ECC791: Project II (A)

Deep Learning-Based Indoor Path Loss Prediction for WLAN at 5 GHz

11/28/2024

Under the Guidance of
Professor: Dr. Dalia Nandi, Dept. of ECE, IIIT Kalyani

By –

- Shirish Manoj Bobde ECE/21152/812
- Divyanshu Kumar ECE/21183/783
- Anubhav Ranjan ECE/21114/774

Declaration

We hereby declare that the work being presented in this thesis entitled, “Deep Learning-Based Indoor Path Loss Prediction for WLAN at 5 GHz”, submitted to Indian Institute of Information Technology, Kalyani in partial fulfillment for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering during the period from July, 2024 to November, 2024 under the supervision of Dr. Dalia Nandi, Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Kalyani, West Bengal 741235, India, does not contain any information and any piece of code and text generated by AI enabled software.

Shirish Manoj Bobde – ECE/21152/812
Divyanshu Kumar – ECE/21123/783
Anubhav Ranjan – ECE/21114/774

Department: Electronics and Communication Engineering
Institute: Indian Institute of Information Technology, Kalyani

This is to certify that the aforementioned statements made are correct to the best of my knowledge.

Signed**:**

Date:

Certificate

This is to certify that the thesis entitled "Deep Learning-Based Indoor Path Loss Prediction for WLAN at 5 GHz" being submitted by Shirish Manoj Bobde (ECE/21152/812), Divyanshu Kumar (ECE/21123/783) and Anubhav Ranjan (ECE/21114/774) in the Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Kalyani, West Bengal, India, for the award of Bachelor of Technology in Electronics and Communication Engineering is an original research work carried by them under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of Indian Institute of Information Technology, Kalyani and in my opinion, has reached the standards needed for submission. The work, techniques and the results presented have not been submitted to any other University or Institute for the award of any other degree or diploma.

Dr. Dalia Nandi

Assistant Professor

Department of Electronics and Communication Engineering

Indian Institute of Information Technology, Kalyani

Date:

Acknowledgements

We would like to express our gratitude to Dr. Dalia Nandi, our supervisor and mentor, for her invaluable guidance and insightful feedback throughout the project. We are sincerely thankful for the opportunity she has provided us to undertake this challenging project. We eagerly anticipate furthering our understanding of this technology through this experience.

Shirish Manoj Bobde (ECE/21152/812)

Divyanshu Kumar (ECE/21123/783)

Anubhav Ranjan (ECE/21114/774)

Department of Electronics and Communication Engineering Indian Institute of
Information Technology, Kalyani

Date:

Table of Contents

1. Introduction

Wireless Local Area Networks (WLAN) are essential for ensuring reliable connectivity in indoor environments, such as homes and offices. Accurate prediction of signal strength, or path loss, is critical for effective network planning, especially at higher frequencies like 5 GHz. However, predicting path loss in indoor settings is challenging due to the complex interactions between radio waves and obstacles like walls and furniture. Traditional models, such as the Free-Space and Log-distance models, often struggle to accurately represent these complexities, particularly at 5 GHz, where signal attenuation is more significant.

To address these limitations, this project proposes developing a deep learning-based model for predicting indoor path loss at 5 GHz. The model will be trained using data collected from a specific indoor layout, incorporating measurements from various receiver locations relative to a WLAN transmitter. Key factors, such as distance from the transmitter and the presence of obstacles, will be considered.

The goal is to create a model that offers improved prediction accuracy over traditional approaches, ultimately enhancing WLAN planning and deployment in indoor environments. This research will contribute to the field of wireless communication by demonstrating the potential of deep learning in high-frequency indoor path loss prediction.

2. Objective

2.1 To develop a deep learning-based model for predicting indoor Wi-Fi path loss with higher accuracy compared to traditional methods like 3D ray-tracing.

2.2 To create a convolutional neural network (CNN) model that can learn the features ofthe indoor environment and accurately predict the received signal strength (RSS) at various locations within a building.

2.3 To evaluate the performance of the deep learning model in different indoor scenarios, focusing on line-of-sight (LoS) and non-line-of-sight (NLoS) conditions.

3. Literature Review

Path loss modelling for wireless communication has evolved significantly with the advent of deterministic and machine learning approaches, particularly in complex environments like urban areas and high-frequency bands such as millimeter waves (mm Waves).

3.1 Traditional Path Loss Models

Deterministic models[8] have historically been central to path loss modelling, relying on ray tracing techniques to simulate how radio waves interact with the environment through reflection, refraction, and diffraction. While these models can provide accurate results, they often demand high computational resources and detailed environmental data, making them less practical for widespread use. On the other hand, empirical models like the Close-In (CI) and Alpha-Beta-Gamma (ABG) models simplify the prediction process by using statistical relationships derived from measurement data. Although these models are computationally efficient, their accuracy is heavily dependent on the similarity between the measured environment and the target scenario, limiting their generalizability.

3.2 Machine Learning Approaches

To overcome the limitations of traditional models, machine learning (ML) techniques have been increasingly adopted in path loss modelling. ML[6] models, such as neural networks and support vector machines, can capture complex relationships between environmental features and path loss without requiring explicit programming of the underlying physics. Convolutional neural networks (CNNs), in particular, have shown promise in processing environmental data to predict path loss with greater accuracy than traditional models. Hybrid models that combine empirical methods with ML techniques have also been developed. These models often use ML algorithms to optimize empirical models' parameters or select the most relevant environmental features. While hybrid models outperform purely empirical approaches, they are still surpassed by fully data-driven ML models in terms of accuracy.

3.3 Deep Learning Innovations

Deep learning, a subset of ML, has revolutionized path loss modelling by enabling automatic feature extraction from raw data. CNNs[3] have been especially effective in improving prediction accuracy in complex environments, such as urban areas. Techniques like attention mechanisms and dilated convolutions have further enhanced CNNs' performance by allowing models to focus on critical input data areas and capture broader context. A notable example is the Attention-Enhanced Convolutional Neural Network (AE-CNN)[1] model, which integrates dilated convolutions and global context blocks to improve feature extraction. This model has demonstrated superior performance in predicting path loss for 28 GHz mm Wave communications in suburban environments, outperforming previous empirical and deterministic methods.

3.4 Comparative Performance Analysis

Comparative studies have consistently shown that deep learning-based models, particularly CNNs with advanced techniques like attention mechanisms, achieve the lowest root mean square error (RMSE) in both line-of-sight (LoS) and non-line-of-sight (NLoS) scenarios. These models excel in environments where traditional models struggle, such as urban canyons or areas with dense vegetation. Innovations like multi-way local attentive learning have been introduced to refine the training process of CNN models, enabling them to learn from multiple perspectives and improve their generalization capabilities. This method has significantly enhanced CNN models' performance in path loss prediction tasks.

3.5 Conclusion

The shift from deterministic and empirical models to machine learning and deep learning techniques marks a significant advancement in wireless communication. Deep learning models, especially those incorporating attention mechanisms and dilated convolutions, represent the forefront of path loss modelling, offering unmatched accuracy and adaptability. As communication technologies continue to evolve, these models will play a crucial role in optimizing network design and performance, particularly in challenging environments.

4. Methodology

4.1 Layout Generation:

We used an indoor environment based on the layout shown in Fig. 1, which represents a well-defined area that includes various structural elements such as walls, open spaces, and obstacles like beds, tables, and the kitchen slab. Three different layouts were created for distinct purposes. The first layout (Fig. 1a) provides an overview of the flat’s structure and boundaries with precise measurements. The second layout (Fig. 1b) depicts the flat's major elements, where the grid represents the tiles, and the center of each tile serves as a receiver point during data collection. This layout was also used for CNN input image generation using the LAMS algorithm for all transmitter-receiver pairs. Lastly, the third layout (Fig. 1c) offers a 3D visualization, enhancing our spatial understanding of the flat.

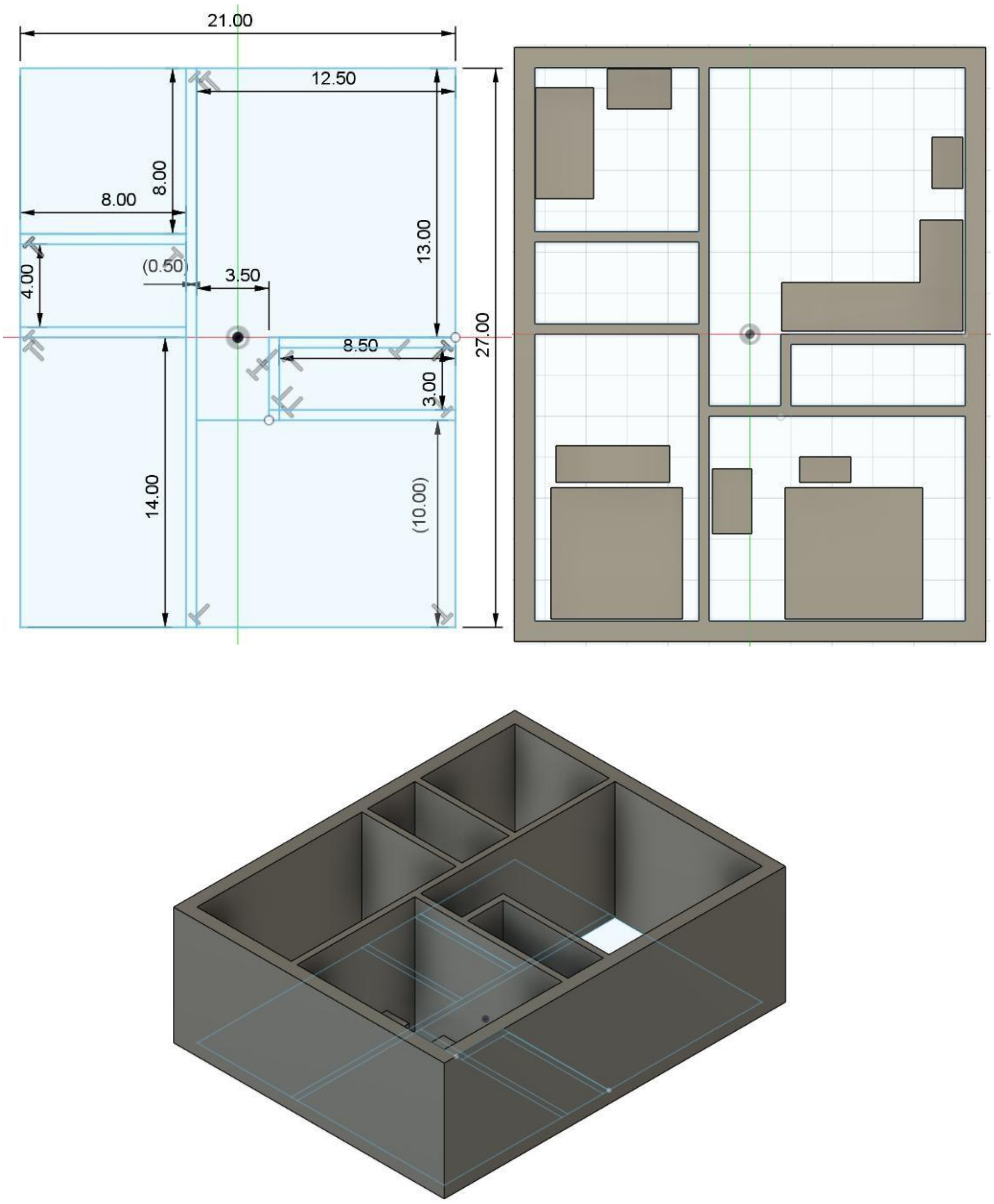


Fig. 1. a) Layout of indoor environment with measurements.
b) 2D layout with measurement grid.
c) 3D layout.

4.2 System Design:

The architecture of the proposed deep learning-based path loss modeling system is illustrated in Fig. 2. The measurement data include the locations of the APs and the measurements as well as the corresponding RSS values. The main input of the proposed CNN path loss model is images generated by a novel input image generation algorithm known as the LAMS algorithm. The LAMS algorithm aims to extract the environmental features from the simplified floor plan. Since the TR separation plays a significant role in path loss modeling, a distance neuron is appended to the first FC layer to make explicit the distance information to the CNN model. The CNN path loss model is trained to predict RSS based on the LAMS images and the TR-separation in the training phase. RMSE is used as the loss function of the CNN model. The trained CNN model can predict the RSS of a given location with the LAMS image and the TR-separation.

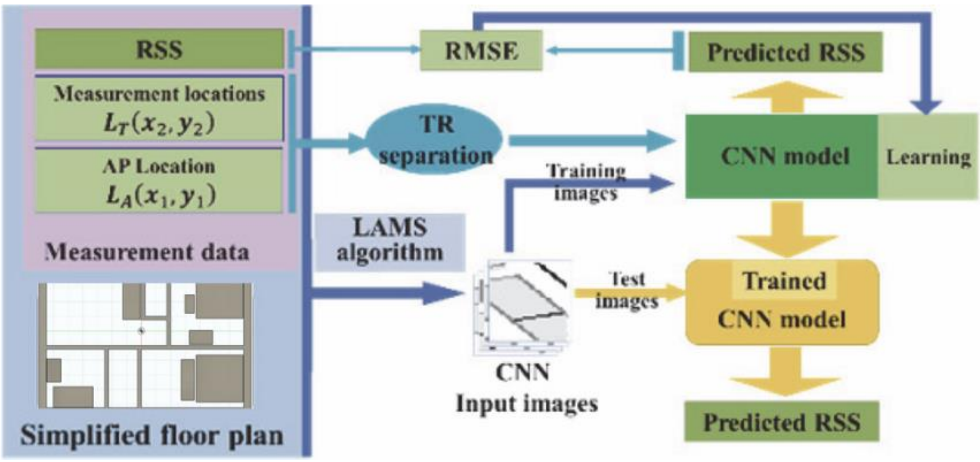


Fig. 2. Flow diagram of deep learning-based path loss modeling system.

4.3 LAMS Algorithm Development:

The Local Area Multi-Line Scanning (LAMS) algorithm was developed to generate input images for a CNN model, providing essential environmental features that influence indoor path loss. This algorithm creates a unique representation of the indoor space by generating multiple scan lines between a Wi-Fi access point (AP) and a receiver location. These scan lines capture details about obstacles such as walls and room divisions, which affect signal propagation. Here’s an overview of the LAMS algorithm development and its functionality:

- **Algorithm Overview**: The LAMS algorithm begins by identifying the AP and receiver coordinates on a floor plan image. It then calculates a line connecting these two points, along which a series of scan lines are generated perpendicularly. Each scan line is defined by a specific width (number of pixels, denoted as w) and the total number of lines(n), capturing spatial details between the AP and receiver locations.

Inputs:

Location of AP point T

Location of measurement R

Size of generated input image (w, n)

Image matrix of simplified floor plan I_F

Calculate the linear equation of l_{TR}

Add n **points on** l_{TR} **and determine the coordinates of these points**

for $i = 1:1:n$

Calculate the endpoint coordinates of each scan line

Copy w pixel values of the scan line on I_F and save them as s_i

end for

Integrate $s_i, i = 1,2,3,...,n$ **as a matrix** I_{LAMS}

Save the matrix I_{LAMS} **as an input image**

- **Code Implementation**:
  1. **Setup and Parameters**: The algorithm uses the input floor plan as a grayscale image, where walls and obstacles are marked with distinct pixel values. User clicks define the AP and receiver coordinates.
  2. **Line Generation**: Using the coordinates of the AP and receiver, the algorithm calculates intermediate points on the line between them. These points serve as the center of each perpendicular scan line.
  3. **Scan Line Creation**: For each center point, the algorithm calculates the perpendicular direction and establishes the endpoints of the scan line. Pixel values along each scan line are extracted from the floor plan image.
  4. **Output Image Generation**: These pixel values are compiled into a matrix (lams_image), representing a simplified view of the environment, suitable for CNN input.

- **Image Visualization and Validation**: After generating the lams_image, it is displayed and saved for model input. For visualization purposes, the algorithm also draws the scan lines directly onto the floor plan image to validate the accuracy and positioning of each scan.
- **Challenges and Refinements**: Key challenges included handling boundary cases where scan lines extended beyond the floor plan's edges. This was mitigated by checking each scan line's coordinates and setting out-of-bounds values to a default empty-space pixel value.

The LAMS algorithm thus enables an effective representation of indoor environmental features, facilitating CNN model training for accurate path loss prediction.

4.4 Data Collection:

Data collection was conducted in a controlled indoor environment, based on the floor plan layout provided. This layout includes multiple rooms and obstacles, where each grid cell represents a tile in the flat, with the center of each tile designated as a receiver point. The purpose of this setup was to accurately capture signal behavior across the apartment under static conditions, which are crucial for Wi-Fi path loss modeling.

A single Wi-Fi access point (AP) was set up at pixel coordinates (312, 350) on the floor plan to act as the transmitter. Using a laptop placed on a stationary chair, data was gathered at 84 receiver points spread across the layout, each located at the center of a tile. Data was collected in intervals of 10 seconds, with 10 measurements taken at each receiver point in order to average out the time fading effect, ensuring data consistency and reducing variability.

The **Flutter Junction Test app** was used to automate the data collection process. The app was configured with a 10-second interval between samples, and each collection session generated an Excel file with the following attributes:

i. **Timestamp**: The exact time of each measurement.
ii. **Download Speed** and **Download Speed (Additional)**: Measures of network performance.
iii. **Signal Strength** and **Signal Strength (Additional)**: Indicators of Wi-Fi signal strength at eachreceiver point.
iv. **Network Type**: The type of network being measured.
v. **Device Status**: Status of the device during measurement.
vi. **Pixel Latitude (X)** and **Pixel Longitude (Y)**: Coordinates on the floor plan image where eachmeasurement was taken.

After collecting data for two minutes at each receiver point, any excess data was removed, retaining only the required measurements. The app output was then saved in an organized format. To determine the exact coordinates for each receiver point, a custom code was used, allowing users to click on the floor plan image to record pixel coordinates. These coordinates were then added as **Pixel Latitude** and **Pixel Longitude** entries in the Excel file.

Once data was fully documented for each receiver point, we moved to the next point and repeated the process. The measurements were taken in a completely static environment, with no human movement or other significant signal interference sources. All doors were closed during data collection to isolate the influence of walls and static obstacles on signal Propagation.

This setup was designed to capture realistic signal degradation caused by physical barriers, room configurations, and layout geometry, which are significant contributors to indoor path loss. The data collected provides a comprehensive foundation for training and testing the CNN model, as it reflects a diverse range of conditions within the enclosed space. The static nature of the environment helps eliminate variability due to transient factors, focusing purely on the spatial and structural elements affecting signal propagation.

Sample Data:

	A	B	C	D	E	F	G	H	I
1	Timestamp	Download Speed	Download Speed	Signal Strength	Signal Strength (/	Network Type	Device Status	Pixel Latitude (X)	Pixel Longitude (Y)
2	2024-10-31 8:57:	887.1388989825582 Mbps		-12.0 dBm		Wi-Fi	Unknown	207	364
3	2024-10-31 8:58:	1105.7093523550723 Mbps		-9.0 dBm		Wi-Fi	Unknown	207	364
4	2024-10-31 8:58:	1121.969784007353 Mbps		-9.0 dBm		Wi-Fi	Unknown	207	364
5	2024-10-31 8:58:	887.1388989825582 Mbps		-10.0 dBm		Wi-Fi	Unknown	207	364

Table 1: Sample of Collected Data

4.5 Data Cleaning and Pre-Processing:

Google **Colab** Notebook is used for data cleaning and pre-processing. Multiple .**xlsx** files containing signal strength data were loaded using Pandas and concatenated into a single dataset for consistency and easier processing. Missing values labeled as "**Unknown**" were replaced with **NaN** to standardize the data. Irrelevant columns, such as "**Download Speed** (**Additional**)" and "**Signal Strength** (**Additional**)," were dropped to reduce noise. Numerical columns, including Download Speed and Signal Strength, were cleaned by removing non-numeric characters like units (e.g., "**dBm**") to ensure uniformity. These columns were then converted to numeric data types, with invalid entries set to **NaN** to facilitate accurate calculations and prevent errors during processing.

4.6 Data Processing:

Google Colab Notebook is used for data processing, starting with the creation of a new Coordinate column by combining the **Pixel Latitude (X)** and **Pixel Longitude (Y)** columns to represent each unique location point. This facilitated grouping and averaging in subsequent steps. The dataset was then grouped by the newly created Coordinate column to analyze each location individually. For each coordinate, the average signal strength (in dBm) and download speed (in Mbps) were calculated, resulting in a cleaner, aggregated dataset. Additional attributes, including Pixel Latitude (X), Pixel Longitude (Y), and Network Type, were retained. Finally, the processed data was previewed to ensure accuracy by verifying dimensions and sample values.

Coordinate	Pixel_Latitude_X	Pixel_Longitude_Y	Network_Type	avg_signal_strength_dBm	avg_download_speed_Mbps
111,123	111	123	Wi-Fi	-18.8	786.2142878715831
111,171	111	171	Wi-Fi	-17.6	717.9304577871408
111,210	111	210	Wi-Fi	-18.0	731.0895543279912
111,268	111	268	Wi-Fi	-18.9	851.1669089803106
111,316	111	316	Wi-Fi	-16.6	962.6044218455916
111,364	111	364	Wi-Fi	-13.6	754.1367786511893
111,412	111	412	Wi-Fi	-14.1	663.615132882559
111,460	111	460	Wi-Fi	-13.8	594.8495769993293
159,123	159	123	Wi-Fi	-16.8	728.0215178267051
159,171	159	171	Wi-Fi	-19.7	634.5391571761328

Table 2: Processed Data

4.7 Data Visualization:

We have used Google Colab Notebook for Data Visualization.

- **Signal Strength Heatmap:**
  - A heatmap was generated to visualize signal strength across different coordinates. Each data point's color intensity represented the signal strength (in dBm), allowing for a spatial overview of signal distribution within the area.
  - Pixel Latitude (X) and Pixel Longitude (Y) were used as the axes, with a color bar to indicate signal strength variations.

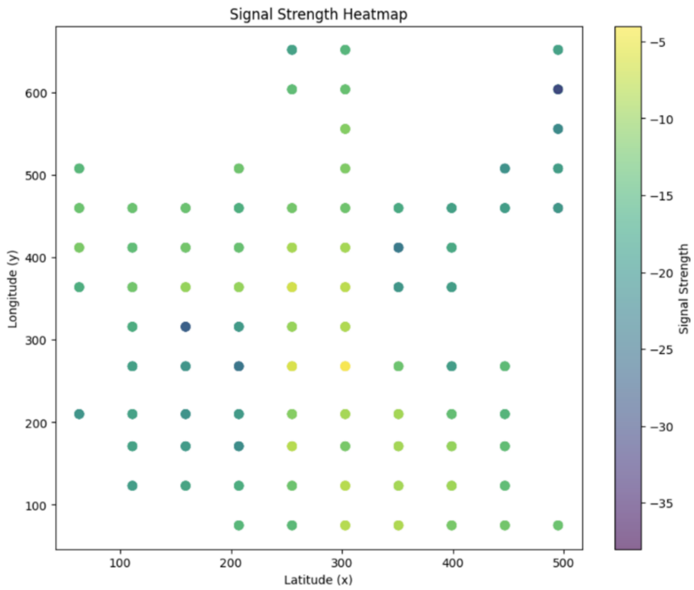


Fig. 3. Signal Strength Heatmap for pixel points(dBm)

- **Download Speed Heatmap:**
  - A similar heatmap was created to display download speed variations across the samecoordinates.
  - This visualization provided a quick view of areas with higher or lower download speeds,aiding in understanding connectivity patterns in the space.

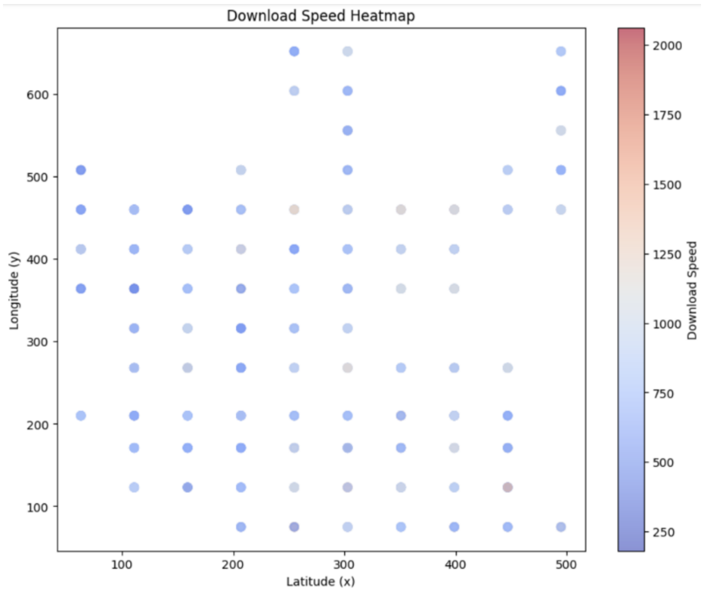


Fig. 4. Download Speed Heatmap for pixel points (Mbps)

- **Overlaying on Floor Plan:**
  - Both signal strength and download speed heatmaps were overlaid on a floor plan image toprovide spatial context.
  - Using Seaborn and Matplotlib, the heatmaps were superimposed with some transparency, allowing underlying room structures to remain visible. This integration facilitated a better understanding of environmental influences on connectivity metrics.

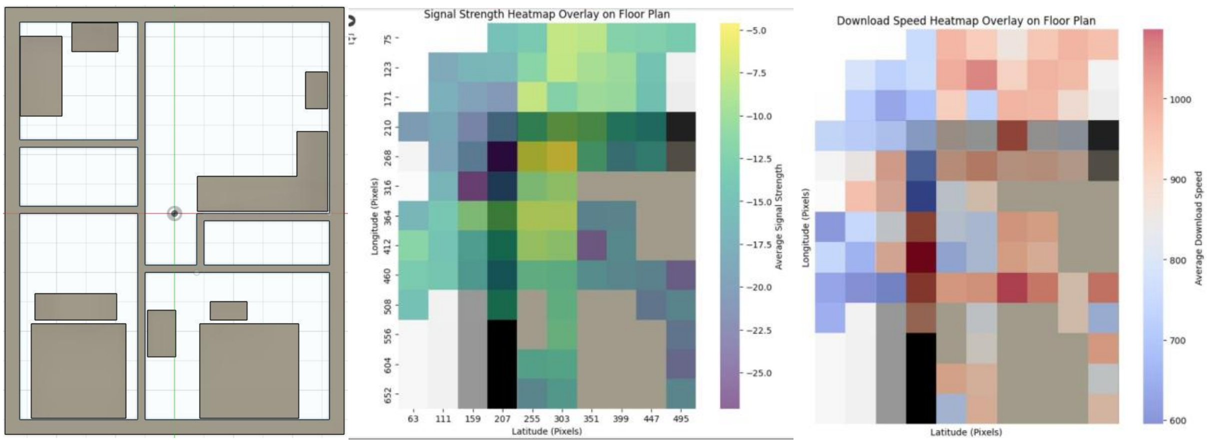


Fig. 5. a) 2D layout of the Indoor Environment (Floor Plan).
b) Signal Strength Heatmap Overlay on Floor Plan (dBm)
c) Download Speed Heatmap Overlay on Floor Plan (Mbps)

4.8 CNN Input Images Generation: LAMS Algorithm:

In our project, we developed the **Local Area Multi-Line Scanning (LAMS) algorithm** to generate input images for the Convolutional Neural Network (CNN), encoding environmental data related to Wi-Fi signal propagation in the indoor area under study. This algorithm extracts significant environmental features, such as wall structures, from a building's floor plan and translates them into grayscale images suitable for CNN-based analysis.

- **Setting up the Environment and Folders**: We initialized the environment by importing required libraries such as NumPy, OpenCV, and pandas, and created directories to save generated images: "CNN Input Images" and "Lams Scan Images."

- **LAMS Algorithm Process**: The LAMS algorithm was implemented to capture relevant environmental features from a fixed access point (AP) to multiple receiver locations across the floor plan:
  - **Defining Transmitter and Receiver Coordinates**: The algorithm initializes the AP coordinates and iteratively sets the coordinates for each receiver point based on its location in the building.
  - **Generating Scan Lines**: For each AP-receiver pair, a set of scan lines perpendicular to the line connecting the AP and receiver is generated. These lines are spaced evenly to form an image matrix that captures building structure details, such as walls and obstacles, in grayscale intensities.
  - **Sampling Pixel Values**: Pixel intensities along each scan line are extracted from the floor plan, capturing structural information at different points between the AP and receiver. The LAMS image is formed by stacking these sampled values into a 2D matrix.
  - **Saving Output Images**: Two sets of images are saved for each receiver location:
    1. **LAMS Image**: The grayscale LAMS image representing environmental features isstored in the "CNN Input Images" folder.

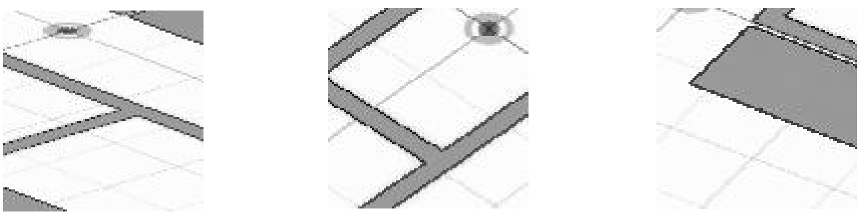


Fig. 6. Lams Output Image

2. **Annotated Floor Plan**: The floor plan is overlaid with the generated scan lines andsaved in the "Lams Scan Images" folder for visual reference.

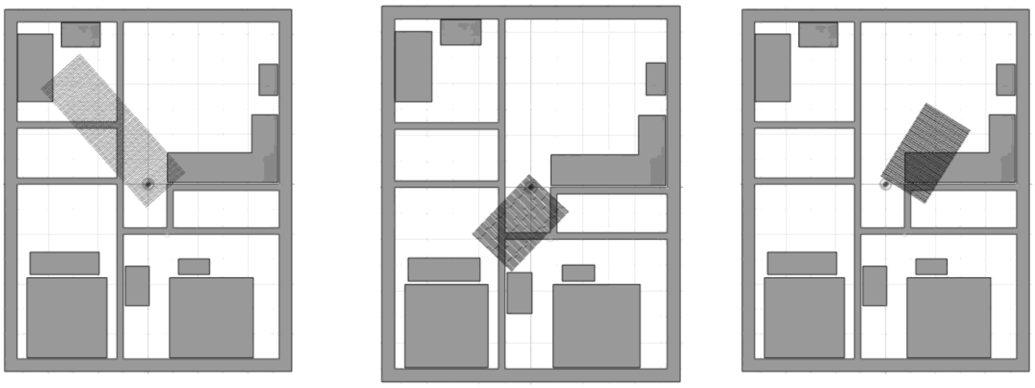


Fig. 7. Lams Scan Images

- **Calculating and Storing TR Separation**: The **Transmitter-Receiver (TR) separation** is calculated as the Euclidean distance between the AP and each receiver location. This distance information is appended to the dataset and saved in a CSV file for later use in model training.
- **Output**: All LAMS images and annotated floor plans were successfully saved, and the TR separation was added to the dataset, allowing for structured input to the CNN model.
- This approach enables the CNN to leverage spatial environmental context for signal strength prediction, transforming spatial layout into a form the model can interpret effectively.

4.9 CNN MODEL DESIGN:

A CNN[1] can learn the underlying functions of the Wi-Fi signal propagation by the automatic feature extraction of the convolutional layers. The distance neuron is appended to the flattened feature map that represents the path loss relevant features in the first FC layer to predict RSS. In this case, the size of the CNN input images is 100 × 100, which is designed to maintain the spatial information of the path loss environment and the directivity from the APs to the measurement locations. There is no pooling layer between two adjacent convolutional layers. In general, the usage of pooling can decrease the parameters of the network, and it can thus accelerate the speed of training and avoid the overfitting problem. For example, in computer vision problems, the pooling operation helps CNN achieve spatial invariance. However, the introduction of pooling layers can lead to a loss of essential information in the input. In our case, we are attempting to maintain the spatial information of the measurement environment as well as the directivity from the AP to the measurement points. The size of the CNN input images in our experiment is 100 × 100, and they are simple small grey images. If there is no pooling layer after the convolutional layers, the size of activation maps can be kept unchanged, and useful features can thus be expected to be extracted from the CNN input images, which can be good for the reasoning and calculation on the FC layers. Consequently, pooling layers are not used in our CNN models. The prediction performance of a CNN model is primarily dependent on its structure, such as the number of convolutional layers and FC layers, and the number and size of filters.

Parameters of convolutional layer

Layer Type	Input Channel	Output Channel	Kernel Size	Stride Size
C1	1	3	6	1
C2	3	6	4	3
C3	6	3	3	2
Cmj	3	3	2	1
Cf	3	2	2	2

Table 3: Five Types of Convolutional Layers

Table details the five types of convolutional layers, which are denoted as C1, C2, C3, Cmj, and Cf . C1, C2, C3 and Cf are the first, second, third, and final convolutional layers, respectively. Cmj is the jth Cm convolutional layer, where Cm indicates that it is between the C2 and Cf layers, and it is a middle convolutional layer. Using the "SAME" padding method, the activation maps of the input and output of Cmj ($j > 1$) layers are of the same size. Thus, the size of the activation maps can be maintained, which is conducive to the implementation of deeper CNN structures. Zero-padding is also introduced. The padding method is set as "VALID" in the first two and the last convolutional layers, while "SAME" padding is used in the rest of the convolutional layers. "VALID" means there is no padding, which can make the size of the activation map shrink fast. "SAME" padding can make the spatial size of the output volumes the same as the input volumes. "SAME" padding tries to pad evenly left and right, but if the number of columns to be added is odd, it will add an extra column to the right. The activation functions used in the convolutional layers and FCs are the ReLU function. The range of the measurement RSS values is between -100 dBm and -10 dBm. Due to the unilateral suppression attribute of ReLU, the negative measurement RSS values cannot be directly used as labels. Therefore, the label data should be shifted into positive values. For example, if the RSS value is -30 dBm, the label value will be 70 if we add 100 to it. The loss function used in the experiment is the RMSE. Another design decision is the choice of the training algorithm. Here, the Adam algorithm is chosen (learning rate $lr = 0.001$, $beta1 = 0.9$, $beta2 = 0.99$, $\varepsilon = e{-8}$) . TensorFlow is used to implement our CNN model. The CNN model has six convolutional layers (including two Cm layers), and seven and three neurons in the first and second hidden FC layers, respectively.

4.10 TRAINING PROCESS:

i. Data Preparation

- **Load and preprocess data**:
  - Load the CSV file containing Pixel_Latitude_X, Pixel_Longitude_Y, TR_separation, and avg_signal_strength_dBm.
  - Load grayscale 100×100 images corresponding to each measurement location.
  - Normalize the images to [0, 1] and expand dimensions for channel compatibility.

    - Adjust avg_signal_strength_dBm values by shifting them by +100 to ensure all target values are positive.
- **Split data**:
    - Split the data into training (90%) and testing (10%) datasets using train_test_split.

ii. **Data Generator**

- Define a custom generator class (DataGenerator) to:
    - Batch-process the images, TR separation values, and RSS labels.
    - Optionally shuffle the data at the end of each epoch.
- Use TensorFlow's tf.data.Dataset API for efficient prefetching and loading:
    - Create datasets for training and testing using the generator.

iii. **Model Architecture**

- **Convolutional Layers**:
    - Use convolutional layers[10] with specified configurations (e.g., stride, kernel size, padding) to extract spatial features.
    - Avoid pooling layers to preserve spatial information of the path loss environment.
    - Use ReLU activation for non-linear feature extraction.
- **Distance Neuron**:
    - Input the TR separation value as a separate feature.
    - Concatenate the flattened convolutional output with the TR separation input.
- **Fully Connected Layers**:
    - Two hidden FC layers with 7 and 3 neurons respectively.
    - A final output layer with a single neuron for RSS prediction.
- **Compilation**:
    - Use Adam[9] optimizer with specified hyperparameters (lr=0.001, beta1=0.9, beta2=0.99, epsilon=1e-6).
    - Loss function: Mean Squared Error (MSE).
    - Metric: Root Mean Squared Error (RMSE).

iv. **Training Configuration**

- **Train the model**:
    - Use the train_dataset for training and test_dataset for validation.
    - Run the training process for 50 epochs, monitoring loss and RMSE.
    - Set verbose=1 to track progress.
- **Track performance**:
    - Save the training history for visualizing metrics (e.g., training and validation loss).

v. **Save and Evaluate**

- **Save the trained model**:
    - Export the model after training using TensorFlow's model.save() for future use.
- **Evaluate performance**:
    - Use the test_dataset to evaluate the RMSE on unseen data.
    - Visualize the predictions versus actual RSS values for additional insight.

5. Experimental Results:

5.1 Model Training and Evaluation Results

The model was trained using 50 epochs, with a training dataset and a validation dataset. The following key results were observed:

- Test Loss: 27.907129287719727
- Test RMSE: 5.28272008895874

```
[35] # 7. Evaluate the Model
     loss, rmse = model.evaluate(test_dataset)
     print(f"Test Loss: {loss}, Test RMSE: {rmse}")

     2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 33.7100 - root_mean_squared_error: 5.7657
     Test Loss: 27.907129287719727, Test RMSE: 5.28272008895874
```

Fig. 8. Model Evaluation Result

The training and validation loss curves indicate how the model's performance evolved during training:

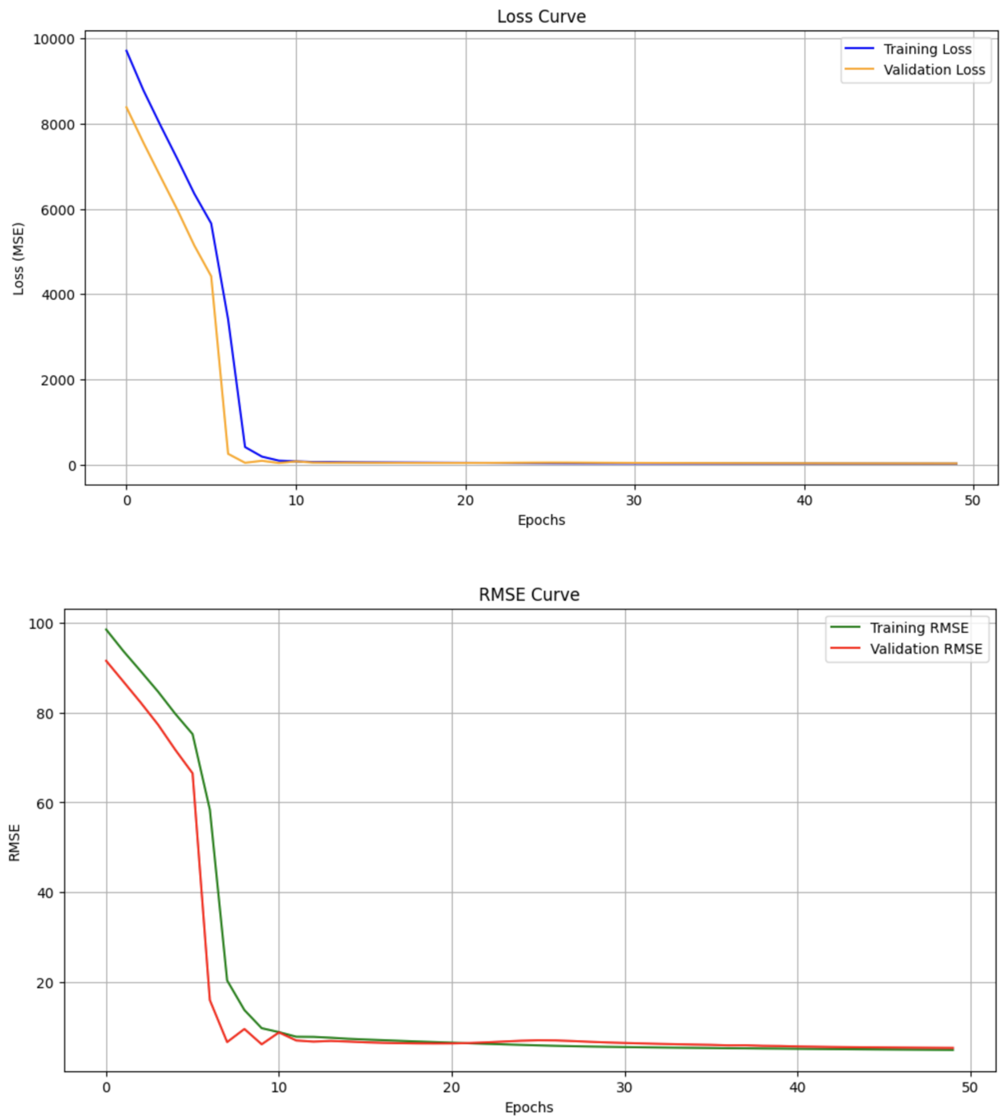


Fig. 9. a) Loss Curve for Training and Validation.
b) RMSE Curve for Training and Validation.

5.2 Predicted RSS Heatmap

The heatmap below visualizes the predicted RSS (Received Signal Strength) over the floor plan. Each pixel represents a coordinate point with the corresponding RSS value, providing insight into signal strength distribution within the test environment.

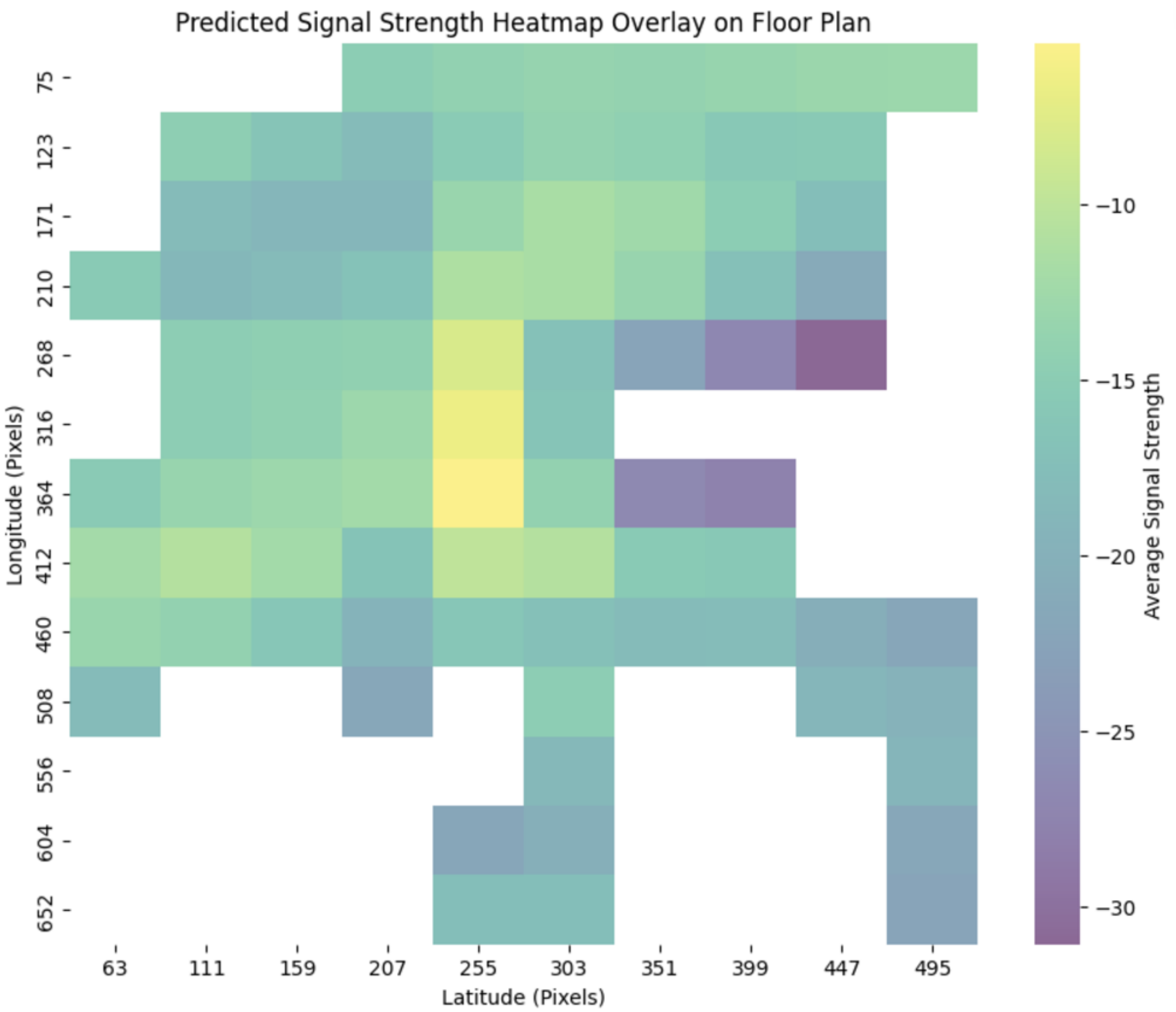


Fig. 10. Predicted Signal Strength Heatmap(dBm)

5.3 Measured vs. Predicted RSS Curve

Scatter plots and curve fits demonstrate the relationship between TR (Transmitter-Receiver) separation and RSS values for both measured and predicted data.

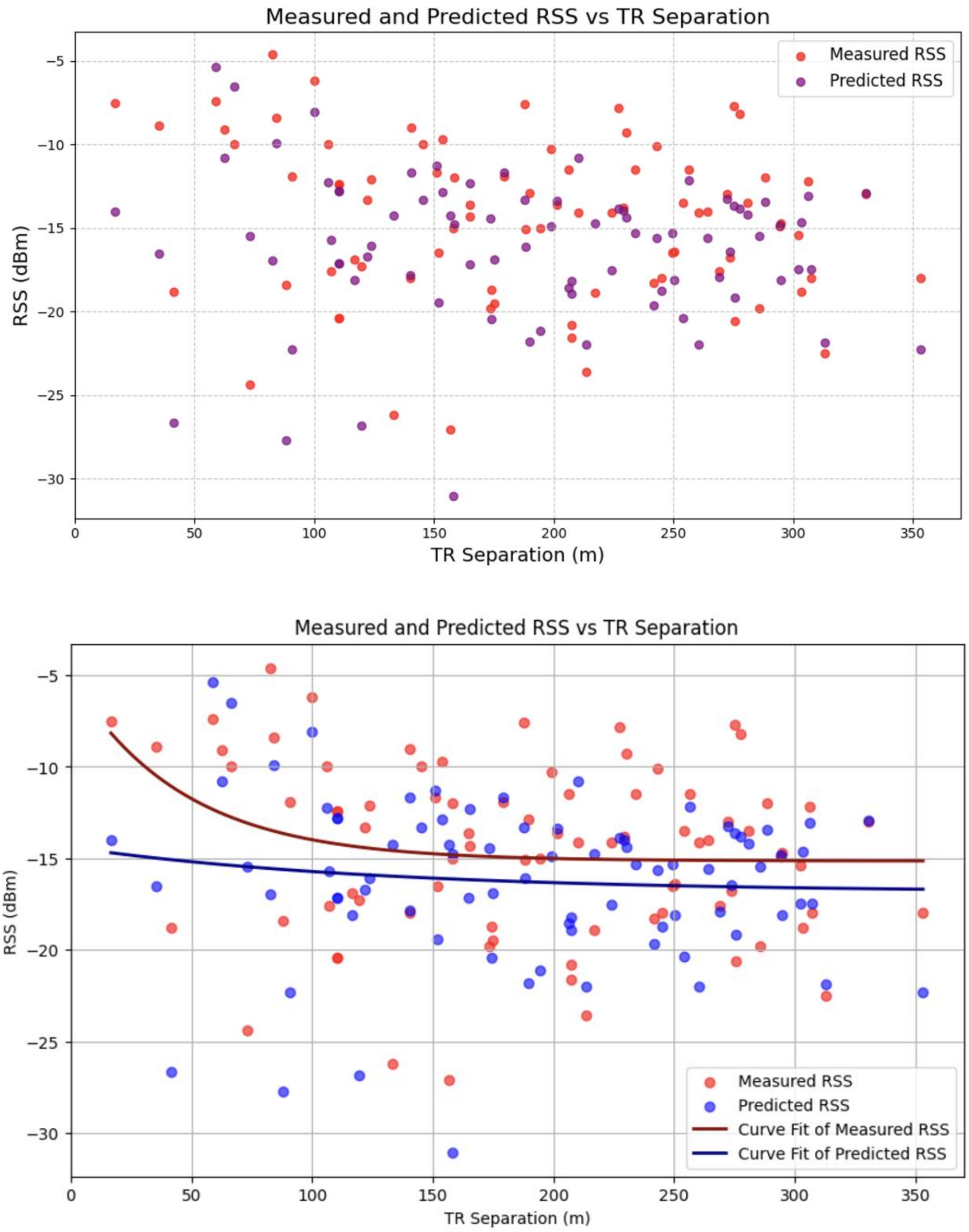


Fig. 11. Measured and Predicted RSS vs. TR Separation.

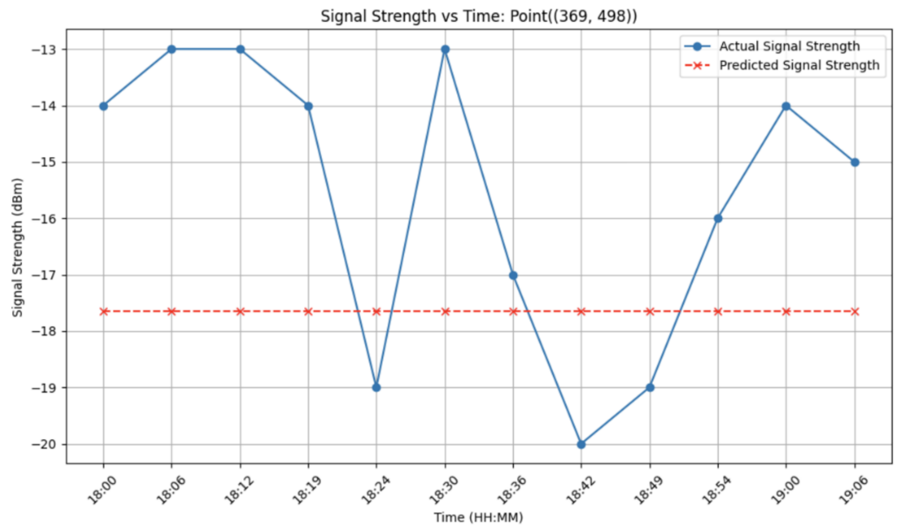
a) Scatter plot
b) Curve fitted plot

These results demonstrate the accuracy of the model in predicting RSS values, with visually close adherence between measured and predicted curves.

5.4 Visualization of Signal Strength Over Time

For few datasets, signal strength over a time span of 1 hour was visualized, comparing actual measured values with CNN-predicted values.

Case 1: Transmitter and Receiver are in different rooms



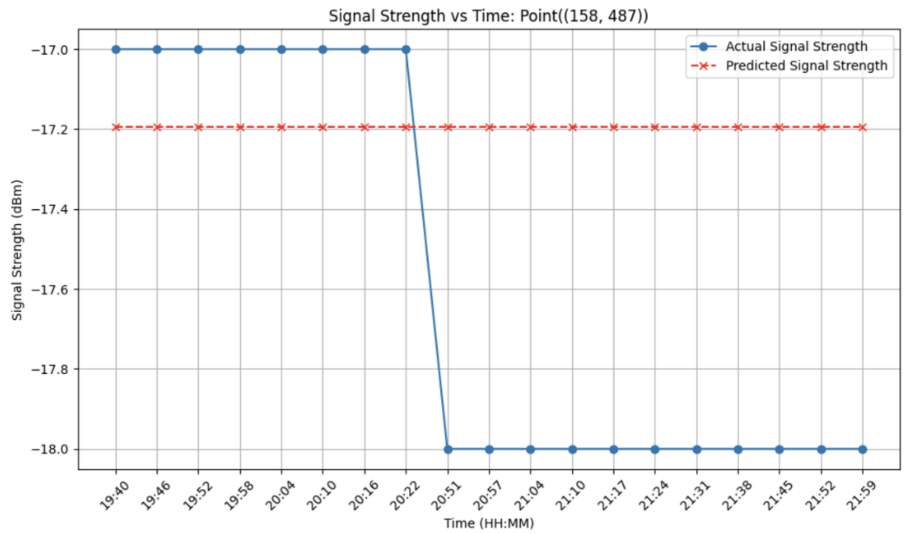
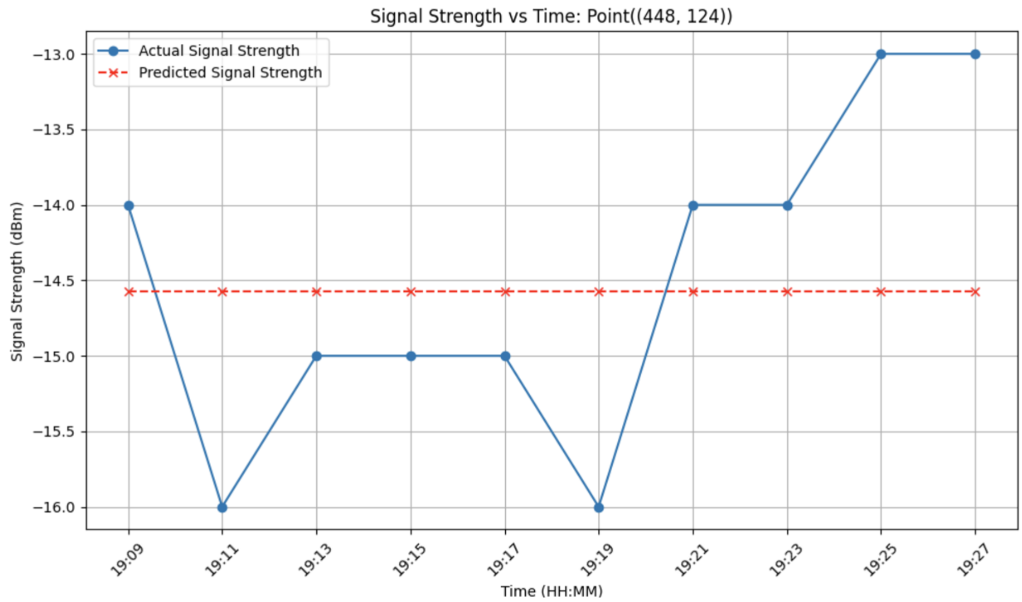


Fig. 12. a) Predicted and Measured Signal Strength vs Time in room 1
b) Predicted and Measured Signal Strength vs Time in room 2

Case 2: Transmitter and Receiver are in same room



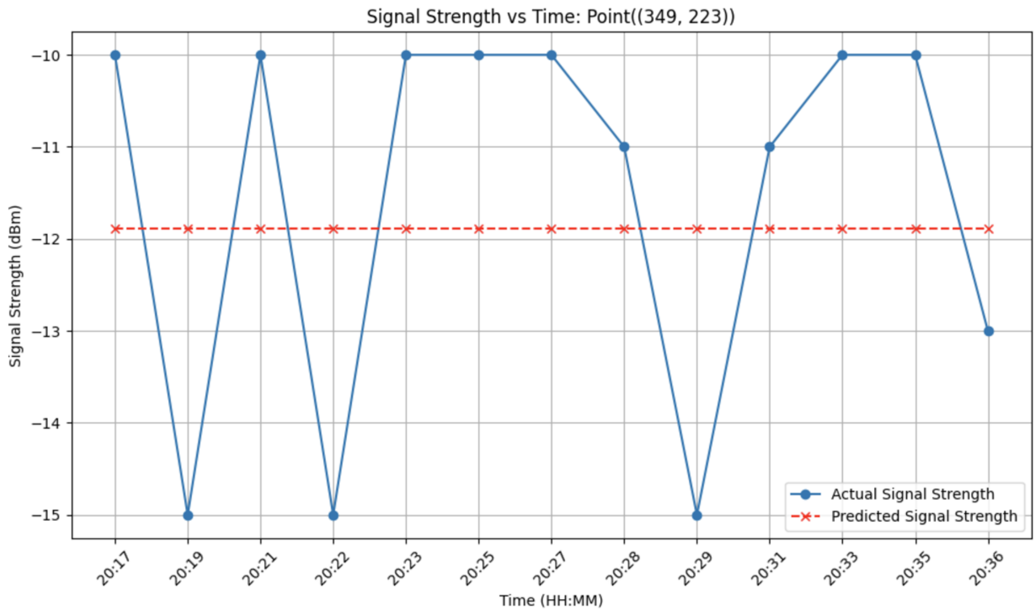


Fig. 13. Predicted and Measured Signal Strength vs Time in same room

Key Observations

1. **Accuracy of Predictions**:
   - The model demonstrated high accuracy in predicting signal strength, with minimal deviation during most time intervals.
2. **Deviations and Errors**:
   - Notable deviations were observed during certain periods, likely due to environmental factors not accounted for in the model.

5.5 Summary of Experimental Findings

1. **Prediction Accuracy:** The predicted RSS closely matches the measured values across varying TR separation distances.
2. **Heatmap Distribution:** The RSS distribution provides a clear overview of signal strength variations, aiding in spatial analysis.
3. **Model Robustness:** The model shows stable training and validation performance, with minimal overfitting, as evident from the loss and RMSE plots.

This analysis underscores the effectiveness of the CNN model in predicting signal strength within a structured environment based on TR separation and LAMS image data.

This methodology outlines a comprehensive approach to building a deep learning-based indoor path loss model, leveraging CNN techniques and the LAMS algorithm. The focus is on creating robust models tailored to specific indoor environments, ensuring their effectiveness in real-world applications.

6. Challenges:

Throughout this project, several challenges were encountered that required careful consideration and problem-solving:

i. **Data Collection:** The initial stages of data collection were particularly challenging. To address this, we created a floor plan and divided it into a 2x2 feet grid map, which provided a structured approach for collecting data at specific locations.
ii. **Data Preprocessing:** The collected data contained a significant amount of unnecessary information, making it crucial to preprocess and clean the dataset before it could be fed into the algorithm. This step required filtering out irrelevant data to improve the quality and accuracy of our model.
iii. **Lack of Predefined Coordinates:** In indoor environments, we don't have predefined longitude and latitude coordinates, which posed a challenge in finding an alternative method for mapping the data accurately within the floor plan.
iv. **CNN Model Architecture:** Designing the architecture for the Convolutional Neural Network (CNN) was another hurdle. Determining the optimal number of layers, kernel size, and strides required extensive experimentation to identify the best configuration for our task.
v. **Initial Model Training Issues:** During the initial model training, a pooling layer was included, which inadvertently removed important features from the data. This resulted in higher root mean square errors (RMSE), but once the pooling layer was removed, the model performance improved significantly.
vi. **Parameter Tuning:** Tuning the parameters of the CNN model proved to be a challenging task. Finding the right combination of learning rates, batch sizes, and other hyperparameters required several iterations and experimentation to achieve optimal results.
vii. **Implementing LAMS Algorithm:** Integrating the Local Area Multi-line Scanning (LAMS) algorithm into the code presented its own set of challenges, requiring careful implementation to ensure its functionality aligned with the overall project goals.

7. Conclusion & Future Scope:

This project aims to develop a deep learning-based model for predicting indoor path loss at a 5 GHz frequency, using a fixed transmitter and receiver setup within a well-defined indoor layout. The innovative use of the Local Area Multi-Line Scanning (LAMS) algorithm allows for the generation of detailed input images that accurately represent the signal propagation environment. These images, combined with distance information, will be fed into a Convolutional Neural Network (CNN) designed to predict Received Signal Strength (RSS)values with high accuracy.

By creating CNN model tailored for general and complex indoor environments, the project ensures robust performance across different scenarios. The methodology focuses on minimizing prediction errors and improving upon traditional path loss models. The model will be evaluated based on their ability to accurately predict path loss, with the Root Mean Square Error (RMSE) serving as the primary performance metric.

This work not only advances the field of indoor path loss modelling but also sets the stage for future research, including applications in higher frequency bands and more dynamic environments. Ultimately, the project aims to enhance WLAN planning and deployment, leading to improved wireless communication in complex indoor settings. Currently, the project focuses on static transmitter and receiver positions but can be extended to work with dynamic transmitter and receiver locations, enabling broader application in real-time scenarios. At present, the model considers only free-space path loss and shadowing factors, but future work could incorporate random fading effects using a larger dataset, allowing for more accurate and realistic predictions. Due to limited resources and access, the project is based on a single transmitter and a limited set of locations, but as resources expand, the model can be scaled to accommodate multiple transmitters and receivers, supporting more comprehensive network planning and optimization. Additionally, if datasets of multiple floor plans are available, assuming all of them have the same powered transmitter, the model could generalize and predict free-space path loss and shadowing factors accurately for any given layout. This would allow the model to scale and predict the received signal strength (RSS) for any new floor plan, further enhancing its applicability in diverse indoor environments. Furthermore, if a time-varying dataset of the signal strength of the transmitter is available, machine learning algorithms could be used to predict the RSS at any particular time. This temporal prediction can be seamlessly incorporated into the current model, improving predictions for specific times and locations, and offering more accurate signal strength forecasting in dynamic environments.

8. References

1. H. Cheng, S. Ma, and H. Lee, “CNN-based mm Wave path loss modeling for fixed wireless access in suburban scenarios,” IEEE Antennas and Wireless Propagation Letters, vol. 19, no. 10, pp. 1694-1698, 2020.
2. H. Cheng, S. Ma, H. Lee, and M. Cho, “Millimeter wave path loss modeling for 5G communications using deep learning with dilated convolution and attention,” IEEE Access, vol. 9, pp. 62867-62879, 2021.
3. H. Cheng, H. Lee, and S. Ma, “CNN-based indoor path loss modeling with reconstruction of input images,” in Proceedings of 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, 2018, pp. 605-610
4. K. Kaemarungsi and P. Krishnamurthy, “Properties of indoor received signal strength for WLAN location fingerprinting,” in Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, MA, 2004, pp. 14-23.
5. S. S. Sidhu, A. Khosla, and A. Sharma, “Implementation of 3-D ray tracing propagation model for indoor wireless communication,” International Journal of Electronics Engineering, vol. 4, no. 1, pp. 43-47, 2012.
6. H. K. Rath, S. Timmadasari, B. Panigrahi, and A. Simha, “Realistic indoor path loss modeling for regular Wi-Fi operations in India,” in Proceedings of 2017 23rd National Conference on Communications (NCC), Chennai, India, 2017, pp. 1-6.
7. Enes Krijestorac, Samer Hanna, Danijela Cabric, Electrical and Computer Engineering Department, University of California, Los Angeles, USA
8. Y. Wang, W. J. Lu, and H. B. Zhu, “An empirical path-loss model for wireless channels in indoor short-range office environment,” International Journal of Antennas and Propagation, vol. 2012, article no. 636349, 2016. https:// doi.org/10.1155/2012/636349
9. D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2014; https://arxiv.org/abs/1412.6980.
10. W. S. Ahmed, “The impact of filter size and number of filters on classification accuracy in CNN,” in Proceedings of 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 2020, pp. 88-93.

Google Drive Link:
https://drive.google.com/drive/folders/1jsVXWjqbktm7LuMKbFPcBnuuS39RCqGn?usp=drive_link