5G mmWave Beam Selection Using ML DNN

Khalid Albagami
Ericsson AB
Riyadh, Saudi Arabia
K.albagami@hotmail.com

Raghad Alturki
Ericsson AB
Riyadh, Saudi Arabia
Raghad.Alturki@Ericsson.com

Bayan Aloquibi
Ericsson AB
Riyadh, Saudi Arabia
bayanaloquibi@gmail.com

I. Introduction

MM-waves (High-band) communication have proven to be quite challenging during the introduction of 5G-NR. In MM-waves communication, the signal tends to attenuate much faster compared to the current deployed low/mid -band networks as signal path loss (L) is directly proportional to the frequency squared f^2 according to the Free-space path loss (FSPL) equation given a zero dB gain for transmitter and receiver, i.e. $G_T G_R = 1$, as below:

$$FSPL = \left(\frac{4\pi d f}{c}\right)^2 \qquad (1)$$

where d is the distance and c is the speed of light. On the other hand, increasing the frequency f for a fixed antenna effective area A_e results in increased antenna gain G as follows:

$$G = \frac{4\pi f^2 A_e}{c^2} \qquad (2)$$

Thus, increasing the frequency has a potential advantage that, if exploited adequately, could theoretically negate the path Loss (L) drawback. This advantage, however, can be rather quite challenging because of the high gain narrow beam antenna design requirements. Choosing the right beam b_i from a large set of beams and direct it towards the receiver at each instant of time t_i would increase the signaling overhead and pre-processing prior to actual data transmission. In traditional NR-5G MM-wave system, beam sweeping mechanism is utilized to select the best beam which requires going through all the possible combinations between the transmitter T_x and receiver R_x beams. Such overhead and pre-processing in beam selection can be diminished by virtue of machine learning (ML).

In *ITU-PS-012 - ML5G-PHY Beam Selection Challenge*, the goal is to develop ML model that can select the best beam or recommend the top-10 beams list. The ML model output dataset represents the distribution of the average received power P_r between all possible beam pairs generated from the 32 T_x and 8 R_x (32 x 8 = 256 pairs of indices) using the beam simulator (Wireless Insite). By feeding the inputs and the simulated output, the ML model is designed to anticipate the distribution of P_r for the beam pairs and rank them accordingly. For the given T_x and R_x pairs, the accuracy of the existing ML classification model for top 10 beams in [1] is around 91%.

This paper is organized as follows: input datasets are explored in section II, the developed ML models are presented in section III, a comparison between the model results are performed in section IV, and finally the conclusion remarks are summarized in section V.

II. Dataset

In this section, we will discuss the different inputs fed to the ML model and what type of valuable data they provide. The data used as inputs to the model is collected through simulation in [2] using several systems which are Wireless Insite (Beam simulator), Blensor (LiDAR system), Blender (Camera system) and SUMO (Location system). The simulated environment in [2] consists of vehicles attached with receivers driving on a street with buildings on both sides and a base station (BS) located in the middle of the street transmitting several beams towards the receivers (as shown in Fig. 1). For more information about the data generation and collection refer to [2]. The collected data for all the input contains 11,194 samples.

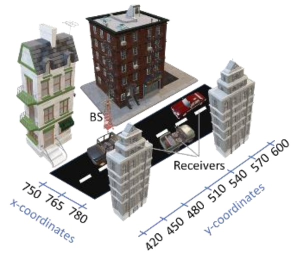

Fig. 1: Coordinates of the simulated environment in [2].

A. Receiver's Coordinates

The first input to the ML model is the user equipment (UE) receivers' coordinates. The data consists of two features which are the x- and y- coordinates. The data can be visualized in Fig. 2. The dots that you see in the figure is the different UE locations for 11,194 samples in the data. The data is stored in an array of dimension 11,194 x 2. The UEs locations with respect to the BS is plotted in Fig. 2.

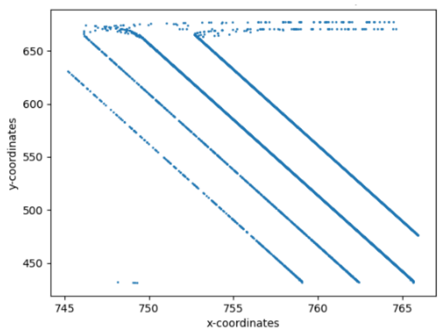

Fig. 2: Receiver's coordinated for 11,194 Samples.

B. Camera Images

Part of the data collected during simulation in [2] is footage taken from three cameras installed under the BS with variant angles. The images would supply the BS with insight regarding obstacles that might block the transmitted signal in line of sight (LOS) and non-LOS (NLOS) cases. The images used as an input to the ML model have been pre-processed and quantized to reduce the size of the data. The image resolution after pre-processing is 81 x 48 pixels. The images are stored in array of dimension 11,194 x 81 x 38. An example of the pre-processed images taken by the 3 cameras with variant angles are shown in Fig. 3.

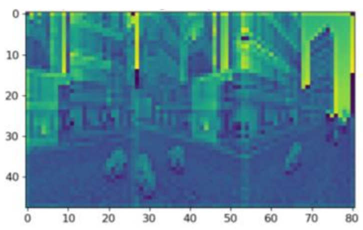

Fig. 3: Pre-processed image sample of 3 cameras with different angles.

C. LiDAR Cloud Inputs

The last input used to feed the ML model is based on collected data based on Light Detection And Ranging (LiDAR) system. LiDAR is a remote sensing technology that uses light in the form of a pulsed laser to detect and measure ranges (distances) to a target. The data collected from such systems could be rich and promising in providing a detailed analysis of the BS surrounding environment for a highly time-variant wireless channel in mm-wave communication. The LiDAR cloud point used as an input to the ML model have already been processed and quantized to reduce data size. The pre-processed LiDAR data clearly identify the location where we have obstacles (e.g. buildings, vehicles with no receivers, etc), base station (BS), receivers and free-space (as shown in Fig. 4). The LiDAR data is stored in 3-dimenssional array with dimension 20 x 200 x 10. The array dimensions represent a region with 10 height-levels along the street width (20-units) and length (200-units).

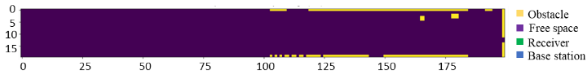

Fig. 4: Sample of LiDAR data for specific height (z=2)

III. Model Training

The training model in this section is built upon the python source code used in [2]. Prior to the model design, the available dataset from S08, which contains a total of 11,194 samples, is split into training (~85%), and validation (~15%) datasets.

Convolutional neural network (CNN) is considered for LiDAR and image input data as it achieves good results with less computational power for image input data. On the other hand, fully connected (FC) NN is utilized for simple input data with only two features such as coordinates.

A. Single-Input Model

AI models for beam selection are implemented for each dataset separately to investigate further the viability and potentials for the given three inputs, and to eventually derive the best model with highest accuracy. For each input, the architecture of neural network (NN) is optimized through a trial-and-error approach to improve the accuracy and overcome underfitting/overfitting challenges.

1) Reciever Coordinates

The coordinates array of dimension 11,194 x 2 is fed to the NN architecture for features extraction. Different number of NN layers are examined with different combination of activation nodes, see Fig. 5. Taking into consideration the model accuracy, three dense layers with (32, 256, 512) activation nodes is found to be the optimum as it achieves good accuracy 88% as shown in Fig. 6. A summary of the model is shown in Fig. 7.

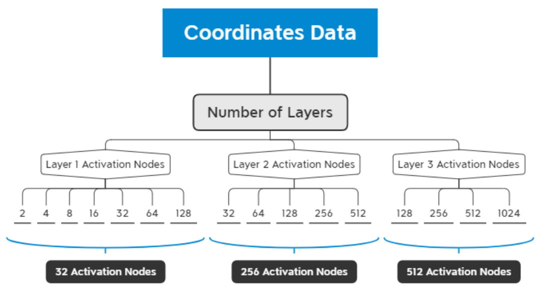

Fig. 5: Coordinates single-input NN model hyperparameter selection.

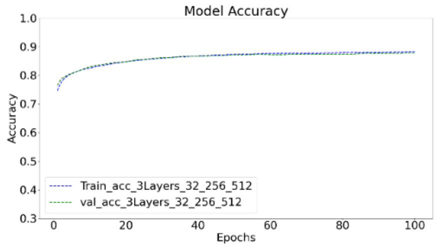

Fig. 6:Training and validation accuracy of best coordinates FC-NN model.

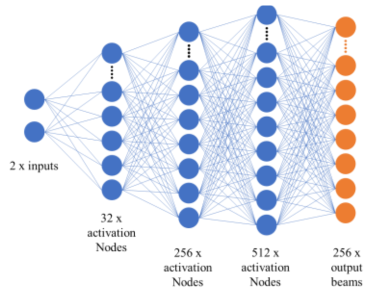

Fig. 7: Coordinates single-input NN model structure.

2) Camera Images

CNN model is an excellent choice for images dataset. Model architectures of up to three layers are trained, and it turns out that one CNN layer model gives an analogous result to the two/three layers models with 62% accuracy. The number of feature maps and FC layer nodes are further examined, Fig. 8 shows the model performance results. We can see that the images-based CNN model from Fig. 8 does not show promising results. A summary of the model is shown in Fig. 9.

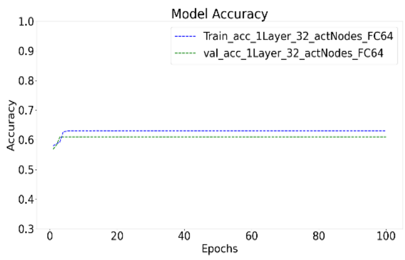

Fig. 8: Training and validation accuracy for best image CNN model.

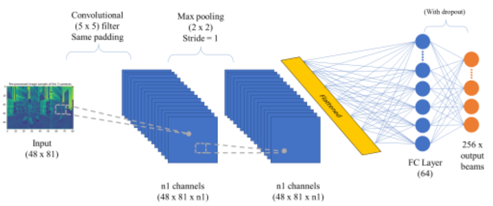

Fig. 9: Image single-input CNN model structure.

3) LiDAR Cloud Points

The 3-dimensional array from LiDAR is a more complex kind of inputs, and thus we opt to use the CNN architecture. We work on a series of experiments to find out how many layers, feature maps, and whether to use FC layers, in addition, regularization techniques were experimented to avoid underfitting or overfitting.

Initially, a comparison is done to different number of layers, i.e., 1, 2 and 3. The results showed that 2 layers gives a high accuracy level with less computational power. The chosen numbers of activation nodes, by experiment, are 8 and 16 for layer one and two, respectively. Next, 512 nodes FC layer is proven to give the best result in terms of accuracy. A range of a dropout (DO) and a regularization factors are investigated, wherein a factor of 0.2 and 0.008, respectively, provides a steady fitting model with 92% accuracy as shown in Fig. 10 and Fig. 11. A summary of the model and the chosen hyperparameters are shown in Fig. 12 and Fig. 13 respectively.

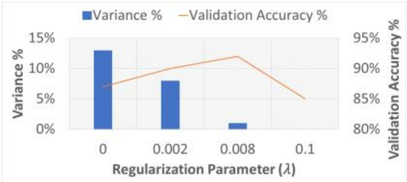

Fig. 10: Regularization impact on validation and training accuracy variance

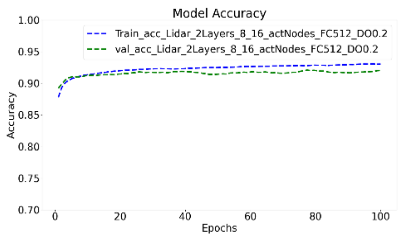

Fig. 11: Training and validation accuracy of best LiDAR CNN model

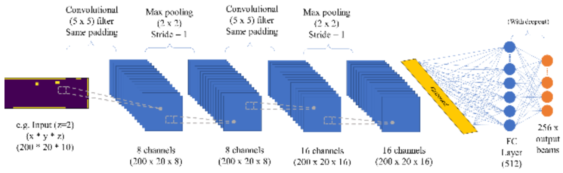

Fig. 12: LiDAR single-input CNN model structure

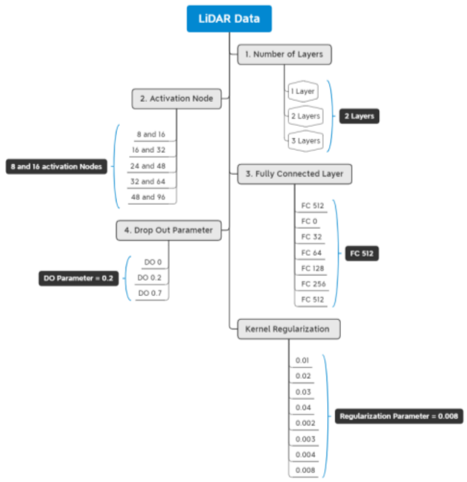

Fig. 13: LiDAR single-input CNN model hyperparameter selection.

B. *Multiple-Input Model*

So far in the previous section we only tested the model performance when fed with only one input. In this section we will see how the model would perform when it is fed to two or multiple inputs.

1) *Receiver coordinates and LiDAR Cloud Points*

After going through the results of the different data inputs used in the single-input model scenario, we see that LiDAR and coordinates data have delivered high top-10 classification accuracy up to 88% and 92% respectively. On the other hand, image input has not shown any good results. Therefore, a combined model with the two inputs can show promising results (i.e. LiDAR and coordinates data). The model structure is the combination of single-input coordinates' model shown in Fig. 7 and LiDAR's model in Fig. 12. Fully connected layer is added to combine the features extracted from both inputs and then fed to 256 output beams layer as shown in Fig. 14. The performance results shown in Fig. 15 after training the model for 100 epochs with 32 batch size achieves accuracy of around 92.3%.

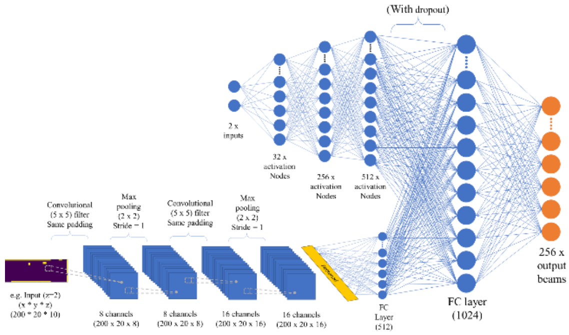

Fig. 14: Multi-input (Coordinates + LiDAR) NN model structure.

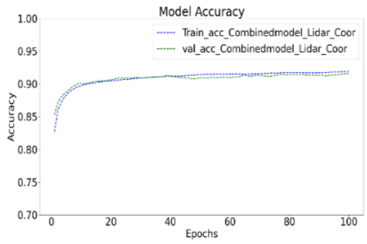

Fig. 15: Training and validation accuracy of double input model

IV. Results

The selection of the best model is based on validation accuracy level of k-top beams (where k=10) using S008 dataset with the aforementioned split. The three selected models are also tested using S009 dataset. Table 1. shows a comparison of accuracy levels of the following models:

Model A: 1 input, Positions of vehicles (Coordinates Data)
Model B: 1 input, LiDAR point clouds
Model C: 2 inputs, Coordinates data + LiDAR

Table 1. ML Models Accuracy Levels

	Model A	Model B	Model C
Training Accuracy (%)	88.2	93.0	93.5
Validation Accuracy (%)	87.9	92.1	92.3
Test Accuracy (%)	86.6	92.5	93.0
Computational Cost (hours)	0.03	0.19	0.27

Based on the results in the table the selected best ML model consists of two features as an input, that is, LiDAR and coordinates features. We can also see from the table that single-input LiDAR model have comparable results to the multi-input combined model. Finally, the best model (**Model C**) is to be used to produce the output result using the S010 dataset input.

V. Conclusion

In conclusion, we have seen that utilizing machine learning (ML) in the problem of beams selection can adequately generate accurate results up to 93% for the Top-10 beams classification case. These types of models could reduce the time and overhead needed for the base station to test the different available combination of beams and select the best beam to transmit on.

References

[1] A. Klautau, N. González-Prelcic, and R. W. Heath Jr., "LIDAR Data for Deep Learning-Based mmWave Beam-Selection," in IEEE Wireless Communications Letters, vol. 8, no. 3, pp. 909-912, June 2019

[2] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang and R. W. Heath, "5G MIMO Data for Machine Learning: Application to Beam-Selection using Deep Learning" in Information Theory and Applications Workshop (ITA), Feb. 2018. **DOI:** 10.1109/ITA.2018.8503086.