Deep Learning Enhanced RIS Configuration for Urban Scenario

Ramana Srivats[1], Shri Harish[1], Aravindan SM[1], P Kasthuri[1], P Prakash*

[1]Department of Electronics Engineering, Madras Institute of Technology Campus, Anna University
Chennai, India

*prakashp_mit@annauniv.edu

***Abstract*— This work introduces a significant advancement in Reconfigurable Intelligent Surfaces (RIS) to optimize wireless systems. Our RIS architecture integrates sparse channel sensors and a deep learning model inspired by AlexNet, addressing the challenge of channel estimation efficiently. With most elements passive and only a few actively connected to the baseband, our approach requires just 4000 data points to achieve performance close to the upper bound, outperforming existing methods. Utilizing the DeepMIMO dataset enhances data efficiency, minimizing training overhead while achieving high accuracy and efficiency gains. Specifically, our method attains the maximum achievable data rate with a significantly reduced dataset size. Traditional methods required over 30,000 samples for similar performance, but our model achieves comparable results with only 4000 data points. This represents a significant advancement in data efficiency, reducing the required dataset size by more than 86%.**

Keywords— Reconfigurable Intelligent Surface, AlexNet, Deep Learning

I. Introduction

In the fast-paced development of wireless communication, the incorporation of Reconfigurable Intelligent Surfaces (RISs) within urban environments holds the promise of transforming the standards of 6G systems [4]. These intelligent surfaces, characterized by arrays of sensing and radiating elements forming an electromagnetically active surface, bring a paradigm shift by intelligently interacting with incident signals. The primary objective is to amplify wireless system coverage and data speeds, addressing the critical requirements of the upcoming 6G era, which include fast data rates and low latency.

One feature contributing to the environmental sustainability of RISs in 6G technology is the utilization of passive elements [1]. This environmentally conscious approach underscores the significance of envisioning RISs as vital components in the 6G landscape. This work proposes an application of deep reinforcement learning for predicting the reflection coefficients of RIS elements. Only a subset of the total elements known as active sensors are used to estimate the channel information and predict the optimal beamforming vectors without the need for any prior training overhead. The proposed deep learning model utilizes alexnet architecture in which the optimal data rates are obtained at much lesser dataset sizes compared to the traditional approaches. Ahmed Alkhateeb [6] proposed a novel approach for Large Intelligent Surfaces (LISs) in beyond-5G wireless systems. Their method integrates passive and selectively activated active elements to mitigate training overhead, employing deep learning techniques for optimal interaction learning. Abdelrahman Taha et al. [3] addressed efficient configuration of Reflecting Intelligent Surfaces (RISs) using deep reinforcement learning (DRL). Their approach enables RISs to predict optimal reflection matrices autonomously, achieving high data rates with minimal training overhead. A. Alkhateeb [10] introduced MIMO dataset, a comprehensive resource for machine learning in wireless communication. The dataset, exemplified by the 'O1' ray-tracing scenario, provides parametrized data for diverse machine learning scenarios. Paper [4] reviewed machine learning (ML) advancements in intelligent reflecting surface (IRS)-enhanced communication. They explored ML techniques for channel estimation and optimization in IRS-enhanced wireless networks, identifying key challenges and research directions for future integration. Paper [5] gave an Ordinary Differential Equation-based Convolutional Neural Network for channel extrapolation in RIS-assisted communication systems. Their approach integrates cross-layer connections for improved convergence speed and system performance.

To summarize the above works Reflecting Intelligent Surfaces will play an important role in the development of 6G wireless technologies. There are many machine learning and deep learning techniques which enhance RIS based communication Channel estimation using conventional Least Square method will have a large overhead. From the above literature survey, we came up with a solution to efficiently predict RIS configuration with the use of only a few active sensors which sense the channel information and predict the optimal beamforming vector using this minimal channel estimation using deep learning.

The central contribution of this paper lies in introducing a deep learning-based RIS reflection design, Simulation results, utilizing ray-tracing channels, demonstrate convergence to near-optimal data rates with minimal beam training overhead and a limited number of active elements. Notably, employing an AlexNet model showcases the efficiency of our method, achieving optimal data rates with only 4000 data samples, compared to 30000 required by an ANN model.

II. DEEPMIMO DATASET

DeepMIMO is a generic dataset designed for millimeter wave (mmWave) and massive MIMO systems [6]. It serves as a common dataset for evaluating algorithms, reproducing results, setting benchmarks, anasdwasdd comparing different solutions in the field of machine learning for mmWave and massive MIMO [6]. The dataset is constructed based on accurate ray-tracing data obtained from Remcom Wireless InSite, capturing the surrounding geometry, materials, and transmitter/receiver locations.

A. Beamforming Codebook

A beamforming codebook is an essential element for designing antenna arrays to broadcast or receive signals in predetermined directions. The system is comprised of a fixed collection of beamforming vectors, with each vector representing a unique beam or direction that the antenna array can concentrate its signals towards. The main purpose of a beamforming codebook is to expedite the adjustment of the antenna array to varying communication conditions, enabling the array configuration to be dynamically modified according to the environment and user positions. The generation of a beamforming codebook entails various essential procedures.

TABLE 1. INPUT PARAMETER VALUES FOR CODEBOOK

INPUT PARAMETER	DESIGNATED VALUE
Mx(Number of elements in x direction)	1
My(Number of elements in y direction)	12
Mz(Number of elements in z direction)	12
over sampling factor for x	1
over sampling factor for y	1
over sampling factor for z	1
ant_spacing(Antenna spacing)	0.5 λ

The codebook generated for the sample input parameters mentioned in table 1 where each row encapsulates a distinct combination of beamforming vectors within the three-dimensional space of the Uniform Planar Array (UPA). These rows represent unique spatial directions or angles in the x, y, and z dimensions, forming complete beamforming vectors for the UPA and each column signifies a specific dimension. The over sampling factor interpolates or multiplies the sampling rate by the given number.

B. Ray-Tracing Scenarios in DeepMIMO Dataset

The DeepMIMO dataset relies on the precision of the Wireless InSite ray-tracing simulator by Remcom to generate scenarios that intricately capture dependencies on environmental geometry, materials, and transmitter/receiver locations. Channel parameters, including angles and path gains, are derived from the simulator outputs, offering researchers the flexibility to fine-tune dataset parameters for their applications.

C. The 'O1' Ray-Tracing Scenario

Illustrating this approach is the 'O1' ray-tracing scenario (shown in Fig. 1) within the DeepMIMO dataset—an outdoor environment featuring two streets and an intersection. The main street spans 600 meters by 40 meters, while a secondary street measures 440 meters by 40 meters. This scenario boasts 18 strategically placed base stations (BS1-BS18), with 12 on the main street (6 on each side) and 6 on the secondary street. The constant separation between specific base stations and a uniform height of 6 meters ensure a well-defined setup. The ray-tracing simulation for this scenario is executed using Remcom’s Site for enhanced accuracy. Fig.1 shows the model scenario of a urban area.

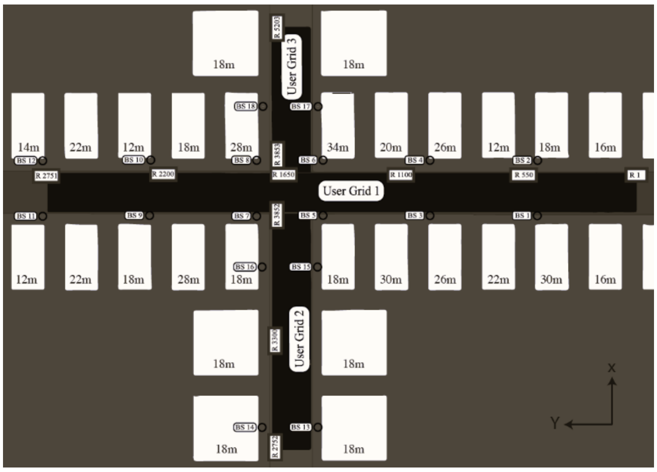

Fig.1. Top view of ‘O1’ ray-tracing scenario

D. Users in the 'O1' Ray-Tracing Scenario

Designed to include over one million users, the 'O1' ray-tracing scenario in the DeepMIMO dataset addresses the necessity for an extensive user population to provide a large dataset for deep learning applications. This intentional user density aims to support machine learning algorithms by offering a diverse and comprehensive training dataset.

E. DeepMIMO Dataset Generation Process

The generation process involves selecting a ray-tracing scenario and adjusting dataset parameters to fit the desired application. Utilizing channel parameters from the ray-tracing simulation as inputs, the dataset generation code builds the DeepMIMO dataset [3] based on both channel and unique system parameters of the chosen scenario.

The simulation setup involves several input parameters shown in table 2. The 'dataset_folder' parameter designates the directory path for loading the dataset. The 'scenario' parameter defines the scenario configuration, such as '01_28' indicating a scenario operating at 28 GHz. The 'dynamic_settings' dictionary includes settings related to dynamic aspects, specifying the starting and last scenes. Other parameters include 'num_paths' for the no. of paths from the transmitter to RIS element, 'active_BS' as a NumPy array indicating active base stations, and 'user_row_first'/'user_row_last' determining the range of rows for user locations. Additional details involve 'bs_antenna' and 'ue_antenna' dictionaries describing base station and user equipment antennas, respectively. 'enable BS2BS' is set to 1, enabling base station-to-base station communication. The 'OFDM_channels' parameter signifies the number of OFDM channels, set to 1. The 'OFDM' dictionary within includes subcarrier-related details.

TABLE 2. CHANNEL PARAMETERS

Parameters	Transmitter to RIS	Receiver to RIS
Scenario	28GHz (01_28)	28GHz (01_28)
Dynamic Setting	First Scene:1, Last Scene:1	First Scene:1, Last Scene:1
Number of Paths	1	1
First User Row	850	1000
Last User Row	850	1300
Row Subsampling	1	1
User Subsampling	1	1
Enable BS2BS (Base Station to Base Station)	True	True
OFDM channels	1	1
OFDM Subcarriers	512	512
OFDM Subcarriers limit	64	64
OFDM bandwidth	0.1 GHz	0.1 GHz

III. SYSTEM MODEL

In a communication system employing a Reflecting Intelligent Surface (RIS) with N passive elements, each equipped with PIN diodes for phase adjustment, we focus on single-antenna transmitters and receivers using OFDM with K subcarriers. RIS elements modify incident signals with phase shifts determined by their individual parameters. The received signal at the receiver, excluding direct links, accounts for RIS-induced modifications. A beamforming vector adjusts RIS reflections. Phase synchronization across subcarriers simplifies the model, though future work may explore per-subcarrier optimizations. Let $h_{T,K}, h_{R,K}^{T} \in \mathrm{C}^{\mathrm{Mx1}}$ denote the channels from the transmitter/receiver to the RIS at the kth subcarrier. Control of RIS elements is facilitated through a separate backhaul link, with disregarded signals undergoing multiple reflections. The received signal at the receiver can be expressed as:

$$y_k = h_{R,K}^{T} \psi h_{T,K} s_K + n_K \qquad (1)$$

$$= (\mathrm{h}_{\mathrm{R,K}} \odot \mathrm{h}_{\mathrm{T,K}})^{\mathrm{T}} \psi\, \mathrm{s}_{\mathrm{K}} + \mathrm{n}_{\mathrm{K}} \qquad (2)$$

The vector ψ represents the interaction effects in a Reconfigurable Intelligent Surface (RIS), where Ψ is a diagonal matrix with elements from ψ. Assuming an RIS design comprising radio frequency (RF) phase shifters, each interaction factor can be expressed as $[\psi]_m = e^{j\varphi m}$. Consequently, the selection of an interaction vector is limited to a predetermined set of options defined by a codebook P.

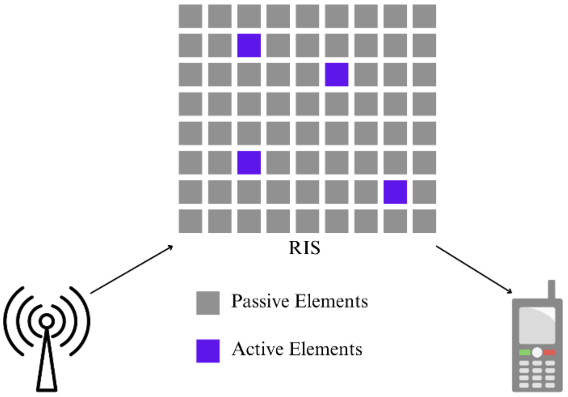

Fig. 2. Reconfigurable Intelligent Surface with Active Sensors

Fig. 2 depicts the proposed Reconfigurable Intelligent Surface (RIS) architecture, wherein M active channel sensors are distributed randomly across the RIS. These active elements operate in two modes: (i) a mode sensing the channel, where they are connected to the baseband to estimate the channels, and (ii) a reflection mode. The remaining RIS elements function as passive reflectors and are not connected to the baseband.

IV. METHODOLOGY

Reconfigurable Intelligent Surfaces (RISs) are garnering attention in the realm of future wireless systems, thanks to their potential for substantial coverage and data rate improvements. Comprising a multitude of nearly passive elements, these surfaces interact with incident signals, such as reflecting them, to enhance overall wireless system performance. However, the challenge of acquiring channel knowledge between the RIS and the transmitter/receiver arises due to the extensive number of elements and the use of nearly passive components.

A. *Proposed Algorithm*

The algorithm broadly consists of 2 stages, the training stage and testing stage. The Training Stage begins by initializing system parameters and iteratively processing each coherence block. Within each block, the channel vectors are estimated using received training pilots at the active sensors, which are then combined to form the channel vector hs. Subsequently, for each beam in the codebook, the algorithm obtains the feedback $\mathrm{R_n(s)}$ by reflecting and receiving the beam. A new data point is added to the training dataset D with the sampled channel information and corresponding feedback. Finally, the DL model is trained using the generated dataset D. In Algorithm 2, the Testing Stage employs the trained DL model to estimate the interaction vector continuously while receiving new pilots at the transmitter and receiver, estimating channel vectors, combining them into hs and using the trained DL model to predict the interaction vector. This iterative process continues until termination.

Algorithm for Training Stage

```
Input: number of active sensors, Beamforming Codebook, two pilots received at the transmitter and receiver
Output: A Trained DL Model
Step 1:  Initialize the Parameters for the system
Step 2:  for each coherence block do:
Step 3:  Estimate the channel vectors $\mathrm{h_t}$ and $\mathrm{h_r}$ using the received training pilots at the active sensors.
Step 4:  Combine the estimated vectors to form $\mathrm{h_s}$ :
Step 5:  $h_s = h_t \odot h_r$
Step 6:  for n = 1 to P do
Step 7:  Obtain $\mathrm{R_n(s)}$ by reflecting and receiving beam.
Step 8:  end for
Step 9:  Add a new data point to the training dataset(D)
Step 10: end for
Step 11: Train the DL model using generated dataset D
```

Algorithm for Testing Stage

```
Input: Trained DL Model, two pilots received at the transmitter and receiver
Output: Interaction Vector
Step 1:  while True do
Step 2:  Estimate the channel vectors $\mathrm{h_t}$ and $\mathrm{h_t}$ using the received training pilots at the active sensors.
Step 3:  Combine the estimated vectors to form $\mathrm{h_t}$ :
Step 4:  $\mathrm{h_t}. h_s = h_t \odot h_r$
Step 5:  Estimate the interaction vector using sampled channels and trained DL model
Step 6:  end while
```

B. Proposed Architecture for RIS

To address the challenge of channel estimation and training overhead, a novel RIS architecture is introduced. This architecture embraces a passive design for all RIS elements, except for a select few actively distributed channel sensors. These active sensors intricately connect to the baseband of the RIS, facilitating the good design of RIS reflection matrices. The channel information sensed by the sparse sensors will be used to train a DL model. This DL model can then be used to predict the beamforming vectors that would determine the output data rate at the receiver end.

E. AlexNet Based CNN Model Analysis

C. AlexNet Model

AlexNet is a convolutional neural network (CNN) architecture (shown in Fig. 2) that played a crucial role in advancing the field of computer vision and deep learning. The architecture of AlexNet demonstrated the effectiveness of deep learning in image classification tasks.

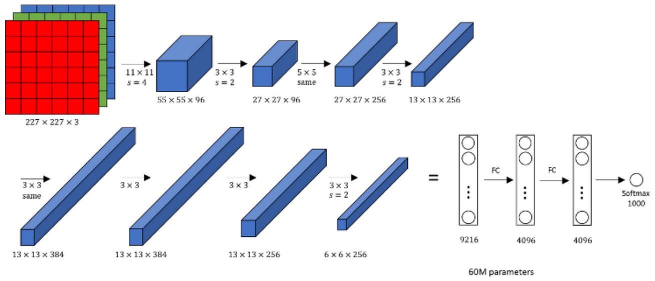

Fig. 3. AlexNet Architecture

D. Modified AlexNet architecture

The proposed neural network architecture is designed for regression tasks, specifically for a problem involving input data and corresponding target outputs. As show in Fig. 4 the architecture leverages transfer learning from the pre-trained AlexNet model and introduces modifications to suit the specific requirements of the research problem. The architecture commences with an input layer, conFig.d to accommodate the dimensions of the training data. Subsequently, layers from the AlexNet model are employed, starting from the transfer layer. Notably, modifications are introduced to the pooling and convolutional layers to enhance the network's ability to capture intricate patterns within the data. The pooling layers are adjusted to incorporate a max-pooling operation with a specified stride and padding, optimizing their contribution to the feature extraction process. Additionally, convolutional layers undergo modifications in terms of filter sizes, number of filters, and groupings, providing flexibility to adapt to the intricacies of the underlying data.

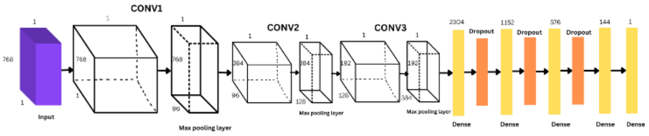

Fig. 4. Alexnet Build Design

Table 3 shows the different layers used in the Alexnet model. The neural network architecture starts with an imageInputLayer with dimensions [768, 1, 1], which acts as]the input layer for a particular image size. The convolutional layers, ReLU activation layers, and pooling layers from the pre-trained AlexNet are transferred to layers 2 to 10. The following layers undergo alterations, beginning with

TABLE 3. MODIFIED ALEXNET MODEL LAYERS

Name	Type	Learnable Properties

Input layer	Image Input	-
Conv1_mod	2-D Convolution	Weights 3×3×1×96 Bias 1×1×96
Relu1	ReLu	-
Norm1	Cross Channel Normalization	-
Pool1_mod	2-D Max Pooling	-
Conv2_mod	2-D Grouped Convolution	Weights 3×3×48×128×2 Bias 1×1×128×2
Relu2	ReLu	-
Norm2	Cross Channel Normalization	-
Pool2_mod	2-D Max Pooling	-
Conv3_mod	2-D Convolution	Weights 3×3×256×384 Bias 1×1×384
Fc6	Fully Connected	Weights 2304×748… Bias 2304×1
Relu6	ReLu	-
Drop6	Dropout	-
Fc7	Fully Connected	Weights 1152×2304… Bias 1152×1
Relu7	ReLu	-
Drop7	Dropout	-
Fc8	Fully Connected	Weights 576×1152 Bias 576×1
Relu8	ReLu	-
Drop8	Dropout	-
Fc9	Fully Connected	Weights 144×576 Bias 144×1
reg	Regression Output	-

pool1_mod, which is a maxPooling2dLayer with a pool size of 2x2, 2 strides, and padding of 1. Next, conv1_mod is a convolution2dLayer that utilizes a 3x3 filter size, 96 filters, a stride of 1, and a padding of 1. Conv2_mod is a groupedConvolution2dLayer with a filter size of 3x3, 128 filters, separated into 2 groups, and has the same stride and padding. Pool2_mod is an altered version of the maxPooling2dLayer, and conv3_mod is the subsequent layer, which is a convolution2dLayer with a filter size of 3x3 and 384 filters. The network architecture includes an extra layer called fc6, which is a fully connected layer with a number of neurons determined by multiplying the outSize by 16. This is followed by a reluLayer (relu6) and dropoutLayer a fullyConnectedLayer called fc7 is implemented with the number of neurons determined by multiplying the outSize by 8. This is then followed by relu7 and drop7 layers. The architectural sequence proceeds with fc8, a fullyConnectedLayer including a quantity of neurons determined by multiplying outSize by 4. This is thereafter followed by relu8 and drop8 layers. The fc9 layer is a fully connected layer that has a specific number of neurons determined by the outSize parameter. The architecture of the network is completed with a regression layer, also known as regLayer. Additional improvements to the architecture involve the addition of a dropout layer (drop9) as the 19th layer, which introduces regularisation with a dropout rate of 0.5. The 20th layer is a fullyConnectedLayer (fc10) consisting of 144 neurons, which is a customised configuration. The 21st layer is a regression Layer called reg Output. It calculates the mean squared error, which quantifies the discrepancy between the predicted and actual results. The mentioned additions enhance the complexity of the network by introducing regularisation to enhance generalisation and a specialised fully linked layer. The regression output layer, which utilizes mean squared error calculation, functions as the ultimate stage for evaluating the model's performance.

F. Alexnet Based Model Workflow

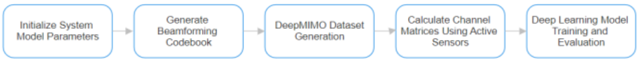
Initialize System Model Parameters
Generate Beamforming Codebook
DeepMIMO Dataset Generation
Calculate Channel Matrices Using Active Sensors
Deep Learning Model Training and Evaluation

Fig. 5. Flowchart Illustrating Code Execution

G. Alexnet Based Model Workflow

The process shown in Fig. 5 begins with the setup of system model parameters, establishing the foundational elements for subsequent operations. Following this, the beamforming codebook which contains all the possible RIS configurations for the specified RIS dimension is generated. Subsequently, the DeepMIMO dataset is generated for the specified system model parameters to simulate real-world scenarios, and channel matrices are calculated between the transmitter and RIS and the RIS and receiver utilizing only a subset of the total elements which are the active sensors. The next step involves training and evaluating a deep learning model using the collected data.

V. Computational Results

This work utilizes AlexNet, a deep neural network architecture, to optimize Reflecting Intelligent Surfaces (RIS) configurations in wireless systems. Compared to traditional Artificial Neural Network (ANN) models, AlexNet achieves optimal data rates with only 4000 data samples, whereas an ANN model requires 30000 data points for similar performance. This highlights the efficiency and effectiveness of AlexNet in enhancing wireless communication systems through deep reinforcement learning.

A. *ANN model output*

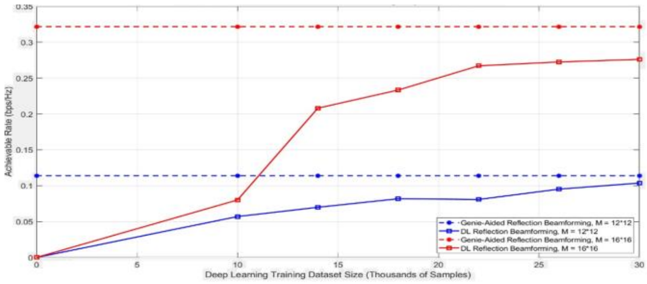

Fig.6. ANN Model Output Graph

Fig. 6 represents the achievable data rates for different data points using ANN model. This simulation is done for a RIS with 6 active elements and it is performed at a frequency of 28GHz. As the size of data set increases the ML model accuracy increases. Around 30000 data points

the predicted data rate gets close to the maximum achievable data rate. The above simulation is done for data set sizes of [10000, 14000, 18000, 22000, 26000, 30000]. We see that the upper bound line of the 16 ×16 RIS and the lower bound line of 12 ×12 RIS gets close to the maximum rate.

Fig. 7 shows the output plot of Alexnet model for data points [4000, 8000, 12000, 16000, 20000] for 28GHz scenario. We consider 6 active channel sensors for the simulation of both 144 element RIS and 256 element RIS. For the above data set size, we get a predicted data rate output of [0.1139, 0.1156, 0.1158, 0.1158, 0.1159] for 12×12 RIS and [0.3130, 0.3163, 0.3165, 0.3169, 0.3169] for 16×16 RIS. We run the simulation for data points of [1000, 2000, 3000, 4000] and see an increasing plot.

An increasing plot is observed and this depicts the better performance of machine learning prediction as more data samples is given. Around 4000 data points the model is able to predict optimum data rate and after 4000 points the graph becomes a straight line reaching the maximum achievable data rate. The observed phenomenon suggests that the AlexNet model reaches an optimal predictive performance around 4000 data points, after which further data doesn't significantly make any changes. The more dense and complex architecture of Alexnet model make optimal prediction with less training overhead and shows better learning ability compared to ANN model.

B. *Alexnet model output*

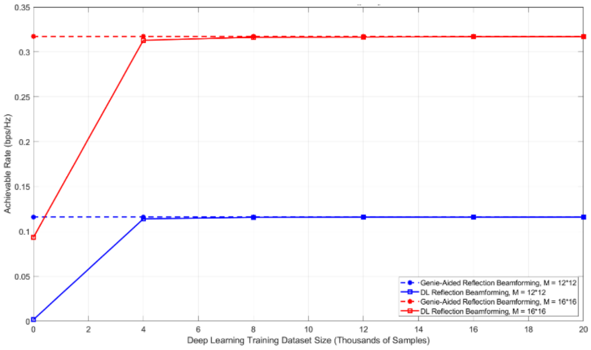

Fig. 7. Alexnet model output for dataset size 20000

An increasing plot is observed and this depicts the better performance of machine learning prediction as more data samples is given. Around 4000 data points the model is able to predict optimum data rate and after 4000 points the graph becomes a straight line reaching the maximum achievable data rate. The observed phenomenon suggests that the AlexNet model reaches an optimal predictive performance around 4000 data points, after which further data doesn't significantly make any changes. The more dense and complex architecture of Alexnet model make optimal prediction with less training overhead and shows better learning ability compared to ANN model.

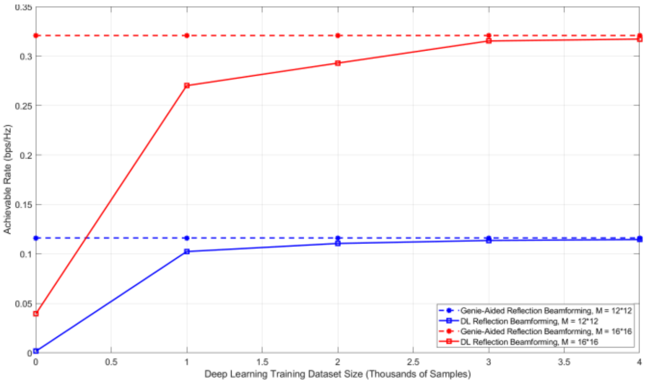

Fig. 8. Alexnet model output graph

Fig. 8 shows the plot for data points up to 4000. Here we can see the accuracy increases as the data set size gets closer to 4000 and then overlaps with the maximum achievable data rate. For the above data set size, we get a predicted data rate of [0.2701, 0.2928, 0.3152, 0.3171] for 16×16 RIS and [0.1024,0.1105,0.1136,0.1145] for 12×12 RIS.

Table 4. Comparison between ANN and Alexnet Model

DATASET SIZE	ANN MODEL ACHIEVABLE DATA RATE FOR 16 × 16 RIS	ALEXNET BASED MODEL ACHIEVABLE DATA RATE FOR 16 × 16 RIS	ANN MODEL ACHIEVABLE DATA RATE FOR 12 × 12 RIS	ALEXNET BASED MODEL ACHIEVABLE DATA RATE FOR 12 × 12 RIS
4000	0.06	0.3130	0.0344	0.1139
8000	0.113	0.3163	0.0563	0.1156
12000	0.145	0.3165	0.0612	0.1158
160000	0.176	0.3169	0.0743	0.1158
200000	0.225	0.3169	0.0992	0.1159

Table 4 shows the data rate obtained by Alexnet and ANN model for each dataset size. We see that the data rate achieved is higher for Alexnet model compared to ANN model. So, by using Alexnet model we are able to get better performance at small dataset size and also a higher data rate is achieved. In Fig. 6 we see the existing deep learning solution is able to reach almost 90% of the ideal rate when the model is trained on 14000 data points. In Fig. 7 we can see the proposed deep learning solution is able to achieve the same around 2000 data points. This significantly reduces the training overhead of the model.

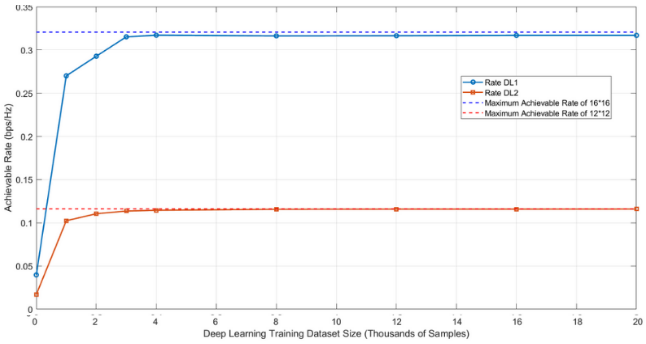

Fig. 9. Alexnet model output graph for 8 points

Fig. 9 shows a combination of both Fig. 7 and 8 and shows values for 8 data set sizes. The above simulation is performed [1000, 2000, 3000, 4000, 8000,12000,16000, 20000] data points. For all the simulations we consider number of channel paths as 1. Utilizing solely the deep learning model for the determination of beamforming vector helps in eliminating the need for beam training overhead. However, the effectiveness of this approach in achieving rates may be susceptible to minor alterations in the surrounding environment.

VI. Conclusion

This work is centered around developing an effective method for forecasting interaction matrices in RIS-assisted wireless communication systems. The goal is to create independent RIS architectures. By utilizing advanced deep learning frameworks, our proposed solution empowers the RIS to independently acquire and forecast the most effective RIS configurations by directly analyzing sampling channel knowledge. Significantly, this approach obviates the requirement for an initial phase of gathering a dataset, setting it apart from solutions based on supervised learning. The simulation results, with ray-tracing channels, show that our technique achieves convergence at data rates close to optimum, with low beam training overhead and a restricted number of active elements. The utilization of an AlexNet model demonstrates the efficiency of our method, attaining optimal data rates with only 4000 data samples, in contrast to an ANN model which necessitates 30000 data points to approximate the maximum data rate [1]. This highlights the effectiveness and high performance of our suggested deep reinforcement learning technique in optimizing RIS functionality for improved wireless communication systems. The traditional method involves the assumption of availability of the entire channel state information which would results in large training overheads and increased complexity, In the above paper we have proposed an RIS architecture wherein we use a small subset of the total sensors called as active sensors which would sense the channel resulting in reduced overhead and complexity.

A detailed model analysis and accuracy of the machine learning model developed in this project and the performance metrics analysis for the same shall be done. The maximum achievable data rate can be increased by increasing the bandwidth of the system since bandwidth is directly proportional to data rate. To do so the architecture of the channel should be modified and proper implementation of the above will result in higher data for the RIS wireless link.

References

[1] Baoling Sheen, Jin Yang, Xianglong Feng, and Md Moin Uddin Chowdhury, A Deep Learning Based Modelling of Reconfigurable Intelligent Surface Assisted Wireless Communications for Phase Shift Configuration. In IEEE Open Journal of the Communications Society, 2021

[2] Özgecan Özdogan, Emil Björnson, Deep Learning-based Phase Reconfiguration for Intelligent Reflecting Surfaces. In IEEE 54th Asilomar Conference on Signals, Systems, and Computers, 2020

[3] Abdelrahman Taha, Yu Zhang, Faris B. Mismar, Ahmed Alkhateeb, Deep Reinforcement Learning for Intelligent Reflecting Surfaces: Towards Standalone Operation. In IEEE International Workshop on Signal Processing Advances in Wireless Communications, 2020

[4] Mohammad Abrar, Md Habibur Rahman, Beom-Sik Shin, Ji-Hye Oh, Young-Hwan You, Hyoung-Kyu Song, Machine Learning for Intelligent-Reflecting-Surface-Based Wireless Communication towards 6G: A Review. In Sensors, 22(14), :2022

[5] Meng Xu, Shun Zhang, Caijun Zhong, Ordinary Differential Equation-Based CNN for Channel Extrapolation Over RIS-Assisted. In IEEE Communications Letters, 2021

[6] Ahmed Alkhateeb, DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications. In Proc. of Information Theory and Applications Workshop, 2019. https://doi.org/10.48550/arXiv.1902.06435

[7] S. V. Hum, J. Perruisseau-Carrier, Reconfigurable reflectarrays and array lenses for dynamic antenna beam control: A review, IEEE Trans. Antennas Propag., 2014.

[8] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, ‘‘Reconfigurable intelligent surfaces for energy efficiency in wireless communication,’’ IEEE Trans. Wireless Commun., Aug. 2019

[9] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, ‘‘A new wireless communication paradigm through software-controlled metasurfaces,’’ IEEE Commun. Mag., Sep. 2018.

[10] A. Alkhateeb, ‘‘DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications,’’ in Proc. Inf. Theory Appl. Workshop (ITA), San Diego, CA, USA, Feb. 2019

[11] A. Faisal, H. Sarieddeen, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, ‘‘Ultra-massive MIMO systems at terahertz bands: Prospects and challenges,’’ 2019, arXiv:1902.11090

[12] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, ‘‘Massive MIMO is a reality—What is next? Five promising research directions for antenna arrays,’’ arXiv:1902.07678v2 [eess.SP] 12 Jun 2019

[12] Gayathri V, Prakash P, Kasthuri P, Pavithra P "Investigation of WDM-MIMO Channel with DLF in Free Space Optical Communication," 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT) 2022.