

Antenna Array Pattern with Sidelobe Level Control using Deep Learning

Muhammad A. Abdullah¹, Alam Zaib¹, Shafqat U. Khan², Shoaib Azmat¹,
Shahid Khattak¹, Benjamin D. Braaten³, and Irfan Ullah¹

¹Department of Electrical and Computer Engineering
COMSATS University, Abbottabad, KPK 22060, Pakistan
ps23.asad@gmail.com, alamzaib@cuiatd.edu.pk, shoaibazmat@cuiatd.edu.pk, skhattak@cuiatd.edu.pk,
eengr@cuiatd.edu.pk

²Department of Electronics
University of Buner, Buner, KPK 19290, Pakistan
shafqatphy@yahoo.com

³Department of Electrical and Computer Engineering
North Dakota State University, Fargo, ND 58102, USA
benjamin.braaten@ndsu.edu

Abstract – Motivated by the demonstrated success of artificial intelligence (AI) in wireless communications systems, this paper proposes a deep learning-based approach for generating a desired radiation pattern with sidelobe level (SLL) control in active electronically scanned array (AESA) antennas. Recent works in this direction are mostly limited to generating radiation patterns with only beam scanning capability, inhibiting their wide-scale applicability. In this work, we propose a unified deep neural network (DNN) model that enable simultaneous control over both beam scanning angles and SLLs across a range of operating scenarios. To accomplish this task, the DNN model efficiently predicts the phase and amplitude of each array element. To learn the DNN model's parameters, we construct a training dataset comprising amplitude values and phases as labeled outputs and corresponding 181-point radiation patterns as input features. The training and validation process of the proposed DNN model reveals high accuracy in terms of R2 score and mean square error (MSE). For prediction, the desired radiation pattern consisting of 181 points is fed to the trained DNN model to yield optimized weights of antenna elements. The numerical results on a 1×8 linear phase antenna array, using an assortment of beam scanning angles and SLLs, demonstrate the effectiveness of the proposed model. The numerical results presented in MATLAB and CST simulators are validated by measurements on a 1×8 microstrip prototype array.

Index Terms – AESA antennas, array pattern, deep learning, deep neural network, sidelobe level control.

I. INTRODUCTION

The overall radiation pattern of an antenna array is dictated by phases and amplitudes of individual array elements. The phases of the array elements are used for beam scanning, and sidelobe level (SLL) is controlled by amplitudes [1, 2]. Though traditional methods (such as numerical, windowing techniques and optimization) [3–6] of beam scanning and SLL control offer advantages such as simplicity and low complexity, but they are less adaptive in dynamic operating environments and under more sophisticated practical applications.

Recently, artificial intelligence (AI), machine learning (ML), and especially deep learning-based radiation pattern synthesis have emerged as strong alternatives that offer novel solutions [7–15]. Although various deep neural network (DNN) models have been proposed for beam scanning of linear antenna arrays, the problem of SLL reduction using deep learning has received limited focus in current research. This paper aims to fill this gap, where we introduce a novel unified approach using DNN for radiation pattern synthesis that simultaneously achieves beam scanning and SLL control. The proposed approach, therefore, generalizes the previously developed DNN-based techniques [14, 15], which are applicable for beam scanning only, enabling the antenna arrays to generate arbitrary radiation patterns with desired main lobe direction and SLLs.

In this work, we consider an eight-element linear antenna array to develop and train a DNN model to predict the phases and amplitudes of each array element for generating radiation patterns with desired scanning directions and SLL. The unified DNN model can

accurately predict the amplitudes and phases of all eight elements from a single input radiation pattern consisting of 181 points. The results are compared by plotting radiation patterns based on actual and learned antenna weights in MATLAB and CST simulators, which are also validated by actual measurements on a prototype array inside an in-house anechoic chamber. Our results show the efficacy of the proposed DNN approach for synthesizing an arbitrary array pattern.

The key advantages of the proposed unified DNN model are as follows.

- (i) The proposed design offers simultaneous SLL control and beam scanning, thereby supporting a greater verity of applications.
- (ii) Leveraging DNN's capabilities, the proposed design offers high accuracy and fast convergence. Also, it can learn complex radiation patterns and its performance improves with increasing data due to the scalability of DNN.
- (iii) Once trained, the DNN model can instantly evaluate the complex weights (both amplitudes and phases) of the antenna elements for any input radiation pattern.

II. DNN ARCHITECTURE DESIGN

The architecture of our proposed unified DNN model is shown in Fig. 1. The radiation pattern of the linear antenna array, consisting of 181 points from 0 to 180 degrees, is fed as input features to the DNN model,

resulting in 181 neurons in the input layer, each corresponding to one angular point in the radiation pattern. The model is designed by using five fully connected hidden layers, each with ascending neuron count: starting at 2048 in the first hidden layer, followed by 1536, 1024, and 800 nodes in the intermediate layers and culminating at 600 in the last layer. In order to reduce the size of the dataset, the amplitudes of the antenna elements are considered symmetrical, with the central element weights set to one. Therefore, it is only necessary to determine the amplitudes of the first three elements and the phases for the seven elements. Note that the phase of the first element is taken as a reference and set to zero. Consequently, the output layer of DNN consists of 10 neurons i.e., three for the amplitudes and seven for the phases. In non-symmetric scenarios with N antenna elements, the size of the output layer would be $2N - 1$, comprising of N amplitudes and $N - 1$ phases. An Adam optimizer is used to reduce the mean square error (MSE) between the actual and DNN estimated array weights.

III. DATASET GENERATION WITH REDUCED COMPUTATIONAL COMPLEXITY

For radiation pattern synthesis, our dataset contains equally spaced 181 points of the radiation patterns as features, which are used by the model to map their relationship with amplitudes and phases of each antenna element acting as output labels of the data. The amplitudes range from 0 to 1 and phases vary from 0° to 360° for each antenna element (0° for the first element).

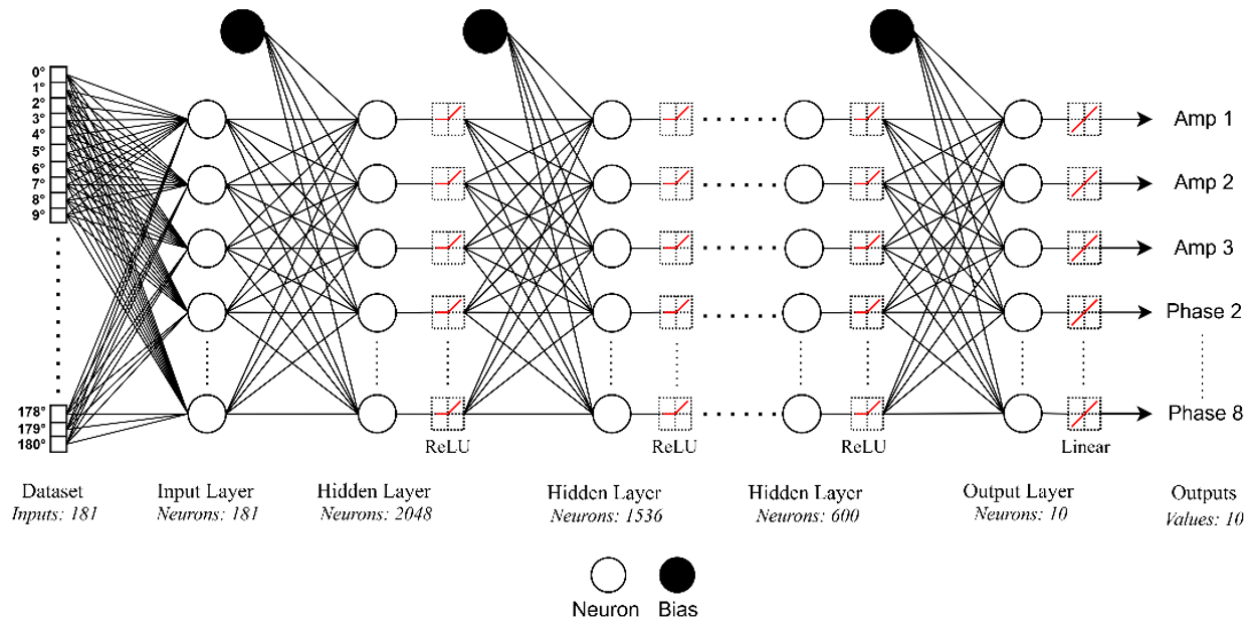


Fig. 1. Proposed DNN model architecture.

A. Amplitudes dataset

We use a structured approach that successively reduces the amplitudes dataset to a tractable size. This approach involves three steps.

1. Symmetry

To reduce the dataset size, we first exploit the symmetry of the radiation pattern about the main lobe which also implies symmetric amplitudes [16] across the array elements. It means if the model can predict the amplitudes of the first four elements in the eight-element array, the remaining four amplitudes can simply be mirrored. This significantly reduces the number of amplitude samples from 100 million to just 1000 (100,000-fold reduction) and with a reduced step size of 0.1 allowing us finer resolution in amplitude that improves model training accuracy.

2. Tapering

In this technique, the amplitudes of the array elements are generated by a suitable window function [3] to achieve the desired SLL. The effect of tapering on the radiation pattern would be that the main lobe remains high, the second sidelobes are smaller than the first, the third sidelobes are smaller than the second, and so on [17, 18]. The tapering step creates a highly relevant dataset and establishes a strong relationship between all the inputs and the desired radiation pattern.

3. Data cleaning

Tapering introduces redundancy in the dataset as it may result in many samples being repeated in the dataset. Thus, it is necessary to discard the recurring samples from the dataset by performing data cleaning. To efficiently generate the training dataset, tapering and data cleaning can be performed jointly in a single step. For the eight-element antenna array, again considering a step-size of 0.1, an additional up to 84% reduction in dataset size is possible by using both these steps.

The overall computational complexity for training the neural network can be expressed using big- O notation as $O(E \cdot D \cdot I \cdot H)$ as demonstrated in [19, 20], where E is the number of epochs, D is the size of training dataset, I is the size of the input layers and H is the number of hidden units. As the number of antenna elements N increases, a higher angular resolution is often required, leading to a proportional increase in the size of the input layer and hidden layers. For linear arrays, this results in a complexity that scales linearly with N . However, transitioning from a linear to a planar antenna array significantly increases the dimensionality of the radiation pattern, causing the complexity to scale quadratically with N [21].

B. Phases dataset

Since phases vary over a wide range 0° to 360° , the constant step size strategy is impractical here as it will generate a very large data set. To cope with this problem, we adopt a method proposed by the authors in [15] for reducing the dataset in array beamforming. The method in [15] uses uniformly spaced values of the phase difference (not the absolute phases) to generate the dataset. This significantly reduces the dataset size without comprising phase diversity. The dataset is generated using MATLAB software and is publicly accessible for download as referenced in [22].

It should be noted that a dataset of reduced complexity can also be generated for planar arrays with the same number of antenna elements by under sampling their radiation patterns in two dimensions. The proposed DNN model predicts the antenna weights with high accuracy. However, these results have not been included due to space constraints.

IV. TRAINING AND VALIDATION

In our experiments, we considered 80% of the dataset for training, 10% for validation, and 10% for testing. Initially, the number of epochs is set to 250. Then, to speed up the training process, we incorporated early stopping, which halts the training if the validation loss does not improve or begin to increase. The patience for early stopping was set to 10, meaning that if the validation loss does not improve over 10 consecutive epochs, training will stop. The patience for reducing the learning rate, a technique to reduce the learning rate when the validation loss plateaued, was set to five epochs. This means that if the loss did not improve after reducing the learning rate and met the early stopping criteria, the training

Table 1: Hyperparameters of the proposed DNN model

Hyperparameters	Values
Total data	2,011,100
Training data	80%
Validation data	10%
Testing data	10%
Input layer neurons	181
Number of hidden layers	5
Hidden layer neurons (per layer)	2048, 1536, 1024, 800, 600
Activation function in hidden layer	<i>ReLU</i>
Output layer neurons	10
Activation function in output layer	Linear
Optimizer	Adam
Number of epochs	250
Learning rate	0.001

would stop. This approach ensures that the training process is efficient and helps the model converge towards a lower error or loss. Compared to other methods where training continues for the full number of epochs even if the loss increases or remains unchanged, this method optimizes training time and prediction performance. The hyperparameters of the proposed DNN model are given in Table 1.

V. NUMERICAL RESULTS

This section presents a detailed numerical analysis of the proposed DNN model, including training, validation, and prediction results for radiation pattern synthesis of an eight-element linear antenna array. In all these results, R2 score and MSE serve as performance metrics. The numerical results are generated using Python but plotted in MATLAB, which are also verified using CST software. These results are validated through actual measurements on the prototype antenna array in section VI.

A. Training and validation results

The training was performed on a system with an NVIDIA L4 GPU, CUDA 12.2, and 24 GB of GPU

memory. Given the large dataset, we utilized Dask, a parallel computing library in Python, to efficiently handle and process large datasets. After the initial processing with Dask, we converted the data into a format compatible with TensorFlow for model training. Additionally, Joblib was employed to save the model weights and trained scaler to disk for later use. The training process was optimized by the ModelCheckpoint callback, which allowed us to save the best-performing model parameters, based on the lowest validation loss during training.

Figure 2 shows the results, in terms of MSE loss function for training and validation. Specifically, the MSE loss curve quickly converges to 0.000048391 indicating that the proposed method significantly reduces the model loss. Figure 2 also shows an average R2 score of nearly 1.0 across all epochs, indicating that our model faithfully captured the underlying patterns in the training data.

B. Prediction results

After completion of training and validation, the model's performance on entirely new data that is not part of the dataset needs to be evaluated. To this end, we generate optimal array patterns of the desired main

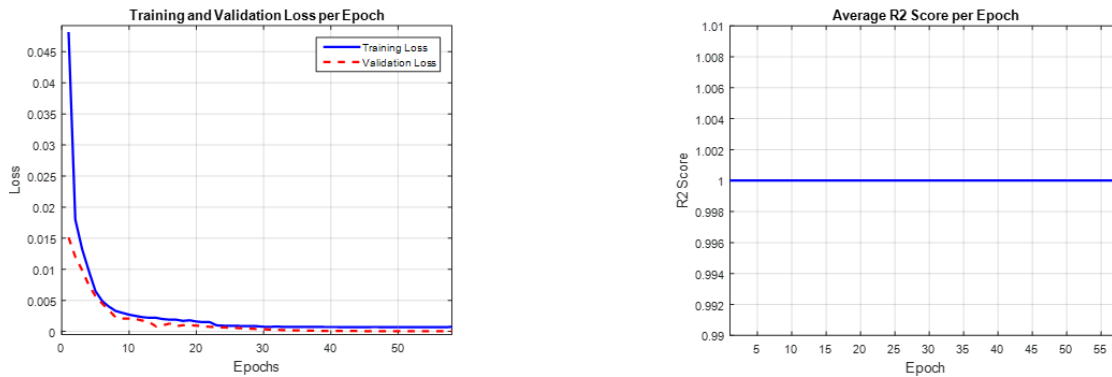


Fig. 2. Training and validation of the proposed DNN model: (a) loss curves and (b) R2 score.

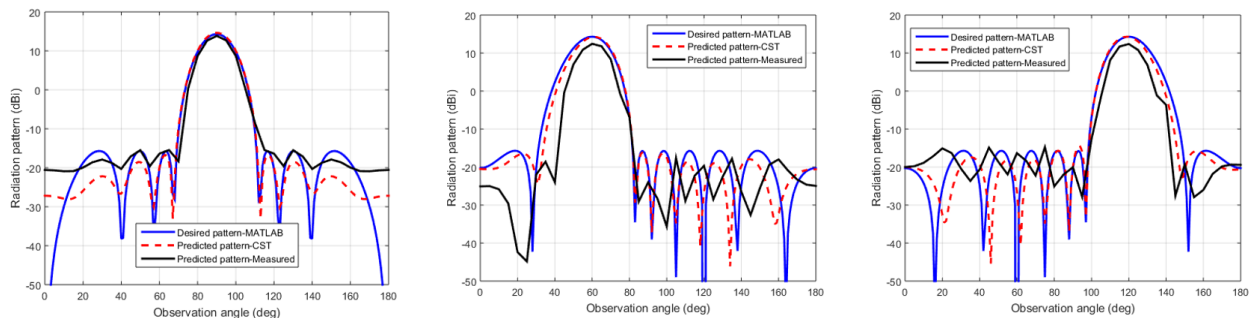


Fig. 3. Measured and predicted-simulated radiation pattern results of a 1×8 prototype array with SLL -30 dB and main beam at scan angles $\theta_s = 90^\circ$ (left), $\theta_s = 60^\circ$ (center), and $\theta_s = 120^\circ$ (right).

beam direction and/or side lobe level (SLL), which are then fed to the DNN model to predict the amplitudes and phases of each array element. To test the robustness of the model, we perform a critical analysis by testing the predictions in the following scenarios: SLL of -30 dB. with main beam scanning at 60° , 90° , and 120° observation angles. These three scenarios are illustrated in Fig. 3, which compare the desired and predicted radiation patterns. The results in Fig. 3 show the desired and predicted radiation patterns generated in MATLAB (blue) and CST (red) by using the complex weights learned by the proposed DNN model. The complex weights computed with the proposed DNN model are given in Appendix A.

VI. MEASUREMENT VALIDATION

To validate the predicted amplitude-phase composite DNN model, a 1×8 linear patch antenna array operating at 2.45 GHz and with an inter-element spacing of $\lambda/2$ was manufactured and is shown in Fig. 4 in the in-house anechoic chamber. The individual patch elements are excited through commercially available phase shifters (part no. HMC928LP5E) and voltage variable attenuators (part no. ZX73-2500-S+) to set the complex weights (see Appendix A) on the individual antenna elements in the array. These measured results are also shown in Fig. 3 for comparison with simulated results. The differences in peak gain and SLLs between simulated and measured

radiation patterns using the predicted array weights with the proposed DNN model are given in Table 2. The theoretical complex weights obtained with DNN model were quantized to the values achievable with available attenuators and phase shifters for the measurement validation. The quantized values obtained from measured datasheets of the attenuator and phase shifter reported in [23] are added in Appendix A. These quantized values were set through voltage-controlled attenuators and phase shifters and the resulting measured radiation patterns are shown in Fig. 3. As can be seen, there are differences in the measured and simulated radiation patterns due to quantized and theoretical weights feeding, however they are still within the acceptable range.

Based on the results in Fig. 3 and Table 2, several observations can be drawn: (a) the overall behavior of the predicted radiation patterns is similar to desired (ideal) patterns. This indicates the accuracy of the estimated composite amplitude-phase DNN model for phased array antennas; (b) there are acceptable small differences in phased array performance parameters of gain and SLLs, which shows that the proposed amplitude-phase DNN model can be used generically in SLL-controlled phased array applications, such as in massive MIMO and radar systems, to reject the interferes and clutters; and (c) the deviations in the measured and simulated patterns are mainly due to limitations of achieving the exact

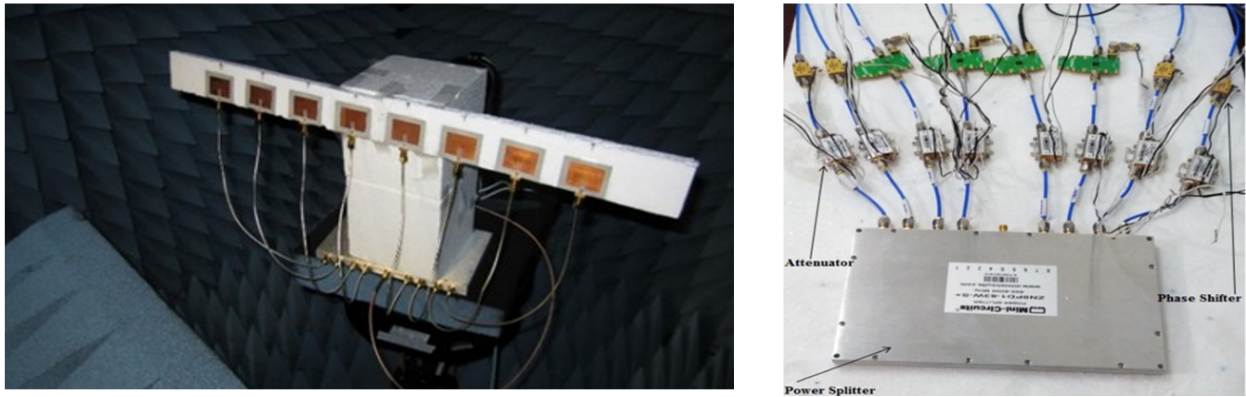


Fig. 4. A 1×8 prototype array (left) with integrated attenuators and phase shifters (right) in an in-house anechoic chamber measurements facility.

Table 2: Comparison between predicted, measured, and simulated results of a 1×8 prototype array with integrated attenuators and phase shifters

Scan Angle (θ_s)	Peak Gain (dB)			Sidelobe Level (dB)		
	Simulated (MATLAB)	Simulated (CST)	Measured	Simulated (MATLAB)	Simulated (CST)	Measured
90°	14.23	14.62	13.87	-29.78	-31.7	-29.32
60°	14.23	14.25	12.43	-29.83	-30.36	-31.03
120°	14.36	14.25	12.38	-30.06	-28.86	-27.18

amplitudes and phases with the microwave components (attenuators, phase shifters) and possibly imperfect measurement environment.

VII. CONCLUSION

In this paper, we proposed a DNN model that is capable of synthesizing the radiation pattern of an eight-element antenna array with precise control over SLL and main beam direction. We also proposed a structured way of generating the training data set that could speed up the learning of DNN model parameters. The model uses 181 points of the radiation pattern as input features set against the amplitudes and phases of array elements as labeled data. The prediction results carried out using MATLAB, CST, and measurements proved to be in close agreement with actual values showing the accuracy of the proposed DNN model with R2 scores approaching 1. In conclusion, deep learning models, especially DNNs, have proven their ability to learn and effectively be utilized in phased arrays. This work not only provides an innovative solution to current challenges in synthesizing radiation patterns but also paves the way for future developments. The success of DNN has opened the door to incorporating more advanced versions of deep learning, such as transformers and specialized transformers, to solve more complex real-world problems in phased array design and optimization.

ACKNOWLEDGMENT

This work is funded by the Higher Education Commission (HEC) Pakistan via Project No.20-17554/NRPU/R&D/HEC/2021.

REFERENCES

- [1] R. L. Haupt, *Antenna Arrays: A Computational Approach*. Hoboken, NJ: John Wiley and Sons, 2010.
- [2] R. C. Hansen, *Phased Array Antennas*. Hoboken, NJ: John Wiley and Sons, 2010.
- [3] Md. R. Sarker, Md. M. Islam, Md. T. Alam, and M. H.-E. Haider, "Side lobe level reduction in antenna array using weighting function," in *International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, Dhaka, Bangladesh, pp. 1-5, Apr. 2014.
- [4] S. Chatterjee, S. Chatterjee, and D. R. Poddar, "Side lobe level reduction of a linear array using Chebyshev polynomial and particle swarm optimization," in *International Conference on Communication, Circuits and Systems (IC3S-Number 1)*, June 2013.
- [5] S. Liang, Z. Fang, G. Sun, Y. Liu, G. Qu, and Y. Zhang, "Sidelobe reductions of antenna arrays via an improved chicken swarm optimization approach," *IEEE Access*, vol. 8, pp. 37664-37683, 2020.
- [6] K. Fu, X. Cai, B. Yuan, Y. Yang, and X. Yao, "An efficient surrogate assisted particle swarm optimization for antenna synthesis," *IEEE Trans. Antennas Propagat.*, vol. 70, pp. 4977-4984, 2022.
- [7] F. Andriulli, P.-Y. Chen, D. Erricolo, and J.-M. Jin, "Guest editorial: Machine learning in antenna design, modeling, and measurements," *IEEE Trans. Antennas Propagat.*, vol. 70, pp. 4948-4952, 2022.
- [8] W. T. Li, H. S. Tang, C. Cui, Y. Q. Hei, and X. W. Shi, "Efficient online data-driven enhanced-XGBoost method for antenna optimization," *IEEE Trans. Antennas Propagat.*, vol. 70, pp. 4953-4964, 2022.
- [9] R. Lovato and X. Gong, "Phased antenna array beamforming using convolutional neural networks," in *IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, Atlanta, GA, USA, pp. 1247-1248, July 2019.
- [10] T. Sallam and A. M. Attiya, "Convolutional neural network for 2D adaptive beamforming of phased array antennas with robustness to array imperfections," *International Journal of Microwave and Wireless Technologies*, vol. 13, pp. 1096-1102, 2021.
- [11] H. Bilel, L. Selma, and A. Taoufik, "Artificial neural network (ANN) approach for synthesis and optimization of (3D) three-dimensional periodic phased array antenna," in *17th International Symposium on Antenna Technology and Applied Electromagnetics (ANTEM)*, Montreal, QC, Canada, pp. 1-6, Aug. 2016.
- [12] Y.-S. Kim, D. Schvartzman, R. D. Palmer, and T.-Y. Yu, "Fast adaptive beamforming using deep learning for digital phased array radars," in *IEEE International Symposium on Phased Array Systems & Technology (PAST)*, Waltham, MA, USA, pp. 1-7, Oct. 2022.
- [13] L. Merad, F. T. Bendimerad, S. M. Meriah, and S. A. Djennas, "Neural networks for synthesis and optimization of antenna arrays," *Radioengineering*, vol. 16, pp. 23-30, 2007.
- [14] J. H. Kim and S. W. Choi, "A deep learning-based approach for radiation pattern synthesis of an array antenna," *IEEE Access*, vol. 8, pp. 226059-226063, 2020.
- [15] A. Zaib, A. R. Masood, M. A. Abdullah, S. Khatkhat, A. B. Saleem, and I. Ullah, "AESA antennas using machine learning with reduced dataset," *Radioengineering*, vol. 33, pp. 397-405, 2024.
- [16] I. Ullah, S. Khattak, and B. D. Braaten, "Improvement of the broadside radiation pattern of a

- conformal antenna array using amplitude tapering,” *Applied Computational Electromagnetics Society (ACES) Journal*, vol. 32, pp. 511-516, 2017.
- [17] N. S. Khasim, Y. M. Krishna, J. Thati, and M. V. Subbarao, “Analysis of different tapering techniques for efficient radiation pattern,” *e-Journal of Science & Technology (e-JST)*, vol. 8, pp. 47-55, 2013.
- [18] P. Delos, B. Broughton, and J. Craft, *Phased array antenna patterns—Part 3: Sidelobes and tapering*, ADI Inc, vol. 54, July 2020. [Online]. Available: <https://www.analog.com/en/resources/analog-dialogue/articles/phased-array-antenna-patterns-part3.html>
- [19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [20] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, “Large scale distributed deep networks,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, 2012.
- [21] T. T. Nguyen and K.-K. Nguyen, “A deep learning framework for beam selection and power control in massive MIMO-millimeter-wave communications,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4374-4387, 2022.
- [22] M. A. Abdullah, *Dataset for antenna array pattern with sidelobe level control using deep learning* [Online]. Available: <https://doi.org/10.5281/zenodo.14533762>
- [23] https://www.ndsu.edu/pubweb/~braaten/dissertation_Irfan.pdf

Appendix A: Desired vs predicted amplitudes and phases for 1×8 phased array antenna

	Desired Parameters	Desired Weights (element 1 to element 8)	DNN Predicted Weights
Amplitude-only scenario	SLL: -20 dB	0.5799 $\angle 0^\circ$, 0.6603 $\angle 0^\circ$, 0.8751 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 0.8751 $\angle 0^\circ$, 0.6603 $\angle 0^\circ$, 0.5799 $\angle 0^\circ$	0.5964 $\angle 0.00^\circ$, 0.6417 $\angle 0.00^\circ$, 0.8822 $\angle 0.00^\circ$, 1 $\angle 0.00^\circ$, 1 $\angle 0.00^\circ$, 0.8822 $\angle 0.00^\circ$, 0.6417 $\angle 0.00^\circ$, 0.5964 $\angle 0.00^\circ$
	SLL: -30 dB	0.2622 $\angle 0^\circ$, 0.5187 $\angle 0^\circ$, 0.8119 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 0.8119 $\angle 0^\circ$, 0.5187 $\angle 0^\circ$, 0.2622 $\angle 0^\circ$	0.2619 $\angle 0^\circ$, 0.5146 $\angle 0^\circ$, 0.815 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 0.8157 $\angle 0^\circ$, 0.5187 $\angle 0^\circ$, 0.2622 $\angle 0^\circ$
	SLL: -40 dB	0.1460 $\angle 0^\circ$, 0.4179 $\angle 0^\circ$, 0.7594 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 0.7594 $\angle 0^\circ$, 0.4179 $\angle 0^\circ$, 0.1460 $\angle 0^\circ$	0.1479 $\angle 0.00^\circ$, 0.4167 $\angle 0.00^\circ$, 0.7584 $\angle 0.00^\circ$, 1 $\angle 0.00^\circ$, 1 $\angle 0.00^\circ$, 0.7584 $\angle 0.00^\circ$, 0.4167 $\angle 0.00^\circ$, 0.1479 $\angle 0.00^\circ$
Phase-only scenario	$\theta_s = 60^\circ$	1 $\angle 0^\circ$, 1 $\angle 90^\circ$, 1 $\angle 180^\circ$, 1 $\angle 270^\circ$, 1 $\angle 360^\circ$, 1 $\angle 450^\circ$, 1 $\angle 540^\circ$, 1 $\angle 630^\circ$	0.9389 $\angle 0.00^\circ$, 0.9853 $\angle 90.00^\circ$, 0.9942 $\angle 179.99^\circ$, 1.00 $\angle 269.99^\circ$, 1.00 $\angle 359.99^\circ$, 0.9942 $\angle 449.99^\circ$, 0.9853 $\angle 539.99^\circ$, 0.9389 $\angle 629.98^\circ$
	$\theta_s = 90^\circ$	1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$	0.9581 $\angle 0.00^\circ$, 0.9849 $\angle 0.00^\circ$, 0.9849 $\angle 0.00^\circ$, 1.00 $\angle 0.00^\circ$, 1.00 $\angle 0.00^\circ$, 0.9581 $\angle 0.00^\circ$, 0.9849 $\angle 0.00^\circ$, 0.9849 $\angle 0.00^\circ$
	$\theta_s = 120^\circ$	1 $\angle 0^\circ$, 1 $\angle -90^\circ$, 1 $\angle -180^\circ$, 1 $\angle -270^\circ$, 1 $\angle -360^\circ$, 1 $\angle -450^\circ$, 1 $\angle -540^\circ$, 1 $\angle -630^\circ$	0.9568 $\angle 0.00^\circ$, 0.9809 $\angle 89.91^\circ$, 0.9836 $\angle 179.89^\circ$, 1.00 $\angle 269.89^\circ$, 1.00 $\angle 359.87^\circ$, 0.9836 $\angle 449.87^\circ$, 0.9809 $\angle 539.86^\circ$, 0.9568 $\angle 629.88^\circ$
Amplitude-Phase scenario	SLL: -30 dB $\theta_s = 60^\circ$	0.2622 $\angle 0^\circ$, 0.5187 $\angle 90^\circ$, 0.8119 $\angle 180^\circ$, 1 $\angle 270^\circ$, 1 $\angle 360^\circ$, 0.8119 $\angle 450^\circ$, 0.5187 $\angle 540^\circ$, 0.2622 $\angle 630^\circ$	Theoretical: 0.2476 $\angle 0.00^\circ$, 0.5201 $\angle 90.00^\circ$, 0.8060 $\angle 179.99^\circ$, 1.00 $\angle 269.99^\circ$, 1.00 $\angle 359.99^\circ$, 0.8060 $\angle 449.99^\circ$, 0.5201 $\angle 539.99^\circ$, 0.2476 $\angle 629.98^\circ$ Quantized: 0.25 $\angle 0^\circ$, 0.5 $\angle 87^\circ$, 0.65 $\angle 178^\circ$, 0.65 $\angle -92^\circ$, 0.65 $\angle 0^\circ$, 0.65 $\angle 87^\circ$, 0.5 $\angle 178^\circ$, 0.25 $\angle -92^\circ$
	SLL: -30 dB $\theta_s = 90^\circ$	0.2622 $\angle 0^\circ$, 0.5187 $\angle 0^\circ$, 0.8119 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 0.8119 $\angle 0^\circ$, 0.5187 $\angle 0^\circ$, 0.2622 $\angle 0^\circ$	Theoretical: 0.2619 $\angle 0^\circ$, 0.5146 $\angle 0^\circ$, 0.815 $\angle 0^\circ$, 1 $\angle 0^\circ$, 1 $\angle 0^\circ$, 0.8157 $\angle 0^\circ$, 0.5187 $\angle 0^\circ$, 0.2622 $\angle 0^\circ$ Quantized: 0.25 $\angle 0^\circ$, 0.5 $\angle 0^\circ$, 0.65 $\angle 0^\circ$, 0.65 $\angle 0^\circ$, 0.65 $\angle 0^\circ$, 0.65 $\angle 0^\circ$, 0.5 $\angle 0^\circ$, 0.25 $\angle 0^\circ$
	SLL: -30 dB $\theta_s = 120^\circ$	0.2622 $\angle 0^\circ$, 0.5187 $\angle -0^\circ$, 0.8120 $\angle -180^\circ$, 1 $\angle -270^\circ$, 1 $\angle -360^\circ$, 0.8120 $\angle -450^\circ$, 0.5187 $\angle -540^\circ$, 0.2622 $\angle -630^\circ$	Theoretical: 0.2611 $\angle 0.00^\circ$, 0.5360 $\angle -89.87^\circ$, 0.8144 $\angle -179.77^\circ$, 1.00 $\angle -269.69^\circ$, 1.00 $\angle -359.57^\circ$, 0.8144 $\angle -449.46^\circ$, 0.5360 $\angle -539.36^\circ$, 0.2611 $\angle -629.27^\circ$ Quantized: 0.25 $\angle 0^\circ$, 0.5 $\angle -86^\circ$, 0.65 $\angle -178^\circ$, 0.65 $\angle -90^\circ$, 0.65 $\angle 0^\circ$, 0.65 $\angle -86^\circ$, 0.5 $\angle -178^\circ$, 0.25 $\angle 87^\circ$

Note: Quantized values were obtained using the measured data sheets in [23]



Muhammad A. Abdullah holds a bachelor's degree in Computer Engineering from COMSATS University Islamabad, Pakistan. He is currently an AI Researcher and Engineer at ADK Technology Co., specializing in the development of machine learning (ML) and deep learning (DL) models for industrial applications, including neural architecture optimization and deployment of generative AI systems.



Alam Zaib received the Ph.D. degree in Electrical Engineering from KFUPM, Dhahran, Saudi Arabia, in 2016. He was an Erasmus Mundus scholar in MERIT master program from 2007 to 2009. Currently he is Associate Professor in the Department of Electrical Engineering at COMSATS University Islamabad, Abbottabad Campus. His research interests are in signal processing, wireless communications and applications of AI and machine learning in antenna arrays and wireless communication.



Shafqat U. Khan received M.S. and Ph.D. degrees in Electronic Engineering from International Islamic University Islamabad and ISRA University in 2008 and 2015, respectively. He was a Post Doc Fellow at the Faculty of Electrical Engineering, University Technology Malaysia, from 2016 to 2017. He is an Associate Professor at the University of Buner. His research interest includes the RF & microwave, antenna arrays and evolutionary algorithms.



Shoaib Azmat received his Ph.D. and M.S. degrees in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2014 and 2011, respectively. Currently, he is an Associate Professor of Computer Engineering at COMSATS University, Abbottabad. His research interests include computer vision, machine learning, and digital image processing, with research publications in reputed international journals.



Shahid Khattak received Dr.-Ing degree from Technische Universität Dresden, Germany, in 2008. He is a distinguished academic and researcher specializing in electrical and computer engineering with a focus on advanced communication systems, signal processing, and applied electromagnetics. He has contributed extensively to both theoretical and applied research, with numerous publications in reputed international journals. Khattak is known for his dedication to academic excellence and mentoring, and he plays a pivotal role in advancing interdisciplinary research in engineering and technology.



Benjamin D. Braaten (Senior Member, IEEE) received the B.S. degree in electrical engineering, the M.S. degree in electrical engineering, and the Ph.D. degree in electrical and computer engineering from North Dakota State University, Fargo, ND, USA, in 2002, 2005, and 2009, respectively. During the 2009 Fall semester, he held a postdoctoral research position with the South Dakota School of Mines and Technology, Rapid City, SD, USA. His research interests include printed antennas, conformal self-adapting antennas, microwave devices, topics in EMC, topics in BIO EM, and methods in computational electromagnetics. He is currently a Chairman of the ECE Department at NDSU, Fargo, ND, USA. He has authored or coauthored more than 100 reviewed journal and conference publications, several book chapters and holds one U.S. patent on wireless pacing of the human heart.



Irfan Ullah received a Ph.D. degree in Electrical and Computer Engineering from North Dakota State University, Fargo, ND, USA, in 2014. He is an Associate Professor in Electrical Engineering Department at COMSATS University Islamabad, Abbottabad Campus. His research interests include beamforming arrays, machine learning in antenna arrays, electromagnetic metamaterials, and topics in EMC.