia, M. R. Camana, and I. Koo, "Ensemble learning aided QPSO-based framework for secrecy energy efficiency in FD CR-NOMA systems," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 649–667, Jun. 2023, doi: 10.1109/TGCN.2022.3219111.

[13] M. Khan and S. Noor, "Performance analysis of regression-machine learning algorithms for predication of runoff time," *Agrotechnology*, vol. 8, no. 1, pp. 1–12, Mar. 2019.

[14] Y. Zhang, J. Wen, G. Yang, Z. He, and J. Wang, "Path loss prediction based on machine learning: Principle, method, and data expansion," *Appl. Sci.*, vol. 9, no. 9, p. 1908, May 2019, doi: 10.3390/app9091908.
[15] M. Sousa, A. Alves, P. Vieira, M. P. Queluz, and A. Rodrigues, "Analysis and optimization of 5G coverage predictions using a beamforming antenna model and real drive test measurements," *IEEE Access*, vol. 9, pp. 101787–101808, 2021, doi: 10.1109/ACCESS.2021.3097633.
[16] H.-S. Jo, C. Park, E. Lee, H. K. Choi, and J. Park, "Path loss prediction based on machine learning techniques: Principal component analysis, artificial neural network, and Gaussian process," *Sensors*, vol. 20, no. 7, p. 1927, Mar. 2020, doi: 10.3390/s20071927.
[17] N. Moraitis, L. Tsipi, D. Vouyioukas, A. Gkioni, and S. Louvros, "Performance evaluation of machine learning methods for path loss prediction in rural environment at 3.7 GHz," *Wireless Netw.*, vol. 27, no. 6, pp. 4169–4188, Aug. 2021, doi: 10.1007/s11276-021-02682-3.
[18] O. Rozenblit, Y. Haddad, Y. Mirsky, and R. Azoulay, "Machine learning methods for SIR prediction in cellular networks," *Phys. Commun.*, vol. 31, pp. 239–253, Dec. 2018.
[19] B. Cha and S.-K. Noh, "Learning using LTE RSRP and NARNET in the same indoor area," in *Proc. 23rd Int. Comput. Sci. Eng. Conf. (ICSEC)*, Oct. 2019, pp. 261–264, doi: 10.1109/ICSEC47112.2019.8974774.
[20] R. He, Y. Gong, W. Bai, Y. Li, and X. Wang, "Random forests based path loss prediction in mobile communication systems," in *Proc. IEEE 6th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2020, pp. 1246–1250, doi: 10.1109/ICCC51575.2020.9344905.
[21] M. F. A. Fauzi, R. Nordin, N. F. Abdullah, and H. A. H. Alobaidy, "Mobile network coverage prediction based on supervised machine learning algorithms," *IEEE Access*, vol. 10, pp. 55782–55793, 2022, doi: 10.1109/ACCESS.2022.3176619.
[22] M. R. Camana, C. E. Garcia, T. Hwang, and I. Koo, "A REM update methodology based on clustering and random forest," *Appl. Sci.*, vol. 13, no. 9, p. 5362, Apr. 2023.
[23] A. L. Imoize, S. O. Tofade, G. U. Ughegbe, F. I. Anyasi, and J. Isabona, "Updating analysis of key performance indicators of 4G LTE network with the prediction of missing values of critical network parameters based on experimental data from a dense urban environment," *Data Brief*, vol. 42, Jun. 2022, Art. no. 108240, doi: 10.1016/j.dib.2022.108240.
[24] A. L. Imoize and O. D. Adegbite, "Measurements-based performance analysis of a 4G LTE network in and around shopping malls and campus environments in Lagos Nigeria," *Arid Zone J. Eng., Technol. Environ.*, vol. 14, no. 2, pp. 208–225, Jun. 2018.
[25] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. IJCAI*, vol. 14, 1995, pp. 1137–1145.
[26] M. R. Camana Acosta, S. Ahmed, C. E. Garcia, and I. Koo, "Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks," *IEEE Access*, vol. 8, pp. 19921–19933, 2020.
[27] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[28] M. Kovačević, N. Ivanišević, P. Petronijević, and V. Despotović, "Construction cost estimation of reinforced and prestressed concrete bridges using machine learning," *J. Croatian Assoc. Civil Eng.*, vol. 73, no. 1, pp. 1–13, Feb. 2021, doi: 10.14256/jce.2738.2019.
[29] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2010, pp. 135–150.
[30] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.
[31] H. Braham, S. B. 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.
[32] B. Murphy. *Kriging Toolkit for Python, Version 1.3.2*. Accessed: Oct. 8, 2017. [Online]. Available: https://pypi.org/project/PyKrige/
[33] P. Schneider and F. Xhafa, "Anomaly detection, classification and CEP with ML methods," in *Anomaly Detection and Complex Event Processing Over IoT Data Streams*. London, U.K.: Elsevier, 2022, pp. 193–233, doi: 10.1016/b978-0-12-823818-9.00020-1.
[34] F. Klemme and H. Amrouch, "Scalable machine learning to estimate the impact of aging on circuits under workload dependency," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 5, pp. 2142–2155, May 2022, doi: 10.1109/TCSI.2022.3147587.
[35] J. Parmar, S. K. Patel, V. Katkar, and A. Natesan, "Graphene-based refractive index sensor using machine learning for detection of mycobacterium tuberculosis bacteria," *IEEE Trans. Nanobiosci.*, vol. 22, no. 1, pp. 92–98, Jan. 2023, doi: 10.1109/TNB.2022.3155264.
[36] M. Abdar, U. R. Acharya, N. Sarrafzadegan, and V. Makarenkov, "NE-nu-SVC: A new nested ensemble clinical decision support system for effective diagnosis of coronary artery disease," *IEEE Access*, vol. 7, pp. 167605–167620, 2019.
[37] X. Zhong, A. Kealy, and M. Duckham, "Stream Kriging: Incremental and recursive ordinary Kriging over spatiotemporal data streams," *Comput., Geosci.*, vol. 90, pp. 134–143, May 2016.

CARLA E. GARCÍA (Graduate Student Member, IEEE) received the B.Eng. degree in electronics and telecommunications engineering from Escuela Politécnica Nacional (EPN), Quito, Ecuador, in 2016, and the M.Sc. and Ph.D. degrees from the University of Ulsan, South Korea, in 2020 and 2023, respectively. She is currently a Research Assistant with the Department of Electrical, Electronic and Computer Engineering, University of Ulsan. Her main research interests include artificial intelligence, security, MIMO communications, NOMA, and optimizations.

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 (GIST), Gwangju, South Korea, in 1998 and 2002, respectively. From 2002 to 2004, he was a Research Professor with the Ultrafast Fiber-Optic Networks Research Center, GIST. In 2003, he was a Visiting Scholar with the KTH 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 research interests include spectrum sensing issues for CRNs, channel and power allocation for cognitive radios (CRs), and military networks, SWIPT MIMO issues for CRs, MAC, and routing protocol design for UW-ASNs, and relay selection issues in CCRNs.