Random Forest Antenna Model

Dataset Description

The dataset, `Data_sets_loop_1 - Copy.csv`, contains antenna design parameters and corresponding performance metrics derived from simulations or measurements. It comprises 46 columns: 39 features describing antenna geometry and material properties, and 7 target variables representing performance outcomes. The features include parameters such as `Permittivity`, `Height of Sub`, `Slot_width`, `L1_OuterDia`, and `Number of Loop`, which define the antenna's substrate, slot, microstrip, and loop configurations. The targets are `Frequency`, `RL depth`, `Bandwidth (%)`, `Radiation Efficiency (%)`, `Total Efficiency (%)`, `Gain`, and `F/B ratio`, which quantify the antenna's operational characteristics.

Theoretical Background

The Random Forest Regressor is an ensemble machine learning model that combines multiple decision trees to predict continuous outcomes. Each tree is trained on a random subset of the data and features, using bootstrap sampling and feature randomness to reduce overfitting and improve generalization. ***The final prediction is the average of all tree predictions, providing robustness against noise and non-linear relationships***. This makes Random Forest suitable for modeling relationships between antenna design parameters and performance metrics.The model's performance is evaluated using Mean Squared Error (MSE), which measures the average squared difference between predicted and actual values, and R-squared, which indicates the proportion of variance explained by the model. ***Hyperparameter tuning optimizes the number of trees (`n_estimators`) and tree depth (`max_depth`) to balance bias and variance, enhancing prediction accuracy.***

Procedure for Model Development

The development of the Random Forest model follows a structured workflow to ensure robust predictions and meaningful insights into antenna performance.

1. Data Preprocessing

- **Objective**: Prepare the dataset by addressing missing and invalid values and mitigating outliers to ensure model compatibility and reliability.
- **Steps**:
    - Load the dataset, skipping the first two rows (metadata) and trimming to 46 columns to exclude extraneous data.
    - Convert non-standard missing values (i.e empty strings, `"NaN"`, `"null"`) and infinite values (`inf`, `-inf`) to `np.nan` for consistent handling.

    - Drop rows with any `NaN` values to remove incomplete records, as the 209 missing target values and 200–208 missing feature values likely represent invalid simulations, and imputation may introduce bias.

2. Data Splitting

- **Objective**: Divide the dataset into training, validation, and testing sets to enable model training, hyperparameter tuning, and unbiased evaluation.
- **Steps**:
    - Split the cleaned dataset into features (39 design parameters) and targets (7 performance metrics).
    - ***Allocate 70% of the data for training, 15% for validation, and 15% for testing, using a fixed random seed (`random_state=42`) for reproducibility.***
    - Verify that no `NaN` values remain in the training targets to prevent training errors.

3. Model Training and Hyperparameter Tuning

- **Objective**: Train a Random Forest Regressor to predict the seven target variables and optimize its hyperparameters for maximum accuracy.
- **Steps**:
    - Initialize a `RandomForestRegressor` with a fixed random seed for results.
    - Perform grid search hyperparameter tuning over:
        - `n_estimators`: [50, 100, 200] (number of trees).
        - `max_depth`: [None, 10, 20] (maximum tree depth).
    - Evaluate each parameter combination on the validation set using R-squared with `multioutput='uniform_average'` to aggregate performance across the seven targets.
    - Select the parameter combination yielding the highest R-squared score.

4. Model Evaluation

- **Objective**: Assess the model's predictive performance on unseen data and quantify its accuracy for each target.
- **Steps**:
    - Generate predictions on the test set.
    - Calculate MSE and R-squared for each target to evaluate prediction error and explain variance.
    - Calculate average MSE and R-squared across all targets to summarize overall performance.

5. Visualization

- **Objective**: Generate visual insights into model performance and feature contributions to guide antenna design optimization.
- **Steps**:
    - Create a **feature importance plot** to visualize the relative importance of each feature, highlighting key predictors (`Permittivity`, `L1_InnerDia`) for antenna performance.
    - Generate **actual vs. predicted scatter plots** for each target, comparing test set predictions to actual values, with a y=x reference line and R-squared values to assess prediction accuracy.
    - Produce **error distribution plots** for each target, showing histograms of prediction errors with kernel density estimation (KDE) curves to evaluate the spread and bias of residuals.

Summary

The Random Forest model is used to predict how well an antenna will perform based on its design settings. It works well because it can handle complex relationships between different inputs. Before training the model, the data goes through careful cleaning to fix missing values and remove any unusual outliers. Then, the data is split into training and testing sets for the model are chosen. Since the model predicts multiple things at once, it's evaluated on how well it does across all of them. Charts and graphs help us understand which design factors matter the most and how accurate the predictions are. By cleaning the data and using a strong model like Random Forest, we get reliable results that can help improve antenna designs.
