International Journal of Advanced Smart Convergence Vol.12 No.1 92-100 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.1.92

IJASC 23-1-12

Estimating Indoor Radio Environment Maps with Mobile Robots and Machine Learning

Taewoong Hwang[1], Mario R. Camana Acosta[2], Carla E. Garcia Moreta[3], and Insoo Koo[†]

[1]*Master Student, Department of Electrical/Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea*
[2]*PhD, Department of Electrical/Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea*
[3]*PhD Student, Department of Electrical/Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea*
[†]*Professor, Department of Electrical/Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea*
E-mail : [1]yuio124@naver.com, [2]mario_camana@hotmail.com, [3]carli.garcia27@hotmail.com, [†]iskoo@ulsan.ac.kr

Abstract

Wireless communication technology is becoming increasingly prevalent in smart factories, but the rise in the number of wireless devices can lead to interference in the ISM band and obstacles like metal blocks within the factory can weaken communication signals, creating radio shadow areas that impede information exchange. Consequently, accurately determining the radio communication coverage range is crucial. To address this issue, a Radio Environment Map (REM) can be used to provide information about the radio environment in a specific area. In this paper, a technique for estimating an indoor REM using a mobile robot and machine learning methods is introduced. The mobile robot first collects and processes data, including the Received Signal Strength Indicator (RSSI) and location estimation. This data is then used to implement the REM through machine learning regression algorithms such as Extra Tree Regressor, Random Forest Regressor, and Decision Tree Regressor. Furthermore, the numerical and visual performance of REM for each model can be assessed in terms of R2 and Root Mean Square Error (RMSE).

Keywords: *Radio Environment Map, Mobile Robot, machine learning, Random Forest Regression, Extra Tree Regressior, DecisionTreeRegressor*

1. Introduction

In the era of the 4th Industrial Revolution, the advent of advanced technologies such as artificial intelligence, big data, IoT, and robotics has led to the emergence of smart factories, which represent the ultimate convergence of these technologies. Smart factories aim to automate production processes while simultaneously tracking them in real-time, allowing for the automatic determination of the process's status and appropriate measures to be taken accordingly [2]. Wireless communication plays a crucial role in smart factories for several reasons. Firstly, it enables real-time tracking and monitoring of production processes. Secondly, it facilitates

Manuscript Received: January. 16, 2023 / Revised: January. 20, 2023 / Accepted: January. 22, 2023
Corresponding Author: iskoo@ulsan.ac.kr
Tel: +82-52-259-1249, Fax: +82-52-259-1686
Professor, Department of Electrical/Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea

the collection and analysis of vital data about the production process. Finally, wireless communication is often more flexible and easier to deploy than wired communication, making it an ideal choice for communication in a dynamic production environment.

Smart Factory utilizes various communication technologies, including MODBUS, PROFIBUS, DeviceNet, Wi-Fi, Bluetooth, and Zigbee, with wireless communication technology becoming increasingly popular due to its support for low power consumption, ease of deployment, and device flexibility. However, the increasing number of wireless devices can cause interference in the ISM band, and obstacles such as metal blocks in the factory can result in weakened communication signals, leading to radio shadow areas that hinder information exchange. This is a significant issue [3], as these areas can impede control, monitoring, and other manufacturing processes, ultimately decreasing production efficiency and posing safety hazards. Therefore, accurately determining the radio communication coverage range is crucial.

In order to tackle the aforementioned issue, we propose a solution that involves implementing and solving a digital Radio Environment Map (REM) [4-6], which provides details about the radio environment in a specific area. Data collection in the area is crucial to implement REM. However, due to the nature of the factory, it is difficult to collect data for wireless communication shadow areas in wide and dangerous and narrow spaces that are inaccessible to humans. One way to address this problem is by utilizing a Mobile Robot to move along a designated path or operate remotely using an input command key. Therefore, in this paper, we introduce a technique that estimates an indoor Radio Environment Map (REM) by employing a mobile robot and machine learning methods. First, the mobile robot collects and processes data, including the Received Signal Strength Indicator (RSSI) and location estimation. After that, the collected data is used to implement the REM through machine learning regression algorithms such as Extra Tree Regressor, Random Forest Regressor, and Decision Tree Regressor. We assess the efficiency of each model using performance metrics such as R2 and Root Mean Square Error (RMSE), and analyze the visual representation of the REM for each model.

The paper is structured as follows: Section2 explains the coverage prediction approach and the machine learning regression algorithms used. Section 3 outlines the experimental setup, including the configuration of the mobile robot, the types of data collected, the location for data collection, the method for collecting data, and the process for merging the data. In Section 4, numerical results are presented, and the performance of each regression model is compared by evaluating the error metrics of Root Mean Square Error (RMSE) and R-squared (R2). Section 5 includes a visual representation of the Radio Environment Map (REM) and compares the outputs of each regression model. Finally, Section 6 provides a summary of the paper and draws a conclusion.

2. Methodology for predicting radio coverage

Our proposed approach is to develop a machine learning regression-based model that can predict the range of RSSI by utilizing the measurement data points. The model is trained using the data sample set. To visualize the outcomes, we create a 1000 x 1000 grid based on the geographic coordinates of the data set, which covers the entire data sector. The trained model is then applied to predict the coverage of each point on the grid. The processed model values and the X, Y coordinates of the grid are converted into a scatter plot using the Pandas and Matplotlib libraries in Python. The X and Y axes are labeled as X and Y, respectively, while the predicted RSSI value is represented by "c". To achieve this, we use the Jet ColorMap for cmap, and set alpha to 0.05. This process is depicted in Figure 1.

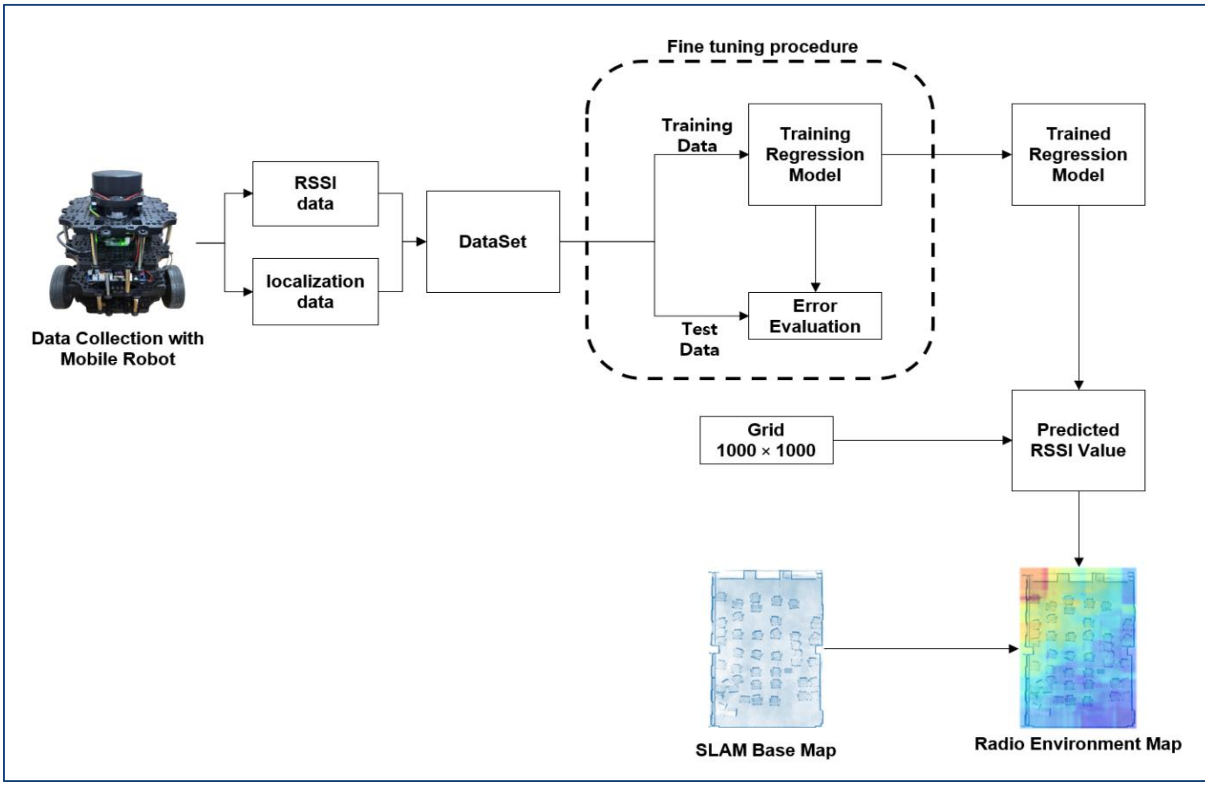

Figure 1. The implementation process of Radio Environment Map (REM) with Mobile Robots and Machine Learning

2.1 Machine Learning Regression Algorithm

Regression algorithms are utilized in machine learning to estimate the relationship between a dependent variable and independent variables by fitting a mathematical function to the data. The most commonly used regression algorithms are linear regression, logistic regression, and polynomial regression. These algorithms can be assessed through a range of metrics such as mean squared error, mean absolute error, and R-squared, which evaluate the accuracy and performance of the model. These metrics are valuable for comparing different algorithms or optimizing the hyperparameters of a particular algorithm. In the paper, to implement the REM through machine learning, we have utilized three regression algorithms: Extra Tree Regressor, Random Forest Regressor, and Decision Tree Regressor.

2.1.1 Decision Tree Regressor

The Decision Tree Regressor[7] is a machine learning algorithm that utilizes a tree-like structure to make predictions based on the rules at each node. Its performance is greatly influenced by the number of nodes from the root to the leaf. This algorithm is known for its interpretability, which makes it a popular choice for regression problems. It can handle missing values in the dataset, but one major drawback of the Decision Tree Regressor is its tendency to overfit. The model can have many nodes and leaves, resulting in a high variance and poor performance on new data. Additionally, small changes in the input data can cause significant changes in the model, making it sensitive to noise.

2.1.2 Random Forest Regressor

The Random Forest Regressor[8] is an ensemble machine learning algorithm that utilizes decision trees to generate predictions. The algorithm constructs several decision trees on random subsets of the training data and features, which are combined to create a strong learner. Random Forest Regressor incorporates both bagging and feature selection methods to enhance the ensemble's performance. Its high accuracy and scalability make it a popular choice for both classification and regression tasks. However, Random Forest Regressor may face computational challenges, particularly when handling large datasets or numerous features.

2.1.3 Extra Trees Regressor

Extra Trees Regressor[9] is a machine learning algorithm that belongs to the class of ensemble methods. Extra Trees Regressor uses an ensemble of decision trees to generate an output. In essence, Extra Trees Regressor is a collection of weak learners that are aggregated to form a strong learner. Each decision tree in the ensemble is built on a random subset of the training data and a random subset of the features at each split. randomness is introduced to increase diversity in the ensemble, which in turn reduces variance and improves performance. additionally, Extra Trees Regressor is highly scalable and can handle large datasets with a large number of features. It can be trained in parallel, which allows for faster computation and reduced training time.

3. Experiment Testbed Configuration

This section will provide a detailed description of the experimental testbed configuration used to obtain measured data for REM construction. Specifically, we will discuss the mobile robot employed in the study, the types of data collected, the location of data collection, the methods used for data collection, and the processing of the collected data.

3.1 Mobile Robot

A mobile robot is a robot that can move independently or under remote operator control, and is used in various industries to perform tasks efficiently and improve productivity. These robots are equipped with sensors, navigation systems, and decision-making algorithms that enable them to move and perform tasks autonomously. The Turtlebot 3 burger [10] was the mobile robot used in the experiment. It consists of several components, including a Single Board Computer (SBC), an embedded controller, a lidar sensor, an IMU sensor, an encoder, and the Robot Operating System (ROS). The SBC, which is powered by a Raspberry Pi, provides algorithm configuration in a Linux environment and uses ROS, an open-source meta operating system for robots, to facilitate communication between different processes. The embedded controller, which uses OpenCR, is primarily responsible for controlling the movement of the mobile robot and utilizing various sensors. The lidar sensor measures the time it takes for laser pulses to be emitted and reflected by the LDS-02, and calculates the position of the reflector, which is used for position estimation. The encoder, a Dynamixel, outputs the position and speed information of rotary and linear motion as electrical signals.

Figure 2. Turtlebot 3 burger used to collect data for REM construction

3.2 Data Organization

The REM data used in the paper includes X, Y coordinates and RSSI values, with the former serving as features and the latter as labels. RSSI is an important metric used in wireless communication to evaluate the quality of wireless signals, and it has applications in network planning, signal mapping, and device location determination. The experiment in the paper focuses on utilizing Wi-Fi to get RSSI. Localization data (X, Y) is used in navigation and location-based services to determine the physical location of the mobile robot. In the experiment, the LIDAR and Wi-Fi interface embedded in the Turtlebot 3 in Figure 2 are used to collect localization data and RSSI values.

3.3 Data Collection Location

The experiment in the paper was conducted in Room 303 of the Electrical and Computer Engineering Building at the University of Ulsan as shown in Figure 3(a). The room is a typical university classroom with desks serving as obstacles. Figure 3(b) illustrates the location of the access point (AP) and the path followed by the mobile robot during data collection.

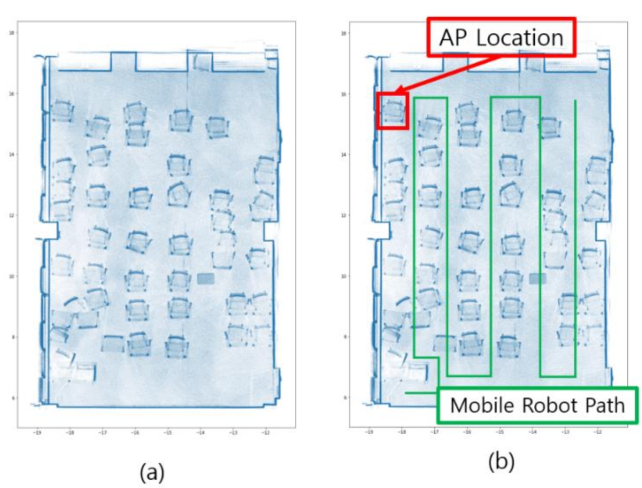

Figure 3. (a) Room 303, Electrical and Computer Engineering Building, University of Ulsan (b) The location of WiFi AP and The path of Mobile Robot

3.4 Data collection methods and data merging

The RSSI data was collected using the built-in Wi-Fi module of the Raspberry Pi and the "iwconfig" command line tool where "iwconfig" command displays information about the wireless network interface, including SSID, signal frequency, quality, and strength. To obtain location data, the LDS-02, LIDAR built in Turtlebot 3 was connected to the Raspberry Pi and merged with encoder and IMU sensor data to create ROS Odometry. The RSSI and location estimation data were collected separately using a shell script in a Linux-based environment and synchronized using timestamps. The merging process was performed by synchronizing the two data sets using a timestamp, which indicates the time when a specific event occurred. Since the RSSI and localization data sets both had timestamps, they were merged based on the timestamps. This ensured that the data was collected at the same time the mobile robot was in operation, and thus was considered synchronized.

4. Numerical Results

For experiment testbed explained in Section 3, we have constructed REM by using configuration methodology suggested in Section 2 for three machine learning methods, such as ExtraTreesRegressor, RandomForestRegressor, and DecisionTreeRegressor techniques. In this section, the authors evaluate the performance of the REMs constructed using each of these machine learning methods, using two error metrics: root mean square error (RMSE) and R-squared (R2).

Root mean square error (RMSE), as depicted in equation (1), is a widely adopted metric for assessing the effectiveness of the regression prediction models.

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

where n refers to the total number of samples, y_i represents the actual value, and $\hat{y}_i$ signifies the predicted value. A lower value of RMSE indicates improved prediction accuracy of the machine learning model.

The R-squared (R2), expressed in equation (2), is a valuable metric to evaluate the performance of prediction models. It measures the degree to which the model fits the data, i.e., how well the input parameters explain the variation in the response variable. A higher R2 value indicates that the model explains a larger proportion of the variability in the data, and thus, signifies better predictive power. R2 values range from 0 to 1, where a value of 1 indicates a perfect fit.

$$R^2 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{2}$$

where $\bar{y}$ is the average value.

Table 1. Error evaluation of dataset

Error metrics	ML Regression Model		
	ExtraTreesRegressor	RandomForestRegressor	DecisionTreeRegressor
RMSE	1.13	1.26	1.58
R2	0.958	0.949	0.926

Table 1 presents the performance of three distinct regression models, namely ExtraTreesRegressor, RandomForestRegressor, and DecisionTreeRegressor, in terms of RMSE and R2. The results reveal that the

ExtraTreesRegressor model achieved the lowest RMSE value of 1.13, followed by the RandomForestRegressor at 1.26, and the DecisionTreeRegressor at 1.58. These findings imply that the ExtraTreesRegressor model demonstrated superior accuracy in predicting the RSSI values compared to the other models.

Moreover, the R2 values for the ExtraTreesRegressor, RandomForestRegressor, and DecisionTreeRegressor were 0.958, 0.949, and 0.926, respectively, indicating that the ExtraTreesRegressor model performed the best in explaining the variance in the response variable. This suggests that the ExtraTreesRegressor model is more effective in capturing the underlying patterns in the data compared to the other models.

Upon comparing these findings with a previous study [11], it was observed that the ExtraTreesRegressor (ET) and RandomForestRegressor (RF) models in this study yielded substantially lower RMSE values of 1.13 and 1.26, respectively, compared to the values of 2.605 and 3.098 reported in the previous research. This suggests that the models used in the present study were more accurate in predicting the outcome variable for the given dataset, which enhances their reliability for future predictions. Overall, the results indicate that the ET model is the most effective in predicting the RSSI values and has significant potential applications in wireless sensor network

5. Graphical Results of the REM

In this section, we present the Radio Environment Maps (REMs) generated by three machine learning models: Extra Trees Regressor, Random Forest Regressor, and Decision Tree Regressor. The REMs are shown in Figure 4, with a grid of 1000 x 1000 points, where the predicted RSSI dBm values are represented by colors. The colors closer to red indicate higher predicted RSSI dBm values, while the colors closer to blue indicate lower predicted RSSI dBm values. The predicted values are also shown on the right side of each figure. As expected, the results show that the predicted RSSI dBm values were higher (closer to -35dBm) when the points were closer to the Access Point (AP) location, as demonstrated in Figure 3(b). Conversely, the predicted values were lower (closer to -60dBm) as the distance to the AP location increased

According to the performance comparison shown in Table 1, we observed that two ensemble models such as ExtraTreesRegressor and RandomForestRegressor had similar relative error values. However, the R2 of the Decision Tree Regressor was relatively lower than that of the ensemble models. This difference can be seen visually in the REM near the Access Point (AP) location as shown Figure 4. Specifically, in Figure 4(a), the AP location was well-represented by the Extra Trees Regressor. Conversely, Figure 4(b) shows that the Random Forest Regressor had high values in many areas surrounding the AP location. Figure 4(c) indicates that the Decision Tree Regressor's color distribution changed rapidly within the AP location range. Upon examining Figure 4 in its entirety, the ensemble models displayed similar performance, whereas the Decision Tree Regressor exhibited lower performance.

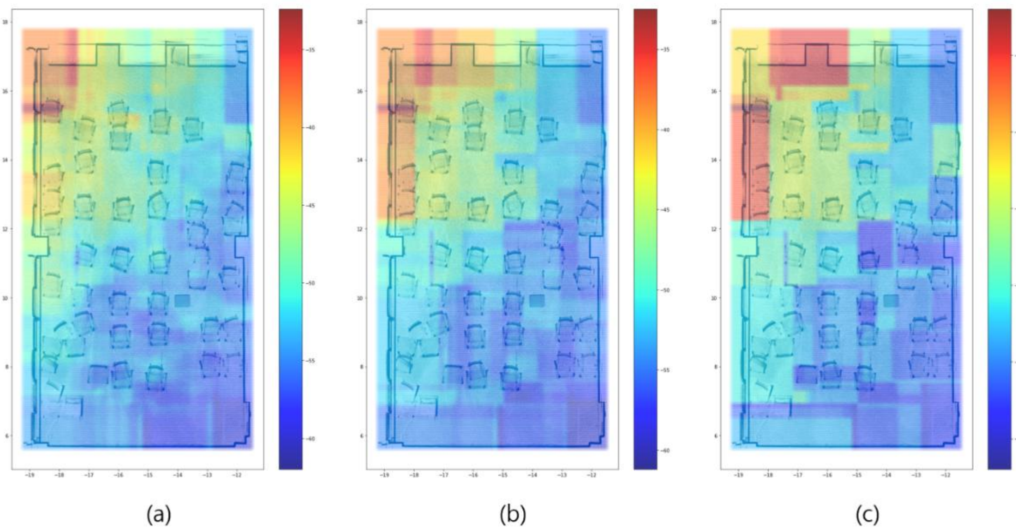

Figure 4. Visualization of the REM using (a) ExtraTreesRegressor, (b) RandomForestRegressor, and (c) DecisionTreeRegressor.

6. Conclusion

In this paper, we employed machine learning and a mobile robot to predict REM. Data was collected using the mobile robot as it drove along a self-determined path. We then employed machine learning regression algorithms, including ExtraTreesRegressor, RandomForestRegressor, and DecisionTreeRegressor, to predict REM. To evaluate the performance of each model, we compared their numerical results using R2 and Root Mean Square Error (RMSE). The ensemble models, ExtraTreesRegressor and RandomForestRegressor, demonstrated similar performance, whereas the DecisionTreeRegressor's performance was relatively lower. Our findings confirmed that utilizing data collected by mobile robots and machine learning can result in superior numerical performance and an easy-to-implement approach for REM prediction.

Acknowledgement

This work was supported in part by the ICT R&D 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.(No. 2022-0-00805, Development of wired/wireless communication technology for installation safety communication network for workers in shaded areas of building ships). In addition, this result was supported by the “Regional Innovation Strategy (RIS)” through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-003)

References

[1] Hassoun, Abdo, et al. "The fourth industrial revolution in the food industry—Part I: Industry 4.0 technologies." Critical Reviews in Food Science and Nutrition (2022): 1-17.
DOI: https://doi.org/10.1080/10408398.2022.2034735

[2] Chien, Chen-Fu, Wei-Tse Hung, and Eddy Ting-Yi Liao. "Redefining monitoring rules for intelligent fault detection and classification via CNN transfer learning for smart manufacturing." IEEE Transactions on Semiconductor Manufacturing 35.2 (2022): 158-165.
DOI: https://doi.org/10.1109/TSM.2022.3164904

[3] Liang, Yewen, Qingfeng Jing, and Weizhi Zhong. "A novel method to reduce indoor signal power fluctuation using multiple adaptive terahertz sources." International Journal of Communication Systems 35.11 (2022): e5196.. DOI: https://doi.org/10.1002/dac.5196

[4] S.-F. Chou, H.-W. Yen, and A.-C. Pang, "A REM-Enabled Diagnostic Framework in Cellular-Based IoT Networks," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5273–5284, Jun. 2019. DOI: https://doi.org/10.1109/JIOT.2019.2900093

[5] S. Bi, J. Lyu, Z. Ding, and R. Zhang, "Engineering Radio Maps for Wireless Resource Management," IEEE Wireless Communications, vol. 26, no. 2, pp. 133–141, Apr. 2019. DOI: https://doi.org/ 10.1109/MWC.2019.1800146

[6] C. E. G. Moreta, M. R. C. Acosta, and I. Koo, "Prediction of Digital Terrestrial Television Coverage Using Machine Learning Regression," IEEE Transactions on Broadcasting, vol. 65, no. 4, pp. 702–712, Dec. 2019. DOI: https://doi.org/10.1109/TBC.2019.2901409

[7] Christa, Sharon, V. Suma, and Uma Mohan. "Regression and decision tree approaches in predicting the effort in resolving incidents." International Journal of Business Information Systems 39.3 (2022): 379-399. DOI: https://doi.org/10.1504/IJBIS.2022.122342

[8] L. Breiman, "Random forest," Mach. Learn., vol. 45, no. 1, pp. 5-32, Oct. 2001. DOI: https://doi.org/10.1023/A:1010933404324

[9] Serpi, Helena, and Christina Tanya Politi. "Machine Learning assisted Indoor Visible Light Communications Radio Environment Maps." 2022 Panhellenic Conference on Electronics & Telecommunications (PACET). IEEE, 2022. DOI: https://doi.org/10.1109/PACET56979.2022.9976382

[10] ROBOTIS e-Manual, "TurtleBot3", https://emanual.robotis.com/docs/en/platform/turtlebot3/features

[11] M. F. Ahmad Fauzi, R. Nordin, N. F. Abdullah and H. A. H. Alobaidy, "Mobile Network Coverage Prediction Based on Supervised Machine Learning Algorithms," in IEEE Access, vol. 10, pp. 55782-55793, 2022 DOI: https://doi.org/10.1109/ACCESS.2022.3176619