

Hybrid Optical/Radio Frequency Communication Channel Model Final Reports

Prashant Shrestha
a1909636

November 23, 2024

Report submitted for **Data Science Research Project A** at the
School of Mathematical Sciences, University of Adelaide



THE UNIVERSITY
of ADELAIDE

Project Area: **Hybrid Optical/Radio Frequency Communication
Channel Model**

Project Supervisor: **Dr Siu Wai Ho**

In submitting this work I am indicating that I have read the University's Academic Integrity Policy. I declare that all material in this assessment is my own work except where there is clear acknowledgement and reference to the work of others.

I give permission for this work to be reproduced and submitted to other academic staff for educational purposes.

OPTIONAL: I give permission this work to be reproduced and provided to future students as an exemplar report.

Abstract

The increasing demand for high-capacity data transmission has made it necessary to develop hybrid communication systems that combine radio frequency (RF) and free-space optical (FSO) channels. While these systems promise enhanced reliability and performance, their efficiency is significantly influenced by weather conditions, such as rain, humidity, distance and so on. This project investigates the relationship between weather parameters and signal attenuation in hybrid RF/FSO systems. A dataset containing weather metrics and corresponding attenuation values is utilized to develop predictive models using Random Forest algorithm.

The study evaluates two modeling approaches: specific models trained for individual weather conditions and a generic model applicable across all conditions. Model performance is assessed using root mean square error (RMSE) and R^2 metrics. Feature importance analysis reveals critical parameters impacting system performance, such as visibility and rain intensity. Results indicate that specific models often outperform the generic approach, particularly under adverse weather conditions. These findings align with the recommendations from ITU standards, which highlight weather as a crucial factor in link design for hybrid systems (ITU Radiocommunication Sector 2019; Sector 2012).

This work builds upon the principles of hybrid RF/FSO channel modeling discussed in prior research. Studies such as "Optical Wireless Hybrid Networks" emphasize the potential of combining RF and optical channels to address the limitations of single-mode communication systems (Chowdhury et al. 2020). Furthermore, methodologies from recent advancements in RF/FSO models provide a foundation for this analysis, especially for feature selection and weather-aware modeling (Han 2023; Nadeem et al. 2010). The outcomes contribute to the broader understanding of hybrid channel modeling, serving as a foundation for developing more resilient communication infrastructures.

1 Introduction

The rapid growth in data demand for modern communication networks has created quite challenges for existing communication systems. While radio frequency (RF) communication has been a foundation for wireless communication, its limited bandwidth and more likely to be interfered by poor weather conditions highlight the need for supporting technologies. For data transmission Free-space optical(FSO) uses infrared light, offers a promising solution with its high bandwidth and immunity to electromagnetic interference (ITU Radiocommunication Sector 2019). However, FSO systems are highly sensitive to atmospheric conditions, including rain, fog, and dust, which cause significant signal attenuation (Sector 2012).

Hybrid RF/FSO communication systems combine the strengths of both RF and FSO channels, leveraging the robustness of RF under poor visibility and the high bandwidth of FSO under clear conditions. Such systems are gaining attention for applications like 5G backhaul, satellite-ground communication, and disaster recovery (Chowdhury et al. 2020; Han 2023). Despite their potential, the design and optimization of hybrid RF/FSO systems require accurate models to predict signal performance under varying weather conditions (Nadeem et al. 2010).

By developing predictive models for hybrid RF/FSO systems, the project aims to address these problems. Two approaches are explored: (1) specific models trained for individual weather conditions and (2) a generic model that generalizes across all conditions. These models will utilize a dataset containing weather metrics and attenuation values, analyzed using Random Forest algorithms. This study seeks to provide insights into the important features affecting system performance and guide the development of more robust hybrid communication systems.

2 Background

The hybrid communication system is the technology with the combination of RF and FSO which are emerging as a Hybrid communication systems that integrate radio frequency (RF) and free-space optical (FSO) technologies are emerging as a promising solution to address the limitations of single-mode communication systems. Both RF and FSO have unique characteristics that make them suitable for specific scenarios but also pose significant challenges, especially under varying atmospheric conditions.

2.1 Radio Frequency Communication

RF communication is a well built and used technology that forms the backbone of modern wireless networks, including 4G, 5G, and satellite-based systems. Its resilience to moderate weather conditions, such as rain and fog, makes it a reliable option for long-range and low-bandwidth communication (ITU Radiocommunication Sector 2019). However, RF channels face challenges such as limited bandwidth and susceptibility to electromagnetic interference, especially in dense urban environments. These limitations hinder its ability to meet the growing demand for high-data-rate applications.

2.2 Free-Space Optical Communication

FSO communication employs infrared light to transmit data through the atmosphere, offering several advantages over RF, such as higher bandwidth, immunity to electromagnetic interference, and low latency (Sector 2012; Chowdhury et al. 2020). It is particularly useful for applications requiring high-speed data transmission, such as data center interconnects, last-mile access, and satellite communication. However, FSO systems are highly sensitive to atmospheric disturbances, including rain, fog, dust, and turbulence, which significantly degrade signal quality. For instance, fog can cause signal attenuation as high as 30–40 dB/km, rendering FSO systems less reliable during adverse weather conditions (Han 2023).

2.3 Hybrid RF/FSO Systems

To mitigate or overcome the limitations of either RF or FSO systems, hybrid RF/FSO systems have been proposed. These systems combine the strengths of both channels: RF for reliable communication during poor visibility and FSO for high-speed data transmission during clear weather (Chowdhury et al. 2020; Nadeem et al. 2010). By dynamically

switching between RF and FSO channels or using them simultaneously, hybrid systems can maintain high performance across varying environmental conditions. Applications of hybrid RF/FSO systems include:

- **5G Backhaul and Fronthaul:** Addressing the bandwidth and latency requirements of next-generation wireless networks.
- **Satellite-Ground Communication:** Enhancing reliability in satellite communication by mitigating weather-induced disruptions.
- **Disaster Recovery:** Providing robust communication links during emergencies when conventional infrastructure is unavailable.

2.4 Weather Impact on Hybrid Systems

Weather conditions are a critical factor in the performance of hybrid RF/FSO systems. Rain, snow, fog, and dust storms affect the attenuation levels of both RF and FSO channels. For example, RF signals experience rain attenuation due to scattering and absorption by raindrops, while FSO signals are heavily attenuated by fog and dust particles (ITU Radiocommunication Sector 2019; Sector 2012). Understanding the relationship between weather parameters and signal performance is crucial for designing effective hybrid systems.

2.5 Modeling Weather-Dependent Attenuation

Accurate modeling of weather-induced attenuation is vital for optimizing hybrid RF/FSO systems. Recent studies have proposed various methods, including empirical, statistical, and machine learning-based models, to predict attenuation levels based on weather metrics (Han 2023; Nadeem et al. 2010). Random Forest algorithms, in particular, have shown promise in handling highly nonlinear relationships and identifying key features influencing system performance. This project builds upon these advancements to develop predictive models that can inform the design and deployment of resilient hybrid communication systems.

3 Methods

This project focuses on developing predictive models for hybrid RF/FSO communication systems to analyze the impact of weather conditions on signal attenuation. The methodology is divided into five key stages: dataset preparation, feature engineering, model training, evaluation, and comparative analysis of specific and generic approaches.

3.1 Dataset Preparation

The dataset used in this project contains weather metrics and their corresponding attenuation values for RF and FSO channels. Key features include:

- **Weather Metrics:** Parameters such as rain intensity, visibility, temperature, and wind speed.
- **Categorical Variables:** `SYNOPCode`, which represents weather conditions like clear weather, rain, or dust storms, is handled as a categorical variable (ITU Radiocommunication Sector 2019; Sector 2012).
- **Target Variables:** Signal attenuation levels for RF (`RFL_Att`) and FSO (`FSO_Att`) channels.

To ensure consistent and accurate analysis, the required preprocessing steps are as follows:

- Handling missing values through imputation.
- Encoding categorical variables using one-hot encoding.
- Normalizing numerical features to improve model performance (Han 2023).

But since, the dataset is clean we don't need to use imputation, 'SYNOPCode' is a categorical one but its not needed to be one hot encoded and finally, for decision based trees like random forest, normalizing is meaningless. So, we can skip these steps.

3.2 Exploratory Data Analysis

The dataset underwent a thorough exploratory data analysis (EDA) to gain insights into variable distributions, relationships, and potential anomalies. The following steps were performed:

- **Descriptive Statistics:** Figure 1 shows summary statistics such as mean, median, standard deviation, and range were computed for all numerical variables.

Descriptive Statistics:

	FSO_Att	RFL_Att	AbsoluteHumidity	AbsoluteHumidityMax	AbsoluteHumidityMin	Distance	Frequency	Particulate	ParticulateMax	ParticulateMin	...
count	91379.000000	91379.000000	91379.000000	91379.000000	91379.000000	91379.000000	9.137900e+04	91379.000000	91379.000000	91379.000000	...
mean	6.769458	11.619098	9.553919	10.032760	9.076251	3297.930328	7.850005e+10	27.065979	28.417120	25.717089	...
std	3.903843	3.438873	5.858577	6.162798	5.575927	1224.305893	5.000027e+09	72.134023	75.761896	68.595239	...
min	0.788363	0.027142	1.141556	1.238270	1.049744	2012.000148	7.350000e+10	0.000000	0.000000	0.000000	...
25%	3.473063	10.829331	4.958993	5.205861	4.709511	2019.431812	7.350000e+10	0.000000	0.000000	0.000000	...
50%	6.336167	11.856560	6.870737	7.205499	6.524046	2959.863686	8.350000e+10	0.000000	0.000000	0.000000	...
75%	8.664984	12.847944	14.049470	14.782679	13.379256	4820.890157	8.350000e+10	16.947618	17.775980	16.038090	...
max	32.455222	46.893150	24.790883	26.407305	24.268431	4827.999971	8.350000e+10	1621.001906	1753.747866	1500.666382	...

8 rows x 27 columns

Figure 1: Summary Statistics for all the features

- **Distributions and Outliers:** Histograms were plotted to visualize feature distributions. Features such as rain intensity and visibility showed significant fluctuation, likely influenced by weather conditions as shown in Figure 2 and 3.

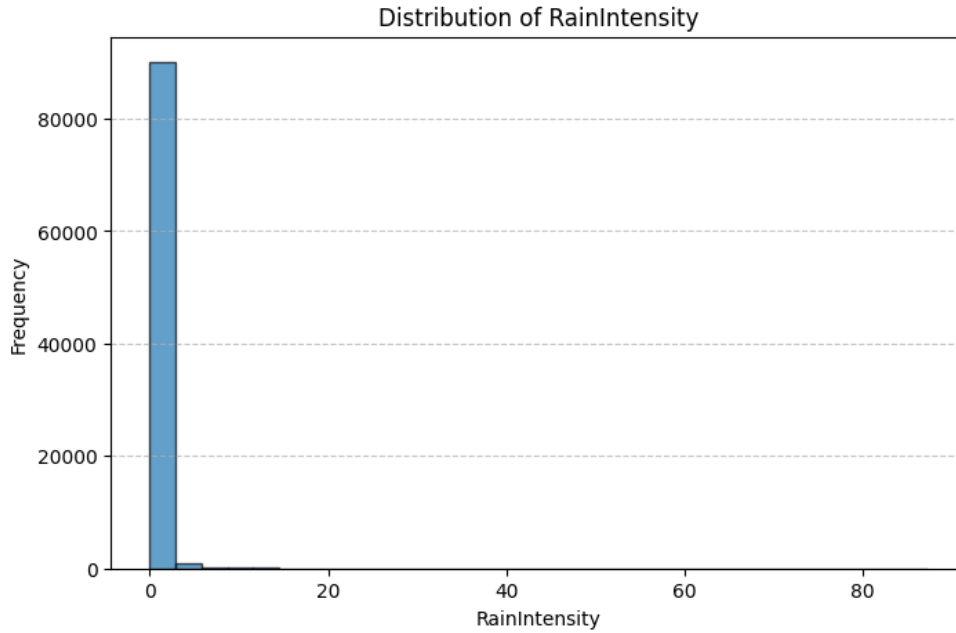


Figure 2: Histogram plot of Rain Intensity

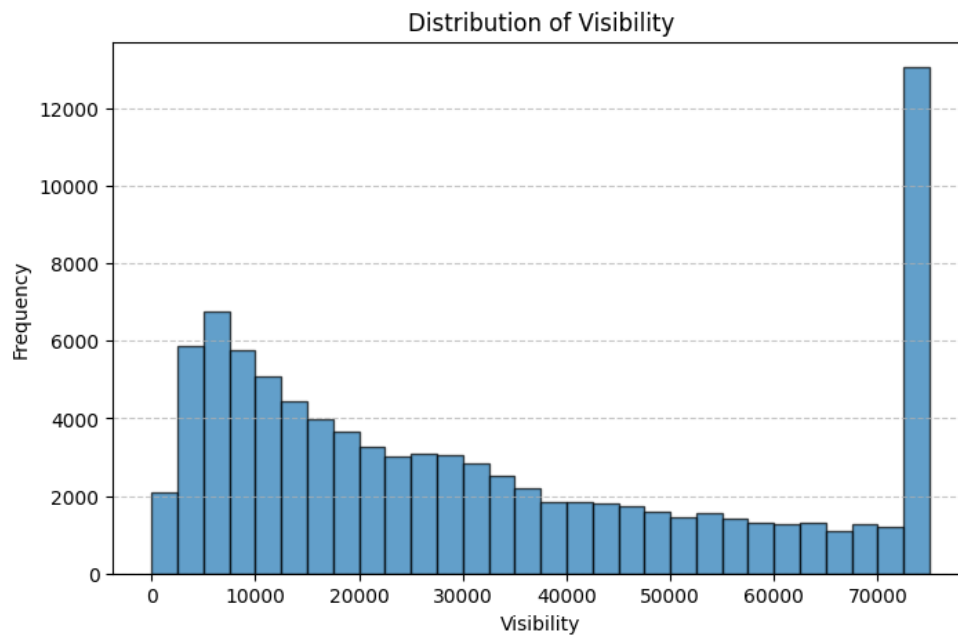


Figure 3: Histogram plot of Visibility

Figure 4 and 5 box-plots highlighted outliers in metrics such as temperature and particulate matter, which may impact model performance. `RFL_Att` exhibited slight skewness.

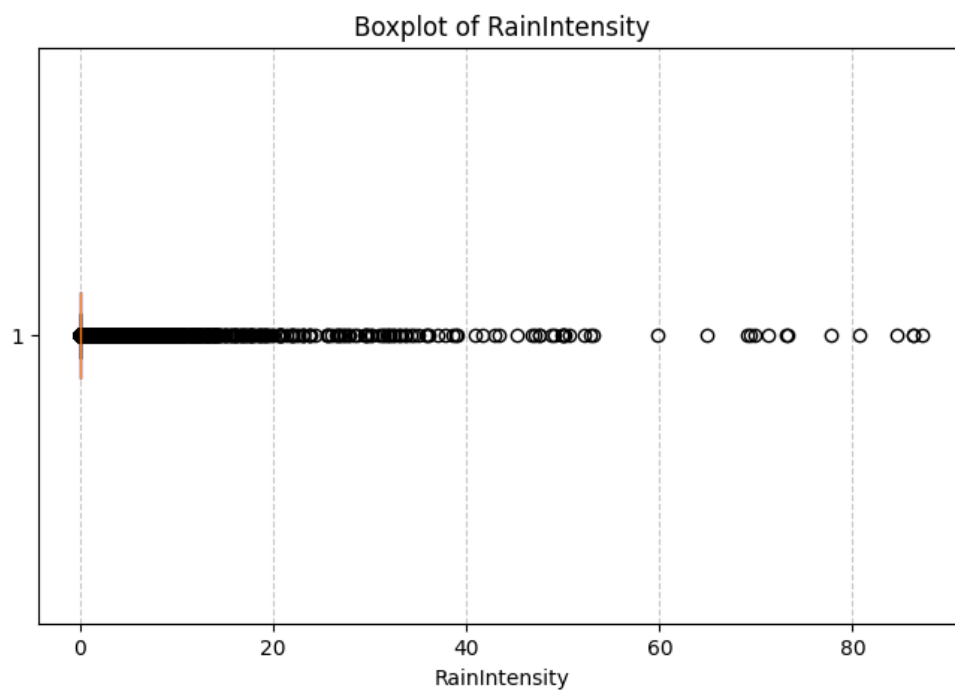


Figure 4: Box-plot plot of Rain intensity

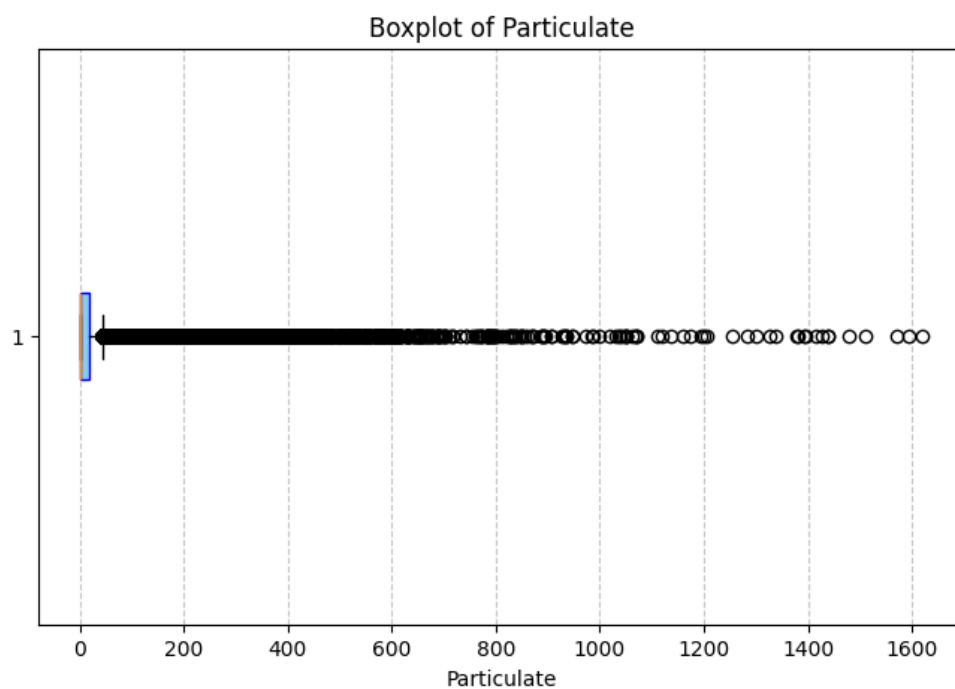


Figure 5: Box-plot plot of Particulate

- **Correlation Analysis:** The correlation matrix in Figure 6 revealed strong relationships between `FSO_Att` and features such as particulate matter and relative humidity. Similarly, `RFL_Att` was strongly correlated with rain intensity and particulate matter.

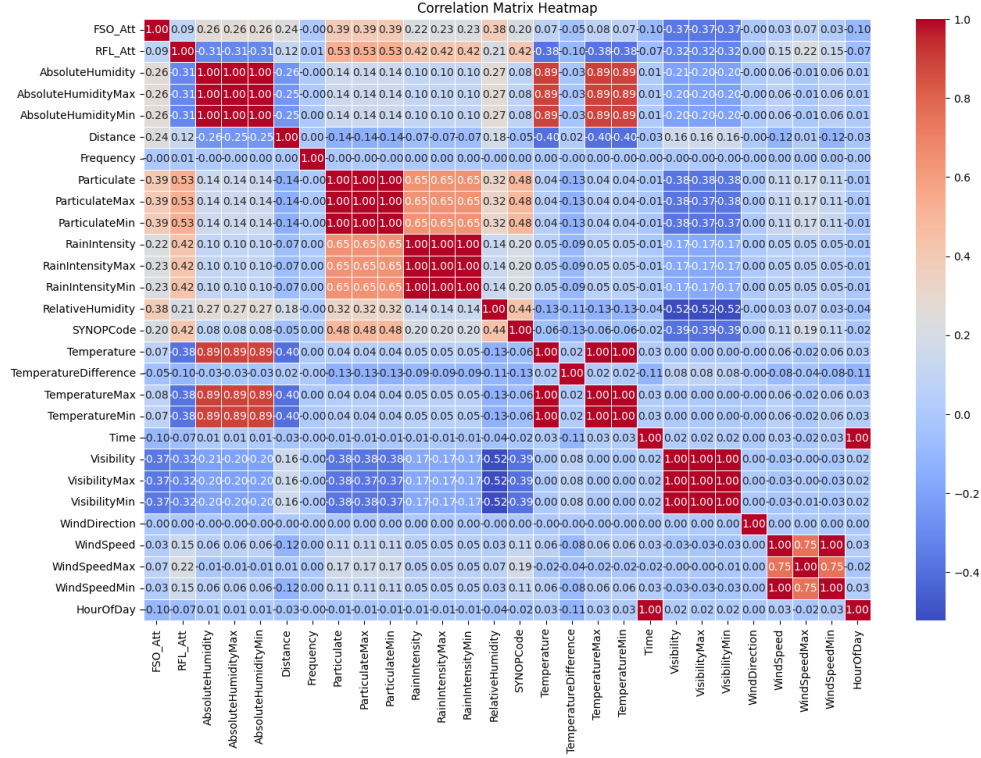


Figure 6: Heat map plot showing correlation of each feature

- **Trends:** Time-based analysis showed hourly trends in attenuation, with significant variations during adverse weather conditions (Chowdhury et al. 2020).

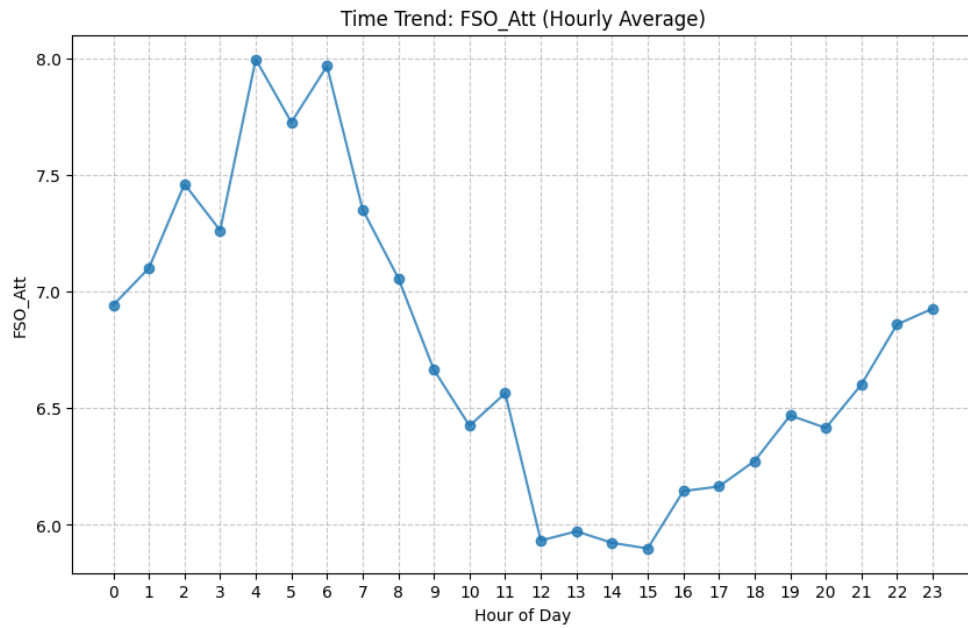


Figure 7: Time Trend for fso_att column

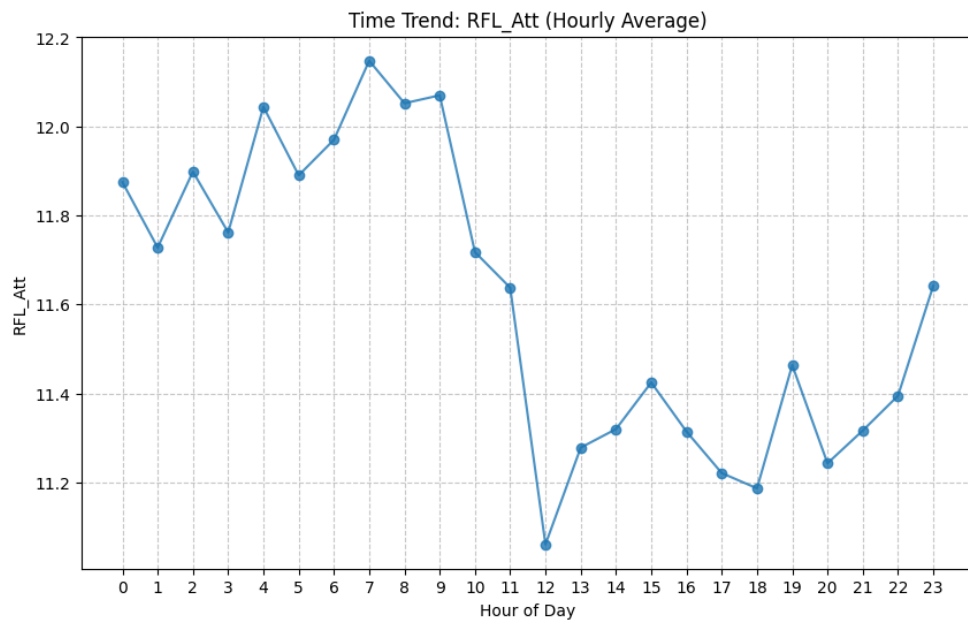


Figure 8: Time Trend for rfl_att column

Figures summarizing these analyses include histograms, boxplots for outliers, and the correlation heatmap.

3.3 Feature Engineering

Feature selection was conducted to improve model performance and reduce redundancy:

- **Random Forest Importance:** Random Forest algorithms were used to rank features based on their contribution to predicting attenuation. Visibility feature was identified as the most critical for FSO attenuation, while absolute humidity was the primary factor for RF attenuation.

Algorithm 1 An algorithm for ranking the importance of predictor variables.

- 1: Let \mathcal{S} be the set of N predictor variables.
 - 2: Let \mathcal{R} be an empty table.
 - 3: The training data containing only the variables in \mathcal{S} are used to train a random forest.
 - 4: The RMSE and R^2 value for the random forest are calculated.
 - 5: The importance of predictor variables is ranked according to the out-of-bag information.
 - 6: The least important predictor is removed from \mathcal{S} and is combined with the RMSE and R^2 value to form a new row at the end of the table \mathcal{R} .
 - 7: If \mathcal{S} is non-empty, go to Step 3.
 - 8: The output is the table \mathcal{R} .
-

Figure 9: Feature Importance Ranking Algorithm

- **Dimensionality Reduction:** Features with minimal importance were removed iteratively, guided by model performance metrics such as RMSE and R^2 .

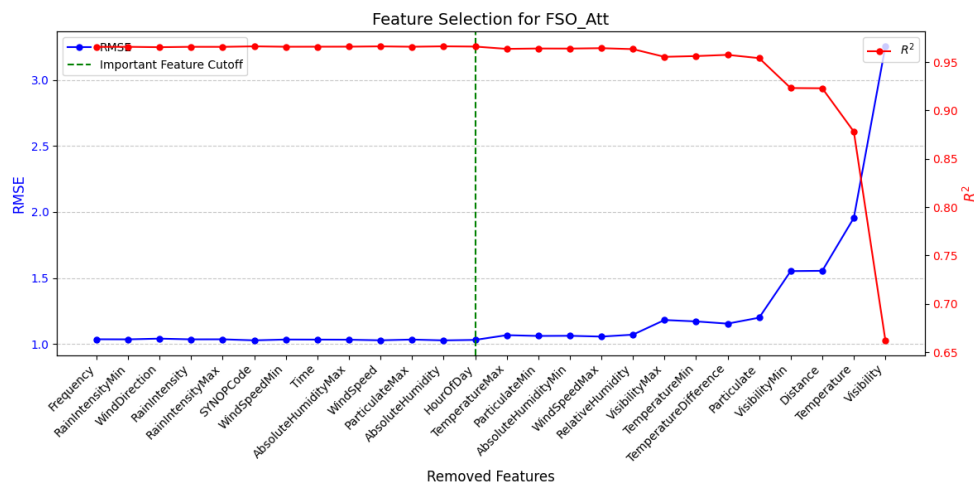


Figure 10: Feature Selection for FSO Attenuation

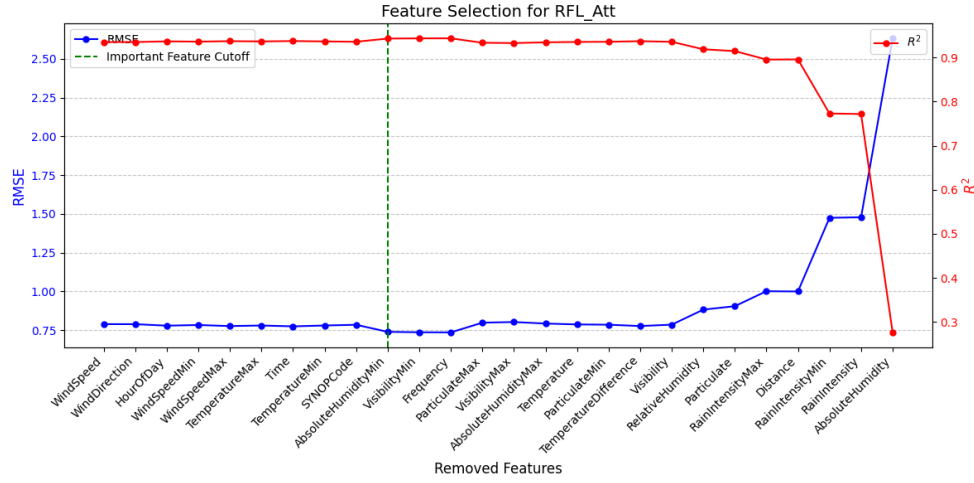


Figure 11: Feature Selection for RF Attenuation

Figure 10 and 11 shows the impact of each feature on the RF attenuation and FSO attenuation. The green dotted line is the line that is the cutoff point which I created by observing the changes in RMSE and R^2 and took the features from that point to the right and the features are as listed below:

Features for FSO: ['HourOfDay', 'TemperatureMax', 'ParticulateMin', 'AbsoluteHumidityMin', 'WindSpeedMax', 'RelativeHumidity', 'VisibilityMax', 'TemperatureMin', 'TemperatureDifference', 'Particulate', 'VisibilityMin', 'Distance', 'Temperature', 'Visibility']

Features for RF: ['AbsoluteHumidityMin', 'VisibilityMin', 'Frequency', 'ParticulateMax', 'VisibilityMax', 'AbsoluteHumidityMax', 'Temperature', 'ParticulateMin', 'TemperatureDifference', 'Visibility', 'RelativeHumidity', 'Particulate', 'RainIntensityMax', 'Distance', 'RainIntensityMin', 'RainIntensity', 'AbsoluteHumidity']

3.4 Model Training

Two distinct modeling approaches were employed:

- **Specific Models (Method 1):** Separate models were trained for each weather condition defined by `SYNOPCode`. This approach allowed the models to specialize in handling unique conditions like heavy rain (ITU Radiocommunication Sector 2019).
- **Generic Model (Method 2):** A single model was trained using the entire dataset, with `SYNOPCode` included as a feature to generalize across all conditions (Chowdhury et al. 2020).

Random Forest regressors were used for both methods due to their robustness in handling nonlinear relationships and categorical variables (Han 2023). The dataset was split into training (70%) and testing (30%) subsets, ensuring no data leakage.

3.5 Evaluation

The models were evaluated using the following metrics:

3.5.1 Root Mean Square Error (RMSE)

The RMSE measures the average squared of prediction errors and is defined as:

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

Where:

- y_i : Actual value of the target variable.
- \hat{y}_i : Predicted value.
- n : Number of observations.

A lower RMSE indicates better model performance, as it reflects smaller deviations between predicted and actual values (Sector 2012).

3.5.2 Coefficient of Determination (R^2)

The R^2 metric measures how well the model explains the variance in the target variable and is defined as:

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

Where:

- \bar{y} : Mean of the actual values.
- $\sum_{i=1}^n (y_i - \hat{y}_i)^2$: Residual sum of squares.
- $\sum_{i=1}^n (y_i - \bar{y})^2$: Total sum of squares.

An R^2 value closer to 1 indicates better performance, as it implies the model explains most of the variance in the target variable (Han 2023; Nadeem et al. 2010).

4 Results

This section presents the findings from evaluating the predictive models for hybrid RF/FSO communication systems. The results are divided into three parts: feature importance analysis, model evaluation, and comparative performance of specific and generic models. The visualizations and performance metrics illustrate the models' effectiveness under various weather conditions.

4.1 Feature Importance Analysis

Feature selection was performed using Random Forest algorithms to determine the key factors affecting RF and FSO signal attenuation. For FSO attenuation, visibility, temperature and distance and were identified as the most critical features as shown in Figure 10. In contrast, RF attenuation was most affected by rain intensity, absolute humidity and rain intensity as shown in Figure 11. These findings highlight the distinct dependencies of RF and FSO systems on specific weather parameters.

4.2 Model Evaluation

The evaluation metrics for specific (Method 1) and generic (Method 2) models are summarized below:

- **Specific Models (Method 1):**
 - FSO:
 - * Mean RMSE: 0.8171
 - * Mean R^2 : 0.9346
 - RF:
 - * Mean RMSE: 0.5420
 - * Mean R^2 : 0.9375
- **Generic Models (Method 2):**
 - FSO:
 - * RMSE: 1.0198
 - * R^2 : 0.9669
 - RF:
 - * RMSE: 0.7739
 - * R^2 : 0.9375

This result implies that model trained with the specific weather condition performs well compared to the generic model as RMSE is lower for specific model making its prediction more accurate and reliable. Moreover, both model shows similar R^2 score which implies that both methods generalizes well.

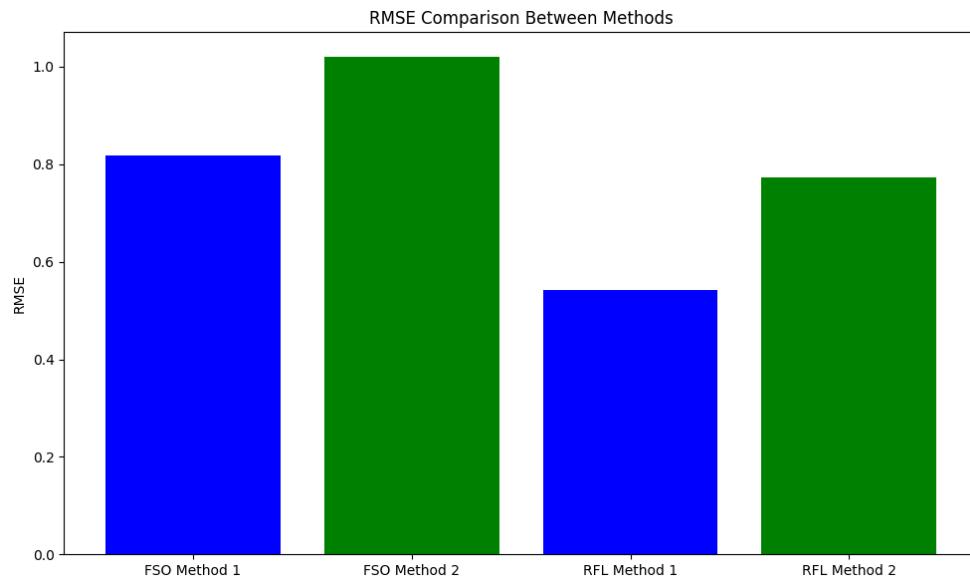


Figure 12: RMSE Comparison Between Methods for RF and FSO Channels

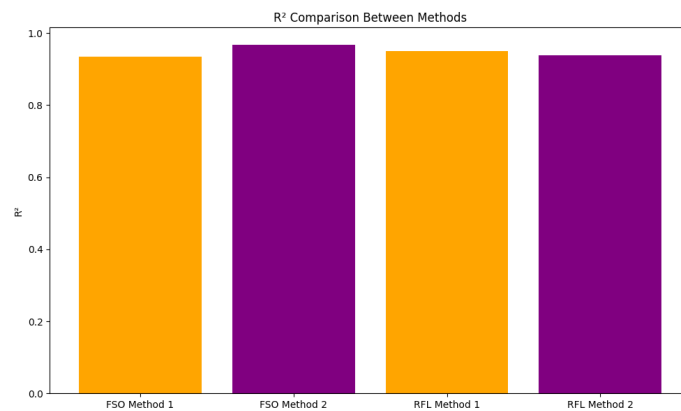


Figure 13: R^2 Comparison Between Methods for RF and FSO Channels

5 Conclusion

This study explored the predictive modeling of hybrid RF/FSO communication systems, focusing on the impact of weather conditions on signal attenuation. Using a comprehensive dataset, various weather metrics were analyzed through exploratory data analysis (EDA) to uncover patterns, variability, and significant correlations. Features such as rain intensity and visibility emerged as critical determinants of attenuation, particularly for RF and FSO systems, respectively. These findings align with established propagation studies, which emphasize the sensitivity of RF signals to precipitation and the vulnerability of FSO systems to reduced visibility during fog and haze (ITU Radiocommunication Sector 2019; Sector 2012).

Two modeling approaches were implemented: specific models based on distinct weather conditions and a generic model designed for broader applications. Specific models demonstrated better performance, particularly for FSO attenuation, achieving lower RMSE values and higher predictive accuracy compared to the generic model. This result highlights the importance of weather-aware modeling for optimizing hybrid systems, as specialized models can better account for the variability inherent in environmental conditions (Nadeem et al. 2010).

The correlation analysis underscored the strong relationships between signal attenuation and features such as particulate matter, relative humidity, and rain intensity. These insights guided feature selection and model optimization, ensuring that only the most relevant predictors were included in the final models. The use of Random Forest regressors proved effective in capturing the nonlinear dependencies between weather variables and signal attenuation, consistent with findings in previous research on weather-induced propagation effects (Chowdhury et al. 2020).

Overall, the study demonstrated the value of hybrid RF/FSO systems in mitigating weather-related disruptions to communication networks. While RF systems can compensate for FSO signal losses during low-visibility conditions, the latter performs more reliably during high-rainfall scenarios, emphasizing their complementary nature. These findings suggest that future advancements in hybrid communication technologies should focus on dynamic, adaptive models that respond to real-time weather data to ensure reliable performance under diverse environmental conditions (ITU Radiocommunication Sector 2019).

The methodology and results presented in this study provide a foundation for further research into hybrid communication systems. Future work could explore the integration of additional weather metrics, such as wind speed or aerosol concentrations, and evaluate their effects on

system performance. Additionally, the use of advanced machine learning techniques, such as ensemble methods or neural networks, may further improve predictive accuracy and adaptability to highly dynamic weather conditions.

Acknowledgements

I would like to express my sincere gratitude to my academic supervisor, Dr Siu Wai Ho, for his consistent guidance and constant support throughout the course of this research. His expertise and encouragement have been crucial in shaping the direction and outcome of this study.

I am also deeply thankful to University of Adelaide for providing access to the resources and tools necessary for conducting this research. The computational facilities and technical support offered by the team were critical in analyzing the dataset and implementing the predictive models.

Lastly, I also want to acknowledge that I took help with the Chat-GPT to get the syntax to create necessary format in the Latex file. Likewise, I also took help from it to create the balanced sampling as I was struggling to create the desired sampling.. Moreover, I also used it to understand some of the topics in deeper level and to understand the research papers used.

Appendices

GitHub Repository

The code and analysis for this project are available in the following publicly accessible GitHub repository:

GitHub Repository: Data Science Research Project

Python Code

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4  import matplotlib.pyplot as plt
5  import numpy as np
6  import seaborn as sns
7  from sklearn.utils import resample
8  from sklearn.model_selection import train_test_split
9  from sklearn.ensemble import RandomForestRegressor
10 from sklearn.metrics import root_mean_squared_error,
    r2_score
11
12 # Loading the Dataset
13 rflfso_data = pd.read_csv('/content/drive/MyDrive/
    RFLFSODataFull.csv')
14
15 # Displaying the dataset
16 rflfso_data.head()
17
18 # Step 1: Inspect the Dataset
19 print("Dataset Info:")
20 rflfso_data.info()
21
22 # Ensuring there are no NA values
23 rflfso_data.isna().sum()
24
25 # Ensuring SYNOP code is integer
26 rflfso_data['SYNOPCode'] = rflfso_data['SYNOPCode'].astype
    (int)
27
28 # Summarizing the Dataset
29 print("\nDescriptive Statistics:")
30 rflfso_data.describe()
31
32 # Checking the columns
33 rflfso_data.columns
34
35 # Visualizing Distributions for each and every columns in
    the dataset
```

```

36 for column in rflfso_data.columns:
37     plt.figure(figsize=(8, 5))
38     plt.hist(rflfso_data[column], bins=30, edgecolor='k',
39             alpha=0.7)
40     plt.title(f'Distribution of {column}')
41     plt.xlabel(column)
42     plt.ylabel('Frequency')
43     plt.grid(axis='y', linestyle='--', alpha=0.7)
44     plt.show()
45
46 # Identifying Outliers
47 for column in rflfso_data.columns:
48     plt.figure(figsize=(8, 5))
49     plt.boxplot(rflfso_data[column], vert=False,
50               patch_artist=True, boxprops=dict(facecolor='skyblue',
51               color='blue'))
52     plt.title(f'Boxplot of {column}')
53     plt.xlabel(column)
54     plt.grid(axis='x', linestyle='--', alpha=0.7)
55     plt.show()
56
57 # Correlation Analysis
58 correlation_matrix = rflfso_data.corr()
59
60 # Heatmap
61 plt.figure(figsize=(15, 10))
62 sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
63            fmt=".2f", linewidths=0.5)
64 plt.title('Correlation Matrix Heatmap')
65 plt.show()
66
67 # Time Trend Analysis
68 # Assuming 'Time' is in hours from 0-23
69 rflfso_data['HourOfDay'] = rflfso_data['Time'] % 24
70
71 # Aggregating by hour of the day
72 key_trend_columns = ['FSO_Att', 'RFL_Att']
73 hourly_means = rflfso_data.groupby('HourOfDay')[
74     key_trend_columns].mean()
75
76 # Plotng the trends
77 for column in key_trend_columns:
78     plt.figure(figsize=(10, 6))
79     plt.plot(hourly_means.index, hourly_means[column],
80             marker='o', linestyle='--', alpha=0.8)
81     plt.title(f'Time Trend: {column} (Hourly Average)')
82     plt.xlabel('Hour of Day')
83     plt.ylabel(column)
84     plt.grid(linestyle='--', alpha=0.7)
85     plt.xticks(hourly_means.index)
86     plt.show()

```

```

81
82 # Checking the unique SYNOPCode categories and their
    counts
83 def plot_synop_counts(data):
84     synop_counts = data['SYNOPCode'].value_counts()
85
86     # Creating a bar graph with counts of each unique
        SYNOPCode
87     plt.figure(figsize=(10, 6))
88     ax = synop_counts.plot(kind='bar', color='skyblue',
        edgecolor='black')
89
90     # Adding count values on top of each bar
91     for index, value in enumerate(synop_counts):
92         plt.text(index, value + 0.5, str(value), ha='center',
            , fontsize=10)
93
94     # Adding titles and labels
95     plt.title('Distribution of Unique SYNOPCode Values',
        fontsize=14)
96     plt.xlabel('SYNOPCode', fontsize=12)
97     plt.ylabel('Count', fontsize=12)
98     plt.xticks(rotation=0)
99     plt.grid(axis='y', linestyle='--', alpha=0.7)
100    plt.tight_layout()
101    plt.show()
102
103 plot_synop_counts(rflfso_data)
104
105 # Finding the median count of SYNOPCode categories
106 synop_counts = rflfso_data['SYNOPCode'].value_counts()
107 median_count = int(synop_counts.median())
108
109 # Separating the dataset by SYNOPCode categories
110 groups = [rflfso_data[rflfso_data['SYNOPCode'] == code]
111            for code in synop_counts.index]
112
113 # Initializing a list to hold the balanced groups
114 balanced_groups = []
115
116 for group in groups:
117     if len(group) > median_count:
118         # Down-sampling majority classes
119         balanced_group = resample(group, replace=False,
120                                   n_samples=median_count,
121                                   random_state=42)
122
123     elif len(group) < median_count:
124         # Over-sampling minority classes
125         balanced_group = resample(group, replace=True,
126                                   n_samples=median_count,
127                                   random_state=42)

```

```

125     else:
126         # Modification not needed for the groups with
            exact median count
127         balanced_group = group
128
129         balanced_groups.append(balanced_group)
130
131 # Combine all balanced groups into a single dataset
132 balanced_data = pd.concat(balanced_groups)
133
134 # Shuffle the dataset
135 balanced_data = balanced_data.sample(frac=1, random_state
            =42).reset_index(drop=True)
136
137 print("Balancing completed.")
138
139 sampled_data = balanced_data # For testing downsampled and
            balanced dataset
140 plot_synop_counts(sampled_data)
141
142 def plot_feature_selection_results(results, target, title):
143     """
144     Plotting RMSE and R2 against removed features with a
            cutoff line.
145
146     Parameters:
147         results (pd.DataFrame): Results from feature
            selection.
148         title (str): Title for the plot.
149     """
150     rmse_values = results['RMSE']
151     r2_values = results['R2']
152     removed_features = results['Removed Feature']
153
154     # Determining cutoff point where RMSE stabilizes or R
            ^2 starts plateauing
155     cutoff_index = 0
156     selected_features = []
157     if target == 'FSO_Att':
158         cutoff_index = 12
159     elif target == 'RFL_Att':
160         cutoff_index = 9
161
162     # Creating the plot
163     fig, ax1 = plt.subplots(figsize=(12, 6))
164
165     # Plotting RMSE on the primary y-axis
166     ax1.plot(removed_features, rmse_values, label='RMSE',
            color='blue', marker='o', markersize=5)
167     ax1.set_xlabel('Removed Features', fontsize=12)
168     ax1.set_ylabel('RMSE', fontsize=12, color='blue')

```

```

169     ax1.tick_params(axis='y', labelcolor='blue')
170     ax1.set_xticks(range(len(removed_features)))
171     ax1.set_xticklabels(removed_features, rotation=45, ha=
        'right', fontsize=10)
172
173     # Adding R2 on the secondary y-axis
174     ax2 = ax1.twinx()
175     ax2.plot(removed_features, r2_values, label='$R^2$',
        color='red', marker='o', markersize=5)
176     ax2.set_ylabel('$R^2$', fontsize=12, color='red')
177     ax2.tick_params(axis='y', labelcolor='red')
178
179     # Add vertical cutoff line
180     ax1.axvline(x=cutoff_index, color='green', linestyle='
        --', label='Important Feature Cutoff')
181
182     # Adding grid, legend, and title
183     ax1.grid(axis='y', linestyle='--', alpha=0.7)
184     ax1.legend(loc='upper left', fontsize=10)
185     ax2.legend(loc='upper right', fontsize=10)
186     plt.title(title, fontsize=14)
187     plt.tight_layout()
188
189     # Showing the plot
190     plt.show()
191     return selected_features
192
193 def feature_selection(data, target, plot_cutoff=False):
194     # Initializing variables
195     S = list(data.columns) # Set of all features
196     S.remove(target) # Removing the target column from
        the feature set
197     R = [] # Table to store results
198     # Excluding the target and any other non-predictor
        columns
199     non_predictor_columns = [target, 'FSO_Att', 'RFL_Att']
200     S = [col for col in data.columns if col not in
        non_predictor_columns]
201
202     while S:
203         # Splitting data into training and testing sets
204         X = data[S]
205         y = data[target]
206         X_train, X_test, y_train, y_test =
            train_test_split(X, y, test_size=0.2,
                random_state=42)
207
208         # Training a Random Forest model
209         rf = RandomForestRegressor(random_state=42)
210         rf.fit(X_train, y_train)
211

```

```

212     # Predicting and calculating RMSE and R2
213     y_pred = rf.predict(X_test)
214     rmse = root_mean_squared_error(y_test, y_pred)
215     r2 = r2_score(y_test, y_pred)
216
217     # Getting feature importances
218     feature_importances = rf.feature_importances_
219     feature_ranking = pd.DataFrame({
220         'Feature': S,
221         'Importance': feature_importances
222     }).sort_values(by='Importance', ascending=False)
223
224     # Removing the least important feature
225     least_important = feature_ranking.iloc[-1]['
226         Feature']
227     S.remove(least_important)
228
229     # Appending results to the table
230     R.append({
231         'Removed Feature': least_important,
232         'RMSE': rmse,
233         'R2': r2
234     })
235
236     # Converting results to DataFrame
237     results_df = pd.DataFrame(R)
238
239     # Plotting results if plot_cutoff is True
240     if plot_cutoff:
241         plot_feature_selection_results(results_df, target,
242             f"Feature Selection for {target}")
243
244     return results_df
245
246 # Feature selection for FSO_Att with visualization
247 fso_results = feature_selection(sampled_data, 'FSO_Att',
248     plot_cutoff=True)
249
250 # Feature selection for RFL_Att with visualization
251 rfl_results = feature_selection(sampled_data, 'RFL_Att',
252     plot_cutoff=True)
253
254 def train_random_forest_by_condition(data, target_column,
255     condition_column):
256     # Dictionary to store models for each condition
257     condition_models = {}
258     mean_rmse = []
259     mean_r2 = []
260
261     # Getting unique weather conditions
262     conditions = data[condition_column].unique()

```



```

258
259     for condition in conditions:
260         print(f"Training model for {condition} weather
261               condition.")
262
263         # Filtering data for the current condition
264         subset = data[data[condition_column] == condition]
265
266         # Splitting data into features (X) and target (y)
267         # Dropping target and condition columns
268         X = subset.drop(columns=[target_column,
269                                condition_column])
270         y = subset[target_column]
271
272         # Splitting into train and test sets
273         X_train, X_test, y_train, y_test =
274             train_test_split(X, y, test_size=0.2,
275                             random_state=42)
276
277         # Training Random Forest model
278         rf = RandomForestRegressor(random_state=42)
279         rf.fit(X_train, y_train)
280
281         # Evaluating the model
282         y_pred = rf.predict(X_test)
283         rmse = root_mean_squared_error(y_test, y_pred)
284         r2 = r2_score(y_test, y_pred)
285         print(f"RMSE for {condition}: {rmse:.4f}")
286         print(f"R^2 for {condition}: {r2:.4f}")
287
288         # Storing the trained model and metrics
289         condition_models[condition] = {
290             "model": rf,
291             "rmse": rmse,
292             "r2": r2
293         }
294         mean_rmse.append(rmse)
295         mean_r2.append(r2)
296
297     return {"model": condition_models, "rmse": np.mean(
298           mean_rmse), "r2": np.mean(mean_r2)}
299
300 # Extracting important features to the list based on
301 # manual cutoff index
302 fso_features = fso_results['Removed Feature'].iloc[12:].
303     tolist()
304 rfl_features = rfl_results['Removed Feature'].iloc[9:].
305     tolist()
306 fso_features, rfl_features

```

```
300 # Adding 'FSO_Att' and 'SYNOPCode' columns for splitting
    and training the FSO model
301 ds_fso_data = sampled_data[fso_features + ['FSO_Att', '
    SYNOPCode']]
302 ds_fso_data.head()
303
304 # Adding 'FSO_Att' and 'SYNOPCode' columns for splitting
    and training the RFL model
305 ds_rfl_data = sampled_data[rfl_features + ['RFL_Att', '
    SYNOPCode']]
306 ds_rfl_data.head()
307
308 # Training specific models for each weather conditions
309 fso_models = train_random_forest_by_condition(ds_fso_data,
    target_column='FSO_Att', condition_column='SYNOPCode')
310 rfl_models = train_random_forest_by_condition(ds_rfl_data,
    target_column='RFL_Att', condition_column='SYNOPCode')
311
312 def train_generic_random_forest(data, target_column):
313
314     # Splitting data into features (X) and target (y)
315     X = data.drop(columns=[target_column])
316     y = data[target_column]
317
318     # Splitting into train and test sets
319     X_train, X_test, y_train, y_test = train_test_split(X,
    y, test_size=0.2, random_state=42)
320
321     # Training Random Forest model
322     rf = RandomForestRegressor(random_state=42)
323     rf.fit(X_train, y_train)
324
325     # Evaluating model
326     y_pred = rf.predict(X_test)
327     rmse = root_mean_squared_error(y_test, y_pred) # Root
    Mean Squared Error
328     r2 = r2_score(y_test, y_pred) # R^2 score
329
330     print(f"RMSE: {rmse:.4f}")
331     print(f"R^2: {r2:.4f}")
332
333     return {"model": rf, "rmse": rmse, "r2": r2}
334
335 # Training the generic model
336 fso_results = train_generic_random_forest(ds_fso_data,
    target_column='FSO_Att')
337 rfl_results = train_generic_random_forest(ds_rfl_data,
    target_column='RFL_Att')
338
339 # Displaying the Mean RMSE and R2 scores for specific
    model
```

```

340 print(f"Mean RMSE for FSO: {fso_models['rmse']:.4f}")
341 print(f"Mean R^2 for FSO: {fso_models['r2']:.4f}")
342 print(f"Mean RMSE for RFL: {rfl_models['rmse']:.4f}")
343 print(f"Mean R^2 for RFL: {rfl_results['r2']:.4f}")
344
345 # Metrics for Method 1 (Specific Model)
346 fso_mean_rmse_method1 = fso_models['rmse']
347 fso_mean_r2_method1 = fso_models['r2']
348 rfl_mean_rmse_method1 = rfl_models['rmse']
349 rfl_mean_r2_method1 = rfl_models['r2']
350
351 # Metrics for Method 2 (Generic Model)
352 fso_rmse_method2 = fso_results['rmse']
353 fso_r2_method2 = fso_results['r2']
354 rfl_rmse_method2 = rfl_results['rmse']
355 rfl_r2_method2 = rfl_results['r2']
356
357 # RMSE Comparison Plot
358 plt.figure(figsize=(10, 6))
359 methods = ['FSO Method 1', 'FSO Method 2', 'RFL Method 1',
            'RFL Method 2']
360 rmse_values = [fso_mean_rmse_method1, fso_rmse_method2,
                rfl_mean_rmse_method1, rfl_rmse_method2]
361 plt.bar(methods, rmse_values, color=['blue', 'green', 'blue', 'green'])
362 plt.title('RMSE Comparison Between Methods')
363 plt.ylabel('RMSE')
364 plt.xticks(rotation=0)
365 plt.tight_layout()
366 plt.show()
367
368 # R2 Comparison Plot
369 plt.figure(figsize=(10, 6))
370 r2_values = [fso_mean_r2_method1, fso_r2_method2,
              rfl_mean_r2_method1, rfl_r2_method2]
371 plt.bar(methods, r2_values, color=['orange', 'purple', 'orange', 'purple'])
372 plt.title('R2 Comparison Between Methods')
373 plt.ylabel('R2')
374 plt.xticks(rotation=0)
375 plt.tight_layout()
376 plt.show()

```

Listing 1: Project Code for RF/FSO System Modeling

References

- Chowdhury, Mostafa Zaman et al. (2020). “Optical Wireless Hybrid Networks: Trends, Opportunities, Challenges, and Research Directions”. In: *IEEE Communications Surveys & Tutorials* 22.2. Licensed under Creative Commons Attribution 4.0, pp. 930–967. DOI: 10.1109/COMST.2020.2966855. URL: <https://doi.org/10.1109/COMST.2020.2966855>.
- Han, Boyu (2023). “A comprehensive review of performance analysis of RF-FSO hybrid communication systems”. In: *Journal of Physics: Conference Series* 2649.1. Published under licence by IOP Publishing Ltd, p. 012019. DOI: 10.1088/1742-6596/2649/1/012019. URL: <https://doi.org/10.1088/1742-6596/2649/1/012019>.
- ITU Radiocommunication Sector (2019). *Propagation Data and Prediction Methods Required for the Design of Earth-Space Telecommunication Systems*. Tech. rep. ITU-R Recommendation P.618-13. International Telecommunication Union.
- Nadeem, Farukh et al. (2010). “Weather Effects on Hybrid FSO/RF Communication Link”. In: *IEEE Journal on Selected Areas in Communications* 27.9, pp. 1687–1697. DOI: 10.1109/JSAC.2009.091215. URL: <https://doi.org/10.1109/JSAC.2009.091215>.
- Sector, ITU Radiocommunication (2012). *Propagation data required for the design of terrestrial free-space optical links*. Recommendation ITU-R P.1817-1. Accessed: [2024-11-19. International Telecommunication Union (ITU). URL: <https://www.itu.int>.