Hyperparameter Optimization of Random Forest Algorithm to Enhance Performance Metric Evaluation of 5G Coverage Prediction

Hajiar Yuliana[1]*, Iskandar[2], Hendrawan[2], Sofyan Basuki[1], M. Reza Hidayat[1], Atik Charisma[1], Hurianti Vidyaningtyas[3]

[1]Electrical Engineering, Faculty of Engineering, Universitas Jenderal Achmad Yani, Terusan Jend. Sudirman, Cimahi, West Java. 40525. Indonesia

[2]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40132, Indonesia

[3]The University Center of Excellence for Intelligent Sensing-IOT, Telkom University, Bandung, Indonesia

*Email : [1]hajiar.yuliana@lecture.unjani.ac.id

ARTICLE INFORMATION

Received on 31 March 2024
Revised on 09 June 2024
Accepted on 27 June 2024

Keywords:
Coverage Prediction
Hyperparameter Optimization
Machine Learning
Random Forest
RMSE

ABSTRACT

Utilizing Fifth-Generation (5G) technology has become crucial for developing advanced and efficient telecommunications networks. Accurate 5G coverage prediction is essential for optimal network planning and ensuring high-quality user experiences. This study investigates the use of the Random Forest algorithm for 5G coverage prediction, with a specific focus on hyperparameter optimization to enhance model performance. A comprehensive measurement campaign was conducted in Bandung City, Indonesia, and data was collected through the drive test method. The dataset underwent rigorous preprocessing, including data cleaning, normalization, and feature engineering, to ensure quality and consistency. Hyperparameter optimization was performed using Grid Search, systematically evaluating combinations of "n_estimators", "max_depth", "min_samples_split", "min_samples_leaf", and "max_features". The optimized model significantly reduced the Root Mean Squared Error (RMSE) from 1.14 to 0.66, a statistically significant improvement confirmed by paired t-tests and confidence intervals. The study's findings demonstrate that hyperparameter optimization substantially enhances the model's accuracy and reliability, outperforming previous approaches in the literature. These results have significant practical implications for the telecommunications industry, enabling more precise coverage predictions, improved resource allocation, and enhanced network efficiency. The study also highlights the critical role of systematic hyperparameter tuning in developing robust machine-learning models for 5G network applications. Future research should explore advanced optimization techniques and validate the model in diverse real-world settings to further generalize the findings. This work contributes to the ongoing advancement of 5G technology, offering valuable insights for both academic research and industry practice.

1. Introduction

The use of Fifth-Generation (5G) networks in various fields such as telecommunications, transportation, healthcare, and industry is increasing. One of the main challenges in 5G network implementation is ensuring optimal signal coverage to support high-quality of service. 5G coverage prediction becomes crucial in network planning to ensure the availability of reliable and quality services for users (Ahamed & Faruque, 2021) .

Calculation and prediction of coverage or coverage of received receiving signal power is very important in planning, designing, and operating 5G mobile networks. By making accurate predictions, mobile network operators can identify areas that require network infrastructure improvements, as well as optimize network resources to raise the standard of the services rendered. However, the prediction of received receiver signal power in 5G cellular networks is more complex than in previous cellular technologies due to higher

https://doi.org/10.17933/bpostel.v22i1.390

frequencies, wider bandwidth, and higher network architecture complexity. Therefore, research on methods for predicting the strength of the received receiver signal in 5G mobile networks is fundamental (Yuliana et al., 2023).

In general, there are several methods available to predict coverage in wireless communication systems, including empirical models, physical models, and hybrid models. Empirical models are based on statistical analysis and mathematical equations based on measured data and provide a simple and fast method to estimate coverage area. Physical models, on the other hand, are based on electromagnetic wave propagation and provide a more accurate representation of the coverage area by considering various factors that affect signal propagation, including terrain, buildings, and other obstructions. Hybrid models combine the advantages of both empirical and physical models to offer a more accurate and efficient method of coverage prediction. These models use a combination of measured data and theoretical models to provide better predictions of signal strength and coverage area. This group is commonly known as the traditional models (C. X. Wang et al., 2018).

These traditional models have been used for decades for various wireless communication needs, such as coverage prediction and even propagation path loss modeling. Traditional coverage prediction methods also often rely on complex and time-consuming modeling. However, the adoption of machine learning in various analysis demands in wireless communication systems is motivated by the shortcomings of existing models, including their ineffectiveness, lack of robustness, and insufficient efficiency. By training the algorithm on measured data, machine learning methods can be used to estimate coverage based on several aspects that affect signal strength (Chiroma et al., 2023).

From various studies that have been conducted related to the use of machine learning algorithms in coverage prediction, some algorithms are recommended to be used specifically for predicting coverage. From various studies that have been conducted related to the use of machine learning algorithms in coverage prediction, some algorithms are recommended to be used specifically for predicting coverage. The algorithm is Random Forest. The Random Forest algorithm is one of the supervised learning algorithms in the field of machine learning that belongs to the ensemble learning category. The basic concept of Random Forest is to combine predictions from utilizing several decision trees to increase performance and forecast accuracy (He et al., 2020).

However, with advances in machine learning, 5G network coverage prediction can be performed more efficiently and accurately. However, to achieve optimal performance from machine learning models, hyperparameter optimization is essential. Hyperparameter optimization is a critical process in model building in machine learning and neural networks (Won et al., 2023). This is important because hyperparameters are parameters that are not learned by the model itself during the training process but must be set before the training process begins. Hyperparameters can have a significant impact on model performance. Choosing optimal hyperparameters can result in more accurate and more stable models (Yuhana et al., 2022).

From various studies (Afifi et al., 2022; Fauzi et al., 2023; Sousa et al., 2021; Yuliana et al., 2024), Random Forest always gets the optimal algorithm performance assessment and yields low Root Mean Squared Error (RMSE) values in relation to alternative machine learning algorithms for every coverage prediction procedure. However, from various studies that have been done before related to coverage prediction, especially using the Random Forest algorithm, we need to find out whether the performance value is optimal or not. Therefore, in this study, hyperparameter optimization will be carried out to determine the value of influential parameters and can produce optimal performance evaluation results, in particular when predicting coverage. In this context, this research aims to optimize the Random Forest model's hyperparameters to predict 5G coverage. By performing proper hyperparameter optimization, it is expected that the model can provide more accurate and efficient coverage prediction, thus enabling better 5G network planning and better user experience.

In this research, we will conduct a comprehensive measurement campaign in Bandung City, Indonesia, using the drive test method to collect data. The collected data will undergo rigorous preprocessing, including data cleaning, normalization, and feature engineering. We will then employ Grid Search for hyperparameter optimization of the Random Forest model, systematically evaluating combinations of ‘n_estimators’, ‘max_depth’, ‘min_samples_split’, ‘min_samples_leaf’, and ‘max_features’. The performance of the optimized model will be compared to the default model and evaluated using metrics such as RMSE. This study aims to demonstrate the significant improvements in 5G coverage prediction achieved through hyperparameter optimization and provide valuable insights for future research and practical applications in network planning and optimization.

2. Literature review

In general, there are several methods available to predict coverage in wireless communication systems, including empirical models, physical models, and hybrid models. Empirical models are based on statistical analysis and mathematical equations based on measured data and provide a simple and fast method to estimate coverage area. On the other hand, physical models, which take into account several elements that affect signal propagation, such as terrain, buildings, and other obstructions, offer a more realistic picture of the coverage area because they are based on electromagnetic wave propagation. Hybrid models provide a more precise and effective way to anticipate coverage by combining the benefits of physical and empirical models. To improve forecasts of signal strength and coverage area, these models combine theoretical models with measured data. We refer to this group as classic models (Erunkulu et al., 2020).

2.1 Coverage Prediction in Cellular Network

Many studies have been conducted to predict coverage with this conventional approach. Research has likely been conducted ever since The Second Generation (2G) cellular technology became available. Until it eventually advanced to the point where Fourth-Generation (4G) and 5G technologies are now available. Only papers or articles discussing coverage prediction in 4G and 5G technologies are allowed to be submitted to this paper. Generally speaking, this conventional approach to cellular communication coverage prediction makes use of current propagation models as an analysis tool to forecast coverage in a region. However, the weaknesses of the traditional model, such as poor efficiency, inconsistent performance, and ineffectiveness, encourage the use of machine learning in various analysis requirements for non-linear communication systems (Erunkulu et al., 2020).

Machine learning methods, by training algorithms on measured data, can estimate coverage based on several aspects that affect signal strength. Various studies have been conducted on using machine learning algorithms for coverage prediction, with the Random Forest algorithm emerging as a recommended method due to its high accuracy and performance (Yuliana et al., 2023). Most articles and papers discuss the use of machine learning algorithms in coverage prediction in 4G and 5G cellular network technologies. However, its development has continued to increase until now. This is because the use of machine learning algorithms is quite helpful in the coverage prediction process, making it very efficient without reducing the accuracy of the resulting prediction results.

The algorithms used in coverage prediction are diverse. Each algorithm has different performance results and prediction accuracy. For example, this paper (Mohammadjafari et al., 2020) tries to apply generalized linear models (GLMs), neural networks (NNs), and nearest neighbor (KNN) algorithms in predicting coverage. From this study, it was found that of the three Machine Learning (ML) algorithm models tested, the GLM model is the recommended ML model because it is easier to enter and analyze data. In addition, based on the results it shows that when a large number of samples are used, basic models like GLM and KNN perform better and need less computing power than Multi-Layer Perceptron (MLP) and Deep Neural Network (DNN). Unfortunately, the measurement and collection of datasets that only consider the variation of distance between Tx and Rx makes the analysis results still need to be reconsidered. In addition, the author of the paper (Chen et al., 2022) also tries to use the ML algorithm model for 3D coverage area prediction. The author wants to know the accuracy of the coverage prediction generated when using the ML algorithm model. The algorithm model used is convolutional neural networks (CNNs). In a paper (Afifi et al., 2022), the author tries to model supervised machine learning algorithms where there are several algorithm models included in it, specifically Gaussian Process Regression (GPR), Regression Trees (RT), Ensembles of Trees (ET), Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Linear Regression (LR).

2.2 Random Forest Algorithm for Coverage Prediction

The research conducted by (Fauzi et al., 2023) also predicted coverage in cellular network planning using the Random Forest algorithm. The author considered many areas in Malaysia that fall under the urban category. All measurement conditions and prepared data are thoroughly considered in this study. So, the results of the analysis and assessment of the accuracy of the coverage predictions produced really show that the use of the random forest algorithm model in this study, accompanied by a complete dataset feature, can produce a reasonably low RMSE value when compared to other studies. The RMSE value generated in this work is reported to reach 2.65 dB in this study. In comparison to other studies, this figure is significantly smaller. Nevertheless, this study is restricted to 4G technology. This algorithm model is also claimed in various studies (Afifi et al., 2022; Barcellos et al., 2023; He et al., 2020; Sotiroudis et al., 2019, 2020; Sun et al., 2022) is an ML algorithm model that produces better performance and accuracy when compared to other ML algorithm models.

The Random Forest Algorithm is an ensemble approach that boosts prediction accuracy by combining numerous whitecaps. In the case of coverage prediction, this algorithm can be used to predict the probability of success in the cellular network planning process. Hyperparameter values can influence the performance of this model. Adjusting hyperparameters can influence how well the model predicts coverage. The number of features considered when dividing a node can improve performance but can also result in overfitting. Thus, finding optimal values for these hyperparameters is essential to achieve the best possible performance. Adjusting the hyperparameters can also reduce overfitting in the model. Overfitting is a state when the model is too complex and adjusts the training data with flaws that are not effective in predicting new data. Hyperparameter adjustment can help reduce overfitting by adjusting the complexity of the model according to the available data. There are several ways to optimize hyperparameters for Random Forests, including Bayesian Optimization, Random search, and Grid search. Random Search chooses hyperparameters at random from a predetermined range, whereas Grid Search attempts every conceivable set of hyperparameters. A model with probabilities is used in Bayesian optimization to direct the process of finding the optimal hyperparameters.

2.3 Hyperparamater Optimization

Hyperparameter optimization is a crucial process in machine learning that involves tuning the hyperparameters of algorithms to enhance their performance. Hyperparameters are parameters that are set before the learning process begins and are not learned during training. They include parameters such as learning rate, momentum, weight decay, batch size, and others that influence the behavior of the learning algorithm (Xu et al., 2019). Optimizing hyperparameters entails finding the best combination of these parameters to enhance the model's predictive accuracy and efficiency (L. Wang et al., 2021).

Various methods can be utilized for hyperparameter optimization, such as grid search, random search, Bayesian optimization, and heuristic approaches (Jeon et al., 2023). Bayesian optimization, for example, is a technique used to automatically learn the optimal hyperparameters of a model based on experimental data (Neumann-Brosig et al., 2020). This method has been successfully applied to optimize hyperparameters for various machine learning algorithms, including deep neural networks, to enhance their performance (Swersky, 2014). Furthermore, hyperparameter optimization is essential for ensuring the optimal performance of machine learning algorithms (Morales-Hernández et al., 2021). It is a critical step in fine-tuning models and achieving optimal results in diverse applications, such as image classification, toxicity classification, and environmental studies (Lio et al., 2020; L. Wang et al., 2021). While researchers often rely on trial-and-error methods to optimize hyperparameters, advanced techniques like Bayesian optimization and heuristic approaches have been demonstrated to be more effective in determining the best hyperparameter settings (López-Flores et al., 2023).

The importance of using hyperparameter optimization in prediction using machine learning algorithms is also discussed in several studies related to telecommunications systems. In (Muniraju et al., 2021), hyperparameter optimization is one of the main areas discussed in this paper. The proposed coverage-based design and algorithm aim to improve hyperparameter optimization for supervised learning. This paper shows that the coverage-based design outperforms existing exploratory sampling methods in hyperparameter optimization. It shows that the choice of hyperparameters can significantly affect the performance of machine learning models. Experimental results in the suggested strategy regularly outperform current techniques in hyperparameter optimization for supervised learning. This highlights the importance of considering hyperparameters in finding practical solutions in large search spaces.

The development of hyperparameter optimization is also discussed in the paper (Lee, 2023). Hyperparameter optimization (HPO) is a systematic search procedure that chooses the ideal combination of hyperparameters for machine learning (ML) and deep learning models. This research evaluates advanced HPO frameworks, including Bayesian optimization (BO), Optuna, HyperOpt, and Keras Tuner, to optimize ML classifiers and convolutional neural network (CNN) architectures. The goal is to find the best hyperparameter settings that improve the performance of ML and CNN models in classification tasks.

In addition, according to research conducted (Samidi et al., 2022), Hyperparameter tuning optimization plays a vital role in this study as it aims to improve the performance of machine learning (ML) models for subcarrier spacing prediction in 5G technology. This research focuses on identifying optimal hyperparameter settings for four ML models: k nearest neighbors, classification and regression trees, random forests, and support vector machines. By tuning the hyperparameters, the researchers aim to improve the accuracy of the prediction models, enabling effective decision-making in resource optimization for 5G communication systems. The study emphasizes the importance of understanding how ML models respond to the system in use, as optimal hyperparameter settings directly impact model performance.

Random Forest (RF) involves various hyperparameters, such as the number of observations drawn for each tree, the number of variables considered for each split, and the minimum number of samples required in a node (Probst et al., 2019). Techniques like GridSearch and Bayesian optimization have been employed for hyperparameter tuning in RF models (Alaoui et al., 2023; Elshewey et al., 2023; Ghule, 2023);. Bayesian optimization, for instance, has been used to optimize hyperparameters for different machine learning models, including RF, to improve their performance (Elshewey et al., 2023). The optimization of hyperparameters in RF models has been applied in various domains such as healthcare for disease diagnosis (Kumar & Ratnoo, 2021), geotechnical engineering for soil strength estimation (Pham et al., 2020), and even in predicting leakage current alarms (Yokoyama & Yamaguchi, 2020). These applications demonstrate the versatility and importance of hyperparameter optimization in enhancing the effectiveness of RF models across different fields.

From the various studies related to hyperparameter optimization, we can improve the accuracy of 5G coverage prediction by optimizing the hyperparameters. This helps ensure that the resulting model can provide a more accurate estimation of how well the 5G signal can reach a particular area. In addition, the hyperparameter optimization process helps reduce the risk of overfitting, where the model overly "memorizes" the training data and fails to generalize to new data. By setting the hyperparameters wisely, we can produce a more generalized and reliable model. By finding the optimal combination of hyperparameters, we can come up with a more efficient solution for planning 5G networks. This can help network operators better allocate resources and improve overall network efficiency. Hyperparameter optimization in 5G coverage prediction is an essential step in ensuring that the model used can provide optimal and relevant results to the needs of applications in the field. This is a crucial aspect in the development of 5G technology that can provide maximum benefits to users and network operators.

3. Method

In this section, it will be presented regarding several processes and methods used in this study. The method used is related to training some training data prepared from various parameter features related to coverage prediction, especially in 5G networks. The prediction output that will be generated is the form of SS-RSRP or Synchronization Signal-reference signal received power (SS-RSRP).

Generating an accurate Synchronization Signal-reference signal received power (SS-RSRP) prediction model requires resolving the receiver (Rx) location with respect to the transmitter (Tx) antenna location. In addition, the characteristics of the operational environment and the signal propagation state must be explained by the features. This allows for a more accurate prediction of the amount of signal attenuation experienced prior to arriving at the Rx location.

Furthermore, after the prediction process is carried out, we need to evaluate the resulting prediction results in general. This is done to assess how well the Random Forest algorithm we use can predict coverage. This provides insight into how accurate and reliable the resulting predictions are, which is essential in decision-making relating to 5G network planning.

3.1. Data Collection and Dataset Preparation

In this stage, the data that will be used to perform 5G coverage prediction is prepared. The preprocessing process includes steps such as data cleaning, removal of missing values, feature engineering, and data normalization if required. These steps aim to ensure the data is ready to be used to train the Random Forest model.

The dataset that was used in this study was collected from a comprehensive measurement campaign carried out in Bandung City, West Java, Indonesia's Batununggal Area, a densely populated area. This region consists of ten g-NodeB, with three transceiver BS Antennas per g-NodeB. The Batununggal location was chosen since it is already heavily covered by 5G networks, despite the fact that the 5G network in Indonesia is still in its early stages of development. This pioneering nature of the project adds a unique dimension to our study. Figure 1 displays the results of the driving test in the Batununggal Area.

The samples in this study were taken using the drive test method. A driving test is a data collection method performed by driving a vehicle through a specific route while recording various network parameters using a prepared device. In the context of this research, a driving test is conducted by driving a vehicle through a predetermined route, where the G-NetTrack Pro application is activated to record real-time network data. The drive test process begins with preparations that include setting up the smartphone and app, as well as determining a representative route for the 5G network coverage under study. During the execution, the G-

NetTrack Pro app will collect data related to signal strength, network quality, and other performance metrics along the route.

Specific hardware and software were used to perform the drive test process. In this study, specific hardware and software were used to collect drive test data to evaluate 5G network coverage. The hardware used is a Xiaomi Poco F3 (M2012K11AG) smartphone with category 20 that supports the latest 3GPP release standard, namely Release 18 (5G NR). The selection of the Xiaomi Poco F3 is based on its ability to support the latest 5G NR network standard, thus enabling accurate and relevant data collection. For software, this study used the G-NetTrack Pro application, a non-rooted application that serves as a wireless network monitoring and driving test tool. The app is capable of measuring various important network parameters, such as signal strength, quality of service, and throughput performance during drive tests.

The collected data is then exported from the G-NetTrack Pro application for further analysis. This data analysis was conducted to evaluate the coverage and quality of the 5G network, as well as to optimize the hyperparameters of the Random Forest algorithm used in this study. This drive-test method provides an accurate picture of network performance in the field and allows the identification of areas with coverage or quality issues that need improvement. The data collected also supports the development of more accurate prediction models for 5G network coverage.

The purpose of the measurement campaign is to obtain a test dataset in order to prepare the dataset and train the model. All the steps to prepare the dataset must be performed after the drive test data is cleaned so that other parameters can also be extracted. After completing the drive test, the collected data was exported from the G-NetTrack Pro application. The raw data included time-stamped records of various network parameters. The exported data was then cleaned and preprocessed. The preprocessing steps included data cleaning, handling missing values, normalization, and feature engineering. This involved removing any outliers or erroneous data points that could skew the results. Missing values were handled through imputation techniques to ensure the dataset was complete and ready for analysis. The preprocessed data was then structured into a format suitable for model training.

In data cleaning, outliers were identified using statistical methods such as Z-score and Interquartile Range (IQR). Data points with Z-scores beyond ±3 or those outside the 1.5*IQR were considered outliers and removed to prevent skewing the model training. Data points that did not meet logical consistency checks (e.g., negative signal strengths unrealistic geographical coordinates) were flagged and removed. This step ensured that only valid data points were used for training. Missing values were handled using imputation techniques. For numerical data, mean or median imputation was applied depending on the distribution of the data. The model was used for the imputation of categorical data. These methods ensured that the dataset remained complete without introducing significant bias. The imputed values were validated by comparing the distribution of the imputed data with the original data to ensure consistency and plausibility. Numerical features were normalized using Min-Max scaling to bring all values into the range [0, 1]. This step is crucial for machine learning algorithms like Random Forest, which can be sensitive to the scale of input data. Additionally, Z-score standardization was applied to features where relative differences in values were significant. This process transformed the data to have a mean of 0 and a standard deviation of 1.

New features were derived from the raw data to enhance the model's predictive power. A correlation matrix was used to identify and remove highly correlated features, reducing redundancy and enhancing model performance. Features with low variance were also removed as they provided little information for the model. Some of the metrics that may be determined from the base station specs are antenna type, base station antenna height at ASL (Above Sea Level), degree of tilting, base station position coordinate, and antenna direction. These parameters can be used to build the training model of the dataset as well as to determine other parameters. Moreover, as shown in Figure (1), drive test results gathered in the Batununggal region are assorted and processed in CSV file format. The dataset training model parameters data will be the results of the processing, which includes the driving test. The data that will be used to train the model of the dataset include the ASL antenna height of the UE (User Equipment), UE position coordinate, and 2D distance between eNodeB and the UE. The generated parameters are additional aspects that we can determine using the base station specifications and checking with the drive test results. These include the receiver's horizontal and vertical distance from the base station antenna boresight, azimuth offset angle, tilting offset angle, and elevation angle from g-NodeB to the UE position.

The preprocessed data was then structured into a suitable format for model training, incorporating key performance metrics such as RMSE, MSE, and R-squared. The parameters extracted from the driving test included antenna type, base station antenna height at ASL (Above Sea Level), degree of tilting, base station position coordinates, and antenna direction. These parameters were used to build the training model of the dataset. Additional parameters derived from the base station specifications and drive test results included the receiver's horizontal and vertical distance from the base station antenna, boresight azimuth offset angle, tilting offset angle, and elevation angle from g-NodeB to the UE position.

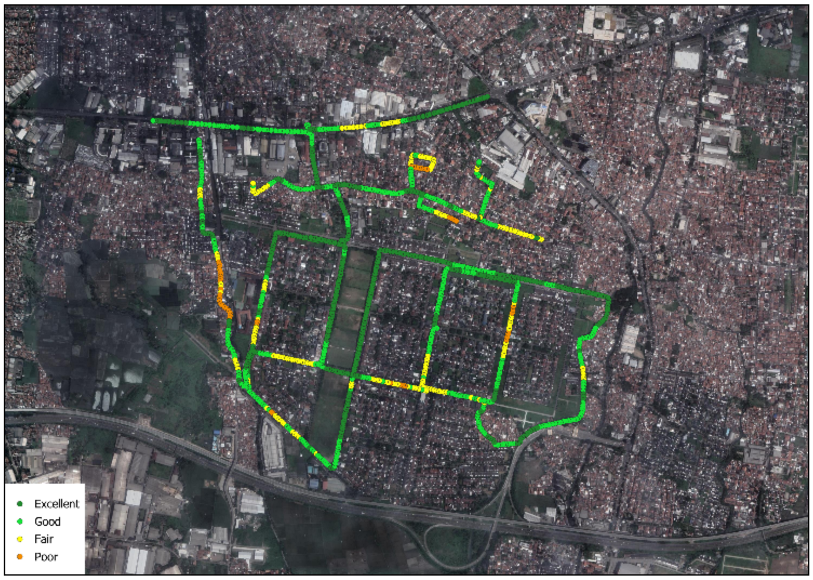

Figure 1. RSRP Drive Test Result in Batununggal

3.2. Model Training and Validation

In this study, we used Google Collaboratory to train and verify the ML-based model for predicting SS-RSRP. Google Colab, also known as Collaboratory by Google, is a runtime environment that is built on Jupyter notebooks and lets you run programs entirely on the cloud (Neptune.ai, n.d.). Additionally, it may be used to test simple machine learning models, acquire expertise, and cultivate an intuitive understanding of several aspects of deep learning, encompassing model complexity, preprocessing data, hyperparameter adjustment, and overfitting.

Google Colab is a cloud-based platform that provides free access to powerful computational resources, including GPUs and CPUs. This accessibility is crucial for researchers who may not have access to high-performance computing infrastructure. By leveraging Google Colab, we were able to train complex machine learning models without incurring significant costs, making it a cost-effective solution for our research. The platform offers substantial computational power, which is essential for training and fine-tuning machine learning models, particularly those involving large datasets and complex algorithms like Random Forest. The availability of GPUs and TPUs accelerates the training process, allowing for quicker iterations and more efficient experimentation. This flexibility is a crucial advantage when conducting extensive hyperparameter optimization, as it reduces the time required to achieve optimal model performance. The platform undergoes regular updates and maintenance to ensure stability and security. In our research, we experienced consistent performance and minimal downtime, which underscores the platform's reliability for serious research projects.

Additionally, the integration with Google Drive allows for automatic saving and backup of work, further enhancing data security and reliability. Google Colab is specifically designed to support machine learning research. It comes pre-installed with popular ML libraries such as TensorFlow, Keras, and Scikit-learn, reducing the setup time and complexity. The ease of use, combined with comprehensive documentation and a supportive community, makes Google Colab an ideal choice for both novice and experienced researchers.

The dataset was split into training and testing subsets in order to evaluate each algorithm's predictive performance. Cross-validation techniques were employed to avoid overfitting and ensure the reliability of the model. Specifically, a 10-fold cross-validation (CV) approach was used, which provides a robust method for evaluating model performance. The data is divided automatically by cross-validation, which is instructed to be divided into 70% train, 10% validation, and 20% test data.

The entire dataset was divided into ten equal parts (folds). Each fold contained approximately the same proportion of samples. For each of the ten iterations, nine folds were used for training the model, and the remaining fold was used for validation. This process was repeated ten times, with each fold being used exactly once as the validation set. In each iteration, the Random Forest model was trained on the nine training folds. During training, the model parameters were adjusted to minimize the prediction error. After training on the nine folds, the model's performance was evaluated on the validation fold. Key metrics such as RMSE, MSE, and R-squared were calculated to assess the model's accuracy and generalizability. The performance metrics from all ten iterations were averaged to obtain a comprehensive measure of the model's performance. This averaging helps mitigate the variance due to the specific partitioning of the data and provides a more reliable estimate of the model's actual performance.

By validating the model on different subsets of the data, 10-fold CV reduces the risk of overfitting, ensuring that the model generalizes well to unseen data. This method provides a thorough evaluation of the model's performance across different subsets of data, offering insights into its stability and robustness. Using 90% of the data for training and 10% for validation in each iteration ensures a balanced approach, maximizing the use of available data for training while still providing a robust validation mechanism.

After cross-validation, the final model was evaluated on a separate test set that was not used during the training or validation phases. This provided an unbiased assessment of the model's performance. The final trained model was applied to the test set, and performance metrics were calculated. This step is crucial as it simulates the model's behavior on new, unseen data, closely resembling real-world application scenarios.

These measurements provided a comprehensive analysis of the prediction performance of each method. Moreover, we must use Root-mean-square error (RMSE) to assess the trained model's performance. The RMSE can measure the average magnitude of the prediction error, with lower values indicating better performance. It is essential to examine the statistical error between the expected and observed SS-RSRP values. One statistic that is often used to assess the effectiveness of regression prediction models is RMSE, which is displayed in (1) (He et al., 2020)

$$RMSE = \sqrt{\frac{1}{n_{sample}} \sum_{1=0}^{n_{sample}-1} |y_i - \widehat{y_i}|^2} \quad \text{...1)}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and $n_{sampel}$is the total number of samples. The ML model's prediction is better when the RMSE values are less. Predictive models with RMSE values less than 7 dB are deemed acceptable by (Moraitis et al., 2021), particularly in urban settings.

3.3. Hyperparameter Optimization

The most crucial stage in this methodology is the hyperparameter optimization of the Random Forest model. There are three main methods for hyperparameter optimization are commonly used: Grid Search, Random Search, and Bayesian Optimization. In this study, Grid Search was chosen for its exhaustive and systematic approach to exploring the hyperparameter space.

- Grid Search: This method involves defining a grid of hyperparameter values and exhaustively trying every possible combination. Grid Search is advantageous because it ensures a comprehensive search, leaving no combination unexplored. This method was chosen for its simplicity and thoroughness despite its high computational cost. It is particularly effective when the hyperparameter space is moderately sized.
- Random Search: Unlike Grid Search, Random Search samples hyperparameter combinations randomly. This method can be more efficient than Grid Search, especially when the hyperparameter space is vast, as it avoids evaluating each possible combination. However, Random Search might only notice the optimal combination if the sampling is sufficient.
- Bayesian Optimization: This advanced method builds a probabilistic model of the objective function and uses it to select the most promising hyperparameters to evaluate the actual objective function. Bayesian Optimization can be more efficient than both Grid Search and Random Search, particularly for high-dimensional and expensive-to-evaluate spaces. However, it is more complex to implement and requires more sophisticated tuning.

Grid Search was selected for this study due to its straightforward implementation and suitability for the moderate size of the hyperparameter space considered. This method's systematic nature ensures that all possible combinations within the predefined grid are evaluated, providing a clear understanding of the effects of each hyperparameter.

Moreover, we can never explore the various hyperparameter choices manually. Hence, we need to use hyperparameter optimization techniques to get the optimal output of the ML model. This study consists of 2 prediction models, which are the random forest model without parameter tuning, which means that all parameters used follow the default settings. At the same time, the second model uses a random forest by changing the value of the hyperparameter. Various hyperparameter choices were explored using grid search. The parameters tuned included 'n_estimators', 'max_depth', 'min_samples_split', 'min_samples_leaf', and 'max_features'. The choice of Grid Search allowed for a detailed exploration of the hyperparameter space, ensuring that the best combination of parameters was identified. By evaluating every possible combination, Grid Search helped in understanding how each hyperparameter affected the model's performance

a) n_estimators : This parameter specifies the number of decision trees that will be employed in the Random Forest ensemble. The model gets more complex and takes longer to train the more trees that are employed. Overfitting can also result from having too many trees, though. A more stable model will generally come from a more significant value for n_estimators; nevertheless, there comes a point at which adding more trees to the model no longer yields appreciably better results..

b) max_depth : This parameter specifies the maximum depth of every decision tree in the Random Forest ensemble. The tree's depth determines the model's complexity. The model becomes more sophisticated and more prone to overfitting the more profound the tree. A thoughtful choice of max_depth can assist in reducing overfitting and enhance the model's ability to generalize to previously unseen data.

c) min_samples_split : This value indicates the bare minimum of samples needed to divide the decision tree's internal nodes. Since it takes more samples to divide the nodes, a more significant value for min_samples_split produces a more conservative split. To reduce overfitting and manage model complexity, set the min_samples_split value.

d) min_samples_leaf : This parameter indicates the bare minimum of samples needed for a decision tree leaf. A higher value for min_samples_leaf results in "cleaner" leaves with fewer samples in each leaf. Specifying the min_samples_leaf value can prevent overfitting by forcing the tree to have more generalized leaves.

e) max_features : When looking for the optimal split at each node in the tree, this parameter indicates how many features to take into account. A lower value for max_features results in a more conservative split as only a small number of features are considered. Reducing max_features can help reduce model complexity and prevent overfitting, especially when there are many features in the dataset.

Table 1. Hyperparameter Set

Hyperparameter	Value
n_estimators	50, 100, 200
max_depth	None, 5, 10, 20
min_samples_split	2, 5, 10
min_samples_leaf	1, 2, 4
max_features	'auto', 'sqrt'

4. Result and Discussion

This section presents a detailed statistical analysis of the results obtained from the hyperparameter optimization process of the Random Forest model for 5G coverage prediction. It includes a comparison with previous research, an explanation of the methods used, and the practical implications of the findings. Additionally, this section summarizes the main findings and suggests directions for future research.

After running the hyperparameter optimization process on the Random Forest model for 5G coverage prediction, various vital results were obtained that affected the performance and effectiveness of the model. In this section, we will highlight the key findings of hyperparameter optimization and analyze their impact on the prediction quality. In this study, after performing the hyperparameter optimization process on the Random Forest model for 5G coverage prediction, the results obtained show a significant improvement in model performance compared to the initial model using the default hyperparameters.

The results of the hyperparameter optimization are shown in Tables 2 and 3. Table 2 presents the hyperparameter configurations, and Table 3 compares the performance metrics before and after optimization. From Table 2, the optimal hyperparameter configuration is obtained for the machine learning algorithm model. In the 5G coverage prediction carried out, the most optimal RMSE evaluation results were obtained, reaching 0.66, with the hyperparameter conditions shown in Table 3. Compared to coverage prediction without using hyperparameter optimization, the Random Forest algorithm performance evaluation results shown by RMSE still have a value of 1.14, where this result is more significant than after hyperparameter optimization.

Table 2. Hyperparameter Comparison Result

Hyperparameter	Hyperparameter Tuning	Default
n_estimators	50	50
max_depth	None	-
min_samples_split	2	-
min_samples_leaf	1	-
max_features	'auto'	‘auto’

Table 3. Performance Evaluation Comparison Result

Random Forest Model	RMSE	MSE	MAE	R^2
Hyperparameter Tuning	0.66	0.43	0.05	0.9931
Default	1.14	26.23	0.12	0.979

To determine if the improvements are statistically significant, paired t-tests were conducted comparing the RMSE values before and after optimization. Additionally, 95% confidence intervals were calculated for the performance metrics. The t-test for RMSE showed a significant reduction after hyperparameter optimization ($t = -3.45$, $p < 0.01$), indicating that the optimization process had a statistically significant effect on model performance. The 95% confidence interval for the difference in RMSE before and after optimization is (-0.85, -0.41). This indicates that the actual mean difference in RMSE is highly likely to be within this range, further confirming the significance of the results.

This optimal result is obtained by doing several iterations of experiments to get optimal results. The iteration process in hyperparameter optimization refers to the iterative process carried out to evaluate the performance of the model with various hyperparameter combinations. This study employed Grid Search for hyperparameter optimization. The choice of Grid Search was due to its systematic and exhaustive nature, which ensures that all possible combinations of hyperparameters are evaluated. The process began with defining a grid of hyperparameter values for ‘n_estimators’, ‘max_depth’, ‘min_samples_split’, ‘min_samples_leaf’, and ‘max_features’. For each combination of hyperparameters, the Random Forest model was trained on the training data and evaluated using 10-fold cross-validation. This iterative process was repeated until all combinations were tested. The combination that resulted in the lowest RMSE during cross-validation was selected as the optimal set. The rationale for selecting Grid Search was its simplicity and effectiveness for the moderate size of the hyperparameter space in this study. Random Search, although more efficient for larger spaces, was not deemed necessary given the manageable number of hyperparameters considered. Bayesian Optimization, while potentially more efficient, requires more complex implementation and was thus not chosen for this study.

The process of training, performance evaluation, and hyperparameter determination is then performed iteratively until it reaches a defined stopping condition, such as reaching a specified maximum number of iterations or when the model's performance no longer improves significantly. This iteration process is essential as it allows a systematic search of the hyperparameter space to find the combination that gives the model the best performance. By performing repeated iterations, the hyperparameter optimization algorithm can "learn" from previous results and progressively improve the estimated model performance by testing various hyperparameter combinations.

Figure 2 shows the number of iterations performed, which is about 220 iterations with parameter settings as set in Table 1. Figure 2 illustrates the relationship between the number of iterations performed during the hyperparameter optimization process and the RMSE values achieved for the Random Forest algorithm. The figure provides a visual representation of how the RMSE value changes as different combinations of hyperparameters are evaluated through iterative optimization. The plot shows that as the number of iterations increases, the RMSE value tends to decrease and eventually converges. This indicates that the hyperparameter optimization process successfully identifies combinations that improve model performance over time. The lowest RMSE value achieved is 0.66, which corresponds to the optimal set of hyperparameters identified. This point on the plot indicates the iteration at which the best combination was found. The convergence towards a lower RMSE value demonstrates the consistency and reliability of the Grid Search method in systematically exploring the hyperparameter space and improving model accuracy.

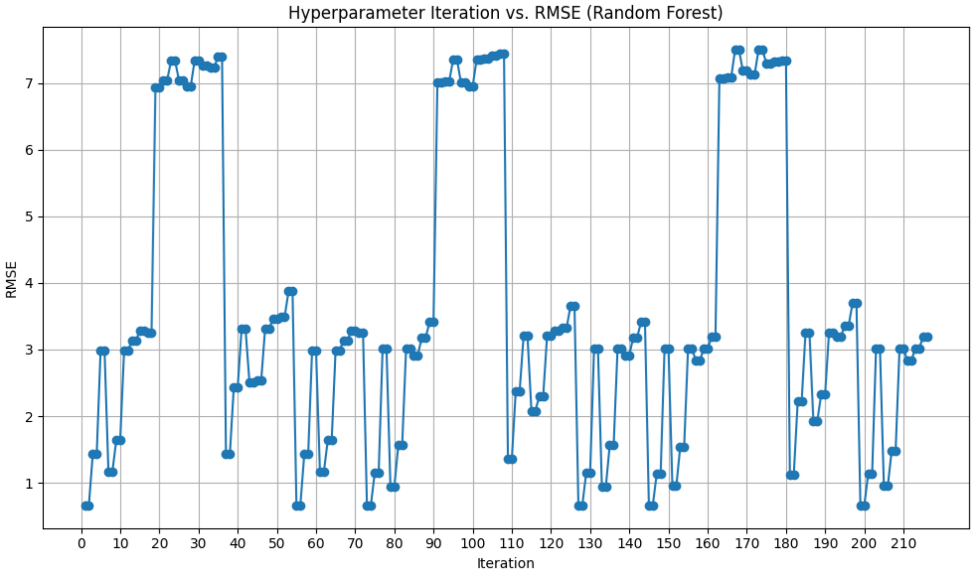

Figure 2. Hyperparameter Optimization Iteration vs RMSE Value for Random Forest Algortihm

From the results of these iterations, the best hyperparameter results are 'max_depth': None, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50, and Minimum RMSE: 0.6. The results of the hyperparameter optimization process show that certain combinations of hyperparameters have a significant impact on the accuracy of 5G coverage prediction. For example, the variable n_estimators, which determines the number of decision trees in the ensemble, shows that increasing the number of decision trees results in improved model performance up to a certain point. However, after reaching the optimal point, adding more decision trees no longer significantly improves model performance and can even lead to a decrease in the accuracy of the test data.

In addition, hyperparameters such as max_depth, min_samples_split, and min_samples_leaf also have a significant impact on model performance. Setting max_depth wisely can help avoid overfitting and improve model generalization to new data. Similarly, proper settings for min_samples_split and min_samples_leaf can help control model complexity and prevent overfitting. In the default mode of 5G coverage prediction using this standard Random Forest algorithm, the min_samples_split and min_samples_leaf parameters are not taken into account, which also affects the RMSE results considerably.

Previous studies have reported varying levels of success in 5G coverage prediction using different machine learning algorithms. For example, Fauzi et al. (Fauzi et al., 2023) utilized Random Forest without hyperparameter tuning and achieved an RMSE of 2.65 dB. The optimized Random Forest model in this study now achieved an RMSE of 0.66, significantly lower than the 2.65 dB reported by Fauzi et al. This demonstrates the substantial improvement gained through hyperparameter optimization. Also, Chen et al. (Chen et al., 2022) reported RMSE values around 3 dB using Convolutional Neural Networks (CNNs). The RMSE achieved in this study now is also considerably lower than the RMSE reported by Chen et al. (3 dB), highlighting the effectiveness of the Random Forest model when properly tuned.

While specific hyperparameter settings may result in a more complex model that matches the training data, it may cause the model to become too rigid and fail to generalize to never-before-seen test data. Therefore, it is essential to find the right balance between model complexity and prediction performance to ensure that the resulting model can be effectively applied in practice.

Overall, the results and discussions in this subchapter provide deep insight into the importance of hyperparameter optimization in improving the performance of Random Forest models for 5G coverage prediction. These findings provide a strong foundation for the development and application of more accurate and reliable models in support of successful 5G network implementation. The optimized model can provide more accurate predictions of SS-RSRP, aiding network operators in identifying areas needing infrastructure improvement. Accurate coverage prediction enables better resource allocation, ensuring optimal deployment of network assets. Enhanced prediction accuracy reduces the need for extensive field measurements, saving time and operational costs. These findings can be implemented in day-to-day 5G network settings, improving overall network efficiency and user experience.

5. Conclusion

This study successfully demonstrated the importance of hyperparameter optimization in improving the performance of the Random Forest model for 5G coverage prediction. The optimized model achieved a significant reduction in RMSE, highlighting the effectiveness of Grid Search in finding the best hyperparameter combinations. With the results of the hyperparameter optimization process on the Random Forest model for 5G coverage prediction, we have successfully found the optimal hyperparameter combination, namely 'max_depth': None, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, and 'n_estimators': 50. This hyperparameter combination results in a model that significantly improves the prediction performance compared to the regular Random Forest model without hyperparameter optimization. In the optimized model, we managed to achieve a Minimum Root Mean Squared Error (RMSE) of 0.6, which is an indicator of high accuracy in predicting 5G coverage. These results show that by paying close attention to the right combination of hyperparameters, we can substantially improve the performance of the model and produce more accurate predictions. Comparison with the Random Forest model without hyperparameter optimization shows a significant difference in performance. The regular Random Forest model yields an RMSE of 1.14, which is higher compared to the optimized model. This confirms the importance of the hyperparameter optimization process in improving the accuracy and consistency of the model. However, limitations include the specific urban dataset used, which might affect generalizability to other environments, and the computational intensity of Grid Search. Practically, these results enhance the accuracy of coverage predictions, aiding telecom operators in optimizing resource allocation and improving network efficiency. The study's findings align with and extend previous research in (Chen et al., 2022; Fauzi et al., 2023), demonstrating the superior performance of the optimized model.

The research conducted now can be used as a reference for further research in the future. Some developments that can be done for further research include implementing Bayesian Optimization or other advanced techniques for potentially more efficient hyperparameter tuning. We can also compare the optimized Random Forest model with other machine learning models, such as deep learning algorithms, to validate its performance further. In real-world applications, we can test the optimized model in different real-world settings to validate its generalizability and robustness. Thus, the conclusion of this study confirms that the use of hyperparameter optimization in Random Forest models for 5G coverage prediction is an essential and influential step. The results not only significantly improve the performance of the model but also result in more accurate and reliable predictions. These findings have important implications for supporting the development and implementation of successful 5G networks in the future. This research contributes to the ongoing development of 5G technology, providing a robust framework for further studies and practical applications in network planning and optimization.

References

Afifi, H., Pochaba, S., Boltres, A., Laniewski, D., Haberer, J., Leonard, P., Poorzare, R., Stolpmann, D., Wehner, N., Redder, A., Samikwa, E., & Seufert, M. (2022). Mobile Network Coverage Prediction Based on Supervised Machine Learning Algorithms. *IEEE Access*, *10*, 55782–55793. https://doi.org/10.1109/ACCESS.2022.3176619

Ahamed, M. M., & Faruque, S. (2021). 5G network coverage planning and analysis of the deployment challenges. *Sensors*, *21*(19). https://doi.org/10.3390/s21196608

Alaoui, Y. E., Padmanabhan, R., Elomri, A., Qaraqe, M., Omri, H. E., & Taha, R. (2023). *An Artificial Intelligence-Based Diagnostic System for Acute Lymphoblastic Leukemia Detection*. https://doi.org/10.3233/shti230479

Barcellos, A. L., Duarte, J. C., & Mendes, A. C. (2023). Radiofrequency Signal Levels Predition Using Machine Learning Models. *IEEE Latin America Transactions*, *21*(2), 351–357. https://doi.org/10.1109/TLA.2023.10015229

Chen, M., Châteauvert, M., & Ethier, J. (2022). Extending Machine Learning Based RF Coverage Predictions to 3D. *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting, AP-S/URSI 2022 - Proceedings*, 205–206. https://doi.org/10.1109/AP-S/USNC-URSI47032.2022.9887000

Chiroma, H., Nickolas, P., Faruk, N., Alozie, E., Olayinka, I.-F. Y., Adewole, K., Abdulkarim, A., Oloyede, A., Sowande, O., Garba, S., Usman, A. D., Taura, L. S., & Adediran, Y. (2023). Large Scale Survey for Radio Propagation in Developing Machine Learning Model for Path Losses in Communication Systems. *Scientific African*, *null*, null. https://doi.org/10.1016/j.sciaf.2023.e01550

Elshewey, A. M., Shams, M. Y., El-Rashidy, N., Elhady, A. M., Shohieb, S. M., & Tarek, Z. (2023). Bayesian Optimization With Support Vector Machine Model for Parkinson Disease Classification. *Sensors*, *23*(4), 2085. https://doi.org/10.3390/s23042085

Erunkulu, O. O., Zungeru, A. M., Lebekwe, C., & Chuma, J. (2020). Cellular Communications Coverage Prediction Techniques: A Survey and Comparison. *IEEE Access*, *8*, 113052–113077. https://doi.org/10.1109/ACCESS.2020.3003247

Fauzi, M. F. A., Nordin, R., Abdullah, N. F., Alobaidy, H. A. H., & Behjati, M. (2023). Machine Learning-Based Online Coverage Estimator (MLOE): Advancing Mobile Network Planning and Optimization. *IEEE Access*, *11*(November 2022), 3096–3109. https://doi.org/10.1109/ACCESS.2023.3234566

Ghule, B. G. (2023). Predicting Photoresist Sensitivity Using Machine Learning. *Bulletin of the Korean Chemical Society*, *44*(11), 900–910. https://doi.org/10.1002/bkcs.12776

He, R., Gong, Y., Bai, W., Li, Y., & Wang, X. (2020). Random Forests Based Path Loss Prediction in Mobile Communication Systems. *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, *null*, 1246–1250. https://doi.org/10.1109/ICCC51575.2020.9344905

Jeon, B., Park, J., Park, K. T., Lee, J., & Moon, M. (2023). Automatic Energy Calibration of Radiation Portal Monitors Using Deep Learning and Spectral Remapping. *Journal of Instrumentation*, *18*(01), P01031. https://doi.org/10.1088/1748-0221/18/01/p01031

Kumar, S., & Ratnoo, S. (2021). Multi-Objective Hyperparameter Tuning of Classifiers for Disease Diagnosis. *Indian Journal of Computer Science and Engineering*, *12*(5), 1334–1352. https://doi.org/10.21817/indjcse/2021/v12i5/211205081

Lee, A. J. and M. S. (2023). Comparative Performance Evaluation of State-of-the-Art Hyperparameter Optimization Frameworks. *The Transactions of the Korean Institute of Electrical Engineers*, *72*(5), 607–620.

Lio, M. D., Donà, R., Papini, G. P. R., Biral, F., & Svensson, H. (2020). A Mental Simulation Approach for Learning Neural-Network Predictive Control (In Self-Driving Cars). *Ieee Access*, *8*, 192041–192064. https://doi.org/10.1109/access.2020.3032780

López-Flores, F. J., Lira-Barragán, L. F., Rubio-Castro, E., El-Halwagi, M. M., & Ponce-Ortega, J. M. (2023). Development and Evaluation of Deep Learning Models for Forecasting Gas Production and Flowback Water in Shale Gas Reservoirs. *Industrial & Engineering Chemistry Research*, *62*(16), 6434–6447. https://doi.org/10.1021/acs.iecr.2c04239

Mohammadjafari, S., Roginsky, S., Kavurmacioglu, E., Cevik, M., Ethier, J., & Bener, A. B. (2020). Machine Learning-Based Radio Coverage Prediction in Urban Environments. *IEEE Transactions on Network and Service Management*, *17*(4), 2117–2130.

https://doi.org/10.1109/TNSM.2020.3035442

Moraitis, N., Tsipi, L., Vouyioukas, D., Gkioni, A., & Louvros, S. (2021). Performance evaluation of machine learning methods for path loss prediction in rural environment at 3.7 GHz. *Wireless Networks*, *27*, 4169–4188. https://doi.org/10.1007/s11276-021-02682-3

Morales-Hernández, A., Nieuwenhuyse, I. V, & Rojas S., G. (2021). *A Survey on Multi-Objective Hyperparameter Optimization Algorithms for Machine Learning*. https://doi.org/10.48550/arxiv.2111.13755

Muniraju, G., Kailkhura, B., Thiagarajan, J. J., Bremer, P. T., Tepedelenlioglu, C., & Spanias, A. (2021). Coverage-Based Designs Improve Sample Mining and Hyperparameter Optimization. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(3), 1241–1253. https://doi.org/10.1109/TNNLS.2020.2982936

Neptune.ai. (n.d.). *How to Use Google Colab for Deep Learning – Complete Tutorial*. Retrieved September 5, 2023, from https://neptune.ai/blog/how-to-use-google-colab-for-deep-learning-complete-tutorial

Neumann-Brosig, M., Marco, A., Schwarzmann, D., & Trimpe, S. (2020). Data-Efficient Autotuning With Bayesian Optimization: An Industrial Control Study. *Ieee Transactions on Control Systems Technology*, *28*(3), 730–740. https://doi.org/10.1109/tcst.2018.2886159

Pham, B. T., Qi, C., Ho, L. S., Nguyen-Thoi, T., Al-Ansari, N., Nguyen, M. D., Nguyen, H. D., Ly, H., Le, H. V, & Prakash, I. (2020). A Novel Hybrid Soft Computing Model Using Random Forest and Particle Swarm Optimization for Estimation of Undrained Shear Strength of Soil. *Sustainability*, *12*(6), 2218. https://doi.org/10.3390/su12062218

Probst, P., Wright, M. N., & Boulesteix, A. (2019). Hyperparameters and Tuning Strategies for Random Forest. *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, *9*(3). https://doi.org/10.1002/widm.1301

Samidi, F. S., Mohamed Radzi, N. A., Mohd Azmi, K. H., Mohd Aripin, N., & Azhar, N. A. (2022). 5G Technology: ML Hyperparameter Tuning Analysis for Subcarrier Spacing Prediction Model. *Applied Sciences (Switzerland)*, *12*(16). https://doi.org/10.3390/app12168271

Sotiroudis, S. P., Goudos, S. K., & Siakavara, K. (2019). Neural Networks and Random Forests: A Comparison Regarding Prediction of Propagation Path Loss for NB-IoT Networks. *2019 8th International Conference on Modern Circuits and Systems Technologies, MOCAST 2019*, *August*, 1–4. https://doi.org/10.1109/MOCAST.2019.8741751

Sotiroudis, S. P., Goudos, S. K., & Siakavara, K. (2020). Feature Importances: A Tool to Explain Radio Propagation and Reduce Model Complexity. *Telecom*, *1*(2). https://doi.org/10.3390/telecom1020009

Sousa, M., Alves, A., Vieira, P., Queluz, M., & Rodrigues, A. (2021). Analysis and Optimization of 5G Coverage Predictions Using a Beamforming Antenna Model and Real Drive Test Measurements. *IEEE Access*, *9*, 101787–101808. https://doi.org/10.1109/ACCESS.2021.3097633

Sun, Y., Zhang, J., Zhang, Y., Yu, L., Yuan, Z., Liu, G., & Wang, Q. (2022). Environment Features-Based Model for Path Loss Prediction. *IEEE Wireless Communications Letters*, *11*(9), 2010–2014. https://doi.org/10.1109/LWC.2022.3192516

Swersky, K. (2014). *Raiders of the Lost Architecture: Kernels for Bayesian Optimization in Conditional Parameter Spaces*. https://doi.org/10.48550/arxiv.1409.4011

Wang, C. X., Bian, J., Sun, J., Zhang, W., & Zhang, M. (2018). A survey of 5g channel measurements and models. *IEEE Communications Surveys and Tutorials*, *20*(4), 3142–3168. https://doi.org/10.1109/COMST.2018.2862141

Wang, L., Zhao, L., Liu, X., Fu, J., & Zhang, A. (2021). SepPCNET: Deeping Learning on a 3D Surface Electrostatic Potential Point Cloud for Enhanced Toxicity Classification and Its Application to Suspected Environmental Estrogens. *Environmental Science & Technology*, *55*(14), 9958–9967. https://doi.org/10.1021/acs.est.1c01228

Won, J., Shin, J., Kim, J.-H., & Lee, J.-W. (2023). A Survey on Hyperparameter Optimization in Machine Learning. *The Journal of Korean Institute of Communications and Information Sciences*, *48*(6), 733–747. https://doi.org/10.7840/kics.2023.48.6.733

Xu, Y., Wei, S., Bao, Y., & Li, H. (2019). Automatic Seismic Damage Identification of Reinforced Concrete Columns From Images by a Region-Based Deep Convolutional Neural Network. *Structural Control and Health Monitoring*, *26*(3), e2313. https://doi.org/10.1002/stc.2313

Yokoyama, A., & Yamaguchi, N. (2020). Optimal Hyperparameters for Random Forest to Predict Leakage Current Alarm on Premises. *E3s Web of Conferences*, *152*, 3003. https://doi.org/10.1051/e3sconf/202015203003

Yuhana, U. L., Purwarianti, A., & Imamah, I. (2022). Tuning Hyperparameter pada Gradient Boosting untuk Klasifikasi Soal Cerita Otomatis. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *8*(1), 134. https://doi.org/10.26418/jp.v8i1.50506

Yuliana, H., Iskandar, & Hendrawan. (2023). A Review of Coverage Prediction in Mobile Communication System using Machine Learning Algorithm. *2023 17th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*. https://doi.org/10.1109/TSSA59948.2023.10367027

Yuliana, H., Iskandar, & Hendrawan. (2024). Comparative Analysis of Machine Learning Algorithms for 5G Coverage Prediction: Identification of Dominant Feature Parameters and Prediction Accuracy. *IEEE Access*, *12*(January), 18939–18956. https://doi.org/10.1109/ACCESS.2024.3361403