AESA Antennas using Machine Learning with Reduced Dataset

Alam ZAIB, Abdur Rehman MASOOD, Muhammad Asad ABDULLAH, Shahid KHATTAK, Aasim Bin SALEEM, Irfan ULLAH *

Dept. of Electrical and Computer Engineering, COMSATS University Islamabad, Abbottabad Campus, P. O. Box 22060, Abbottabad, Khyber Pakhtunkhwa, Pakistan

{alamzaib, skhattak, eengr*}@cuiatd.edu.pk, {a.janvi1217, ps23.asad, rajaasim05}@gmail.com

Submitted March 26, 2024 / Accepted May 29, 2024 / Online first June 17, 2024

Abstract. *This paper proposes a deep neural network (DNN)-based approach for radiation pattern synthesis of 8 elements phased array antenna. For this purpose, 181 points of a desired radiation pattern are fed as input to the DNN and phases of array elements are extracted as the outputs. Existing DNN techniques for radiation pattern synthesis are not directly applicable to higher-order arrays as the dataset size grows exponentially with array dimensions. To overcome this bottleneck, we propose novel and efficient methods of generating datasets for DNN. Specifically, by leveraging the constant phase-shift characteristic of the phased array antenna, dataset size is reduced by several orders of magnitude and made independent of the array size. This has considerable advantages in terms of speed and complexity, especially in real-time applications as the DNN can immediately learn and synthesize the desired patterns. The performance of the proposed methods is validated by using an ideal square beam and an optimal array pattern as reference inputs to the DNN. The results generated in MATLAB as well as in CST, demonstrate the effectiveness of the proposed methods in synthesizing the desired radiation patterns.*

Keywords

Machine learning, neural networks, deep neural networks, active electronically scanned array, phased array, array pattern, Computer Simulation Technology

1. Introduction

Beam scanning is a signal processing technique commonly used in Active Electronically Scanned Array (AESA) antennas [1] to steer the main lobe of the radiation pattern towards a specific direction while reducing the energy radiated in unwanted directions. The practical utilization of AESA antenna systems spans both space and ground applications, starting from radars [2], to satellite communication [3], to data links [4] and 5G networks [5] to provide high-speed data connectivity. Beam scanning can be performed mechanically by adjusting the direction of the array to a specific direction, or it can be achieved electrically [1–4]. Electrical beam scanning involves adjusting the phase of the signals at each antenna element to achieve constructive interference in the desired direction and destructive interference in other directions. Beam scanning is conventionally done using numerical techniques which give the excitation phases to move the beam in the desired direction. However, the conventional methods for synthesizing radiation patterns in AESA antennas are often computationally intensive and time-consuming, making them challenging to implement in real-time critical applications [6]. In the domain of antenna arrays, machine learning has been successfully applied for antenna design and pattern synthesis, failure diagnosis of microstrip patch antenna array, and antenna selection as well as the direction of arrival estimation [7–16]. With the emergence of machine learning techniques, a paradigm shift has occurred in radiation pattern synthesis. Machine learning approaches offer significant advantages, as they enable the training and testing of models using multiple data points without the need for repetitive computations. Once trained, these models can be readily implemented on embedded systems or microcontrollers [17], providing a practical and efficient solution for radiation pattern synthesis.

The advancement in machine learning techniques has revolutionized the field, offering a more accessible and scalable approach to performance enhancement of antenna arrays. The neural network (NN)-based techniques such as convolutional neural networks (CNN) [12], [13] and deep neural networks (DNN) [18–20] have been developed for antenna array pattern synthesis to optimize the radiation patterns for specific purposes. In the realm of radiation pattern synthesis, previous research efforts have focused on 2D synthesis using CNN architecture. CNN, known for its expertise in image processing tasks due to its ability to automatically learn spatial features from data, has been used to process 3D geometry and antenna array features. By employing CNN, researchers have optimized antenna array element excitation and phase weighting to achieve desired radiation patterns, such as beamforming, null steering, sidelobe reduction, reflect arrays and 2D pattern synthesis.

More recently, researchers have found ways to improve the radiation patterns for a highly coupled antenna array with inter-element spacing being 0.28λ, where λ is the

DOI: 10.13164/re.2024.0397

wavelength. These NN-based methods are trained on various input radiation patterns to determine the complex antenna weights for individual array elements. In this line of work, the authors in [18] used DNN for a 4-element antenna array to synthesize a radiation pattern that follows an ideal pattern. The results showed that the DNN-based method was accurate and flexible as the patterns of the four-element antenna array were almost perfectly following the ideal pattern. Despite the performance gain, there are some major drawbacks in [18] as the approach used to generate the dataset for DNN is not scalable and requires a tremendous amount of training even for a small array of just 4 elements. The training dataset size grows exponentially with increasing the array size, making it impractical for higher-order antenna arrays. This motivated us to further extend this work to different array structures and array sizes without increasing the computational complexity of DNN by limiting the size of the training dataset. By extending the scope of investigation to larger arrays, we aim to uncover new insights and advancements in 1D radiation pattern synthesis, contributing to the existing body of knowledge in this field.

In this work, we introduce new approaches to generate datasets for training the DNN architecture for radiation pattern synthesis that overcomes the limitations of previous work [18] and extends it to higher-order linear phased arrays. In our work, we consider a phased-array antenna comprising 8 elements (though the approach can be extended to higher orders) and propose three different methods to encounter the problem of dataset size. The first method is based on restricting the range of phases to 180° thereby limiting the dataset to tractable size. Although this approach provides satisfactory results in terms of main beam scanning, it cannot synthesize the desired sidelobes. To overcome this, we devise another method that uses randomly generated phases with some correlation structure to form the dataset. The correlation is necessary as the DNN cannot learn the relationship between input and output from completely unstructured and independent data. It is shown that by randomly varying the phases over the full range from –180° to 180° better performance can be achieved with a smaller dataset size. The third method generates the dataset based on the constant phase difference between array elements instead of taking all combinations of phases of each element. This approach has clear advantages as the phases of individual elements are trivially known if the phase-shift is known and hence there is no sense of taking all the possible combinations of the phases of each element as is done in [18]. Hence, by leveraging the constant phase shift between array elements, this method can reduce the size of the dataset by several orders of magnitude. This is extremely crucial for the DNN, especially in real-time scenarios as the DNN can readily learn and synthesize the desired array patterns.

The main contributions of our work are summarized as follows:

- A novel DNN-based architecture is presented that could synthesize the radiation pattern for higher order linear phased array antenna.
- Three methods are proposed to generate datasets for training the DNN model with much smaller dimensions compared to existing methods making them feasible for higher-order arrays. This distinguishes our work from [18] which does not apply to higher-order arrays.
- Validation of the proposed DNN-based method is carried out using both MATLAB and electromagnetic simulator CST with ideal square wave and optimal array pattern. The radiation patterns synthesized using DNN followed the desired patterns in beam scanning of the main lobe as well as the sidelobes.

The remainder of the paper is organized as follows. Section 2 presents the mathematical model and the array structure of 8 elements linear phased array as implemented in CST, Section 3 describes the DNN framework for synthesizing antenna patterns, Section 4 presents proposed dataset generating strategies for DNN model, Section 5 describes validation setup for the proposed datasets, Section 6 presents numerical results and discussions, and finally, we conclude the paper in Sec. 7.

2. System Model

The radiation pattern of antenna arrays can be controlled and steered in different directions by adjusting the complex weights fed to the individual array elements. The complex weights of array elements form an array factor which is a function of the position of antennas in the array and the complex weights. In the case of AESA antennas with N elements, the array factor can be written as [21]:

$$AF = W_n \exp\left(\mathrm{j}\psi\right), \tag{1}$$

where

$$\psi = -2\pi\left(n-1\right)d\cos\left(\theta_0\right), \tag{2}$$

and

$$W_n = A\exp\left(\mathrm{j}\vartheta\right). \tag{3}$$

Here $n \in [1,N]$ is the element number, d is the inter-element spacing, ψ is the relative phase between the elements of the array, θ_0 is the scan angle of the main beam, A is the amplitude (which is set to 1 in case of the uniformly excited phased array) and ϑ is the input phase. The numerical approach for beam scanning can be derived from (1). Specifically, assuming $\psi = 0$ in the direction of main beam, equation (2) can be rewritten as:

$$\psi = -kd\cos\left(\theta_0\right) + \beta, \tag{4}$$

$$\Rightarrow\ \beta = kd\cos\left(\theta_0\right) \tag{5}$$

where $k = 2\pi/\lambda$ and β is the progressive phase shift between array elements. By utilizing the numerical method, a radiation pattern representing the desired scan angle can be obtained. For instance, if we want the main beam to steer at 120°, we would obtain the progressive phase shift of –90° from (5). It should be noted that the numerical method of beam scanning provides a benchmark for evaluating the performance of the DNN model to be presented later.

2.1 AESA Antennas Full-Wave Model

The radiation pattern of a phased array depends heavily on the geometry and arrangement of the individual antenna elements within the array. We consider an 8-element linear phased-array antenna placed in *xy*-plane and the radiation pattern is produced in the *z*-plane. The broadside of the array is observed at 90°. Figure 1(a) represents the CST model of 1 × 8 patch antenna array for the simulation of radiation patterns corresponding to the input signal of the antenna element. The substrate used is RT5880 with a dielectric constant of 2.2 and a thickness of 1.575 mm. The dimensions of an individual antenna element are shown in Fig. 1(b). The operating frequency of the AESA antenna is 10 GHz (X-band) and the inter-element spacing of 0.5λ is considered. The simulation is performed on CST, a commercial three-dimensional EM simulator.

3. DNN-Based Modeling

This section presents a comprehensive and detailed architecture of DNN designed for pattern synthesis in the antenna array. The DNN provides a powerful method for generating the phases for desired radiation patterns in array antennas. DNN uses various layers, optimizers, and an accuracy checker to generate accurate radiation patterns.

3.1 Architecture of DNN

The DNN is a type of neural network organized into multiple layers of interconnected nodes or neurons. Typically, it consists of an input layer, multiple fully connected hidden layers and an output layer. The deep architecture of DNN allows it to learn complex data patterns from input data.

The architecture of the proposed DNN model is shown in Fig. 2 which is designed to meet the specific requirements of the pattern synthesis task that allows an accurate and efficient radiation pattern generation. There are 181 neurons in the input layer corresponding to a specific angle point ranging from 0° to 180° in the radiation pattern synthesis. The input layer of the neural network receives individual data points associated with angles as inputs and allows the network to capture the complete pattern information.

The DNN uses a sequence of 8 interconnected hidden layers, boosting the network's ability to understand complex patterns and correlations within data. Each successive hidden layer represents a reduction of 20 neurons, promoting a gradual reduction in the dimensional representation of the network. Specifically, the first hidden layer has 160 neurons, the second layer has 140 and the last hidden layer has 20 neurons.

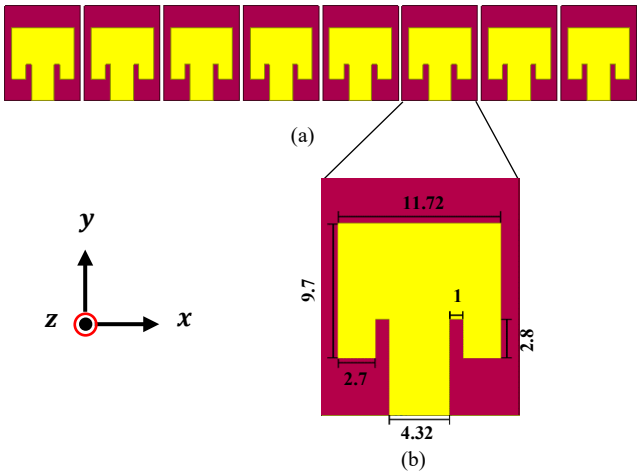

Fig. 1. (a) AESA antennas in CST full-wave simulator; (b) enlarged view of individual antenna elements. All dimensions are in mm.

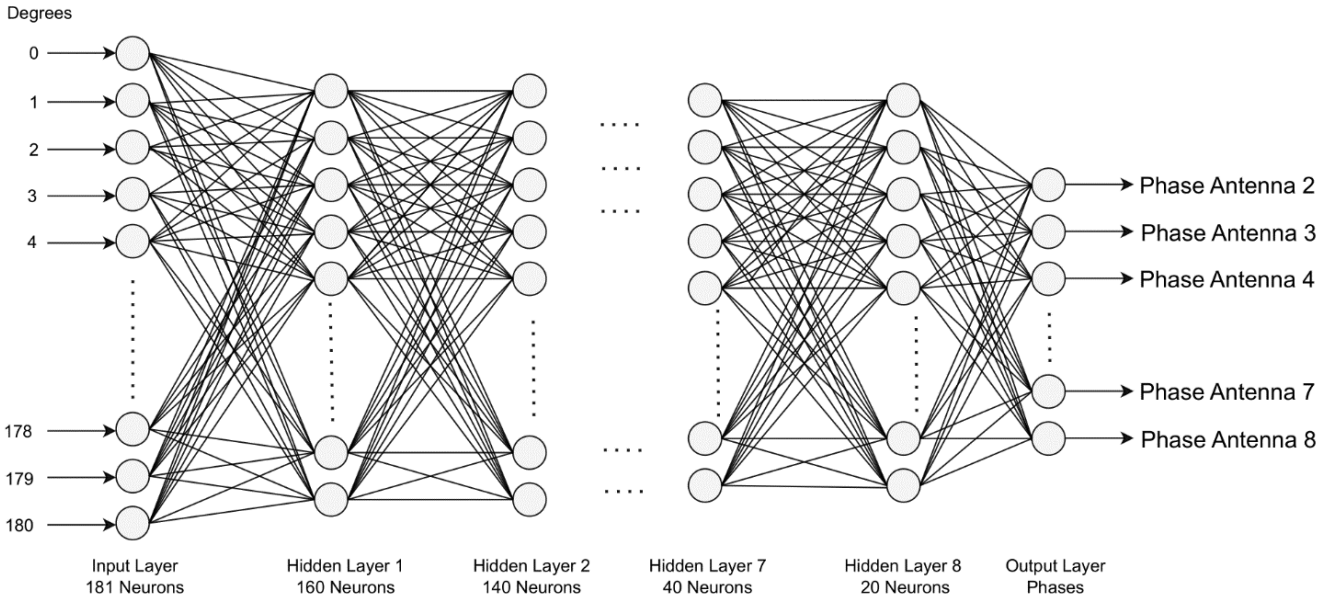

Fig. 2. Architecture of the DNN.

We use a rectified linear unit (ReLU) activation function in each layer of DNN architecture. This activation function is commonly employed in deep learning models for its ability to capture complex patterns effectively and introduce non-linearity into the network. The output layer of DNN corresponds to phases of the antenna array elements. For the 8-element phased array antenna considered in our work, the output layer shows 7 neurons corresponding to phases of array element 2 up to 8. The phase of element 1 is taken as a reference and is set to 0°, while the amplitudes are all set to unity. The mean squared error (MSE) is used to measure the average squared difference between the predicted and target values. It shows whether the network can accurately predict the desired radiation pattern. The Adam optimizer is also used to update the weights of the network during the training process.

4. Dataset Generation for DNN

The performance of DNN strongly depends on the dataset used for training. The dataset is generated by extracting the radiation patterns of antenna array which are acquired by changing the phases of excitations given to antenna elements. For this purpose, the radiation pattern is extracted on a linear scale and the phases are normalized to unity. The first element of the array is set to reference by fixing its phase to 0°. The phases of each of the other antenna elements were varied and a corresponding forward radiation pattern of the array is extracted that consists of 181 points of radiation pattern (0° to 180°). A similar approach is used in [18], where the radiation patterns were acquired by varying the phase of each antenna from 0° to 360° with a step size of 20°. Hence, the number of cases per antenna was 19, and for the considered 4-element array, the size of the dataset was $1 \times 19^3 = 6{,}859$. The validation dataset (which has to be different from the training dataset) was also generated similarly, but varying the phase from 10° to 130° with a step size of 40°. This resulted in 4 cases per antenna with a validation dataset size of $1 \times 4^3 = 64$. The datasets so generated have shown to achieve fairly accurate results for various desired patterns as the DNN was able to learn the radiation patterns accurately [18].

The major problem with the above approach is that it is not scalable to higher-order antenna arrays, and is also inefficient for AESA antennas which is the focus of this work. This is because the size of the dataset grows exponentially by adding more elements to the array. For instance, if we take an antenna array of 8 elements, the dataset size would grow to $1 \times 19^7 = 893{,}871{,}739$, which is prohibitively large making it intractable to use. The large dataset is difficult to generate or store and also results in increased complexity and processing time during training and testing of DNN.

Secondly, the previous approach is inefficient for AESA antennas, because a sizeable number of entries in the dataset would be redundant or meaningless, and as such, do not contribute to the learning process of DNN. The fact is that phased array antennas are characterized by constant phase shifts between array elements which is not exploited as yet. Next, we demonstrate that by leveraging the constant phase difference characteristic of a phased array antenna, a significant improvement in terms of dataset size reduction and the learning process can be achieved.

4.1 Proposed Methodology for Dataset Reduction

To overcome the drawbacks of the previous approach, we propose three different methods to generate the dataset [22] more efficiently. Each of these methods is discussed below in detail.

Method 1: This is the most trivial approach, where we use higher phase increments from $\Delta\theta = 20°$ to $\Delta\theta = 36°$, which gives 11 cases for each antenna and the dataset size of $1 \times 11^7 = 19{,}487{,}171$ for 8 elements array. Clearly, the dataset size is still too large for all practical purposes. To further reduce it, one can restrict the phases to 0° to 180° instead of the full range of 0° to 360°, which would yield the size of $1 \times 6^7 = 279{,}936$ entries. We can see that the dataset [22] size becomes quite tractable. However, we will see later that this method suffers from performance degradation due to phase restriction and large phase increments. Further, this method would not be scalable to higher-order arrays, as it is just a modification of the previous method [18] and is unable to completely solve the problem.

Method 2: In this method, we generate the phase values randomly, instead of varying with a constant phase increment of $\Delta\theta = 36°$ as suggested in Method 1. Specifically, we generate 200,000 uniformly distributed phase values between –180° to +180°, but normalized to ±1. However, it is observed that the DNN model cannot learn the relationship between input patterns and output phases due to the completely random and unstructured nature of phase values. To overcome this, we introduce correlation between the phases of array elements by multiplying the dataset of random phases with N-dimensional matrix $\mathbf{K}^{½}$, where the matrix $\mathbf{K}$ is referred to as the phase correlation matrix of the array antenna. Due to the constant phase shift between array elements in a phased array antenna, the phases of individual elements are highly correlated. Using this fact, the correlation matrix is constructed such that all diagonal entries are set to 1 and all off-diagonal entries are set to 0.9. After multiplication, a new dataset [22] with correlated phases is obtained and the corresponding radiation patterns are extracted and used for training the DNN model. It should be noted that a fairly large dataset should be generated to improve the performance, especially in higher-order arrays.

Method 3: This turns out to be the most efficient method as it takes advantage of constant phase shifts between antenna elements in a phased array antenna. Since the phases of individual array elements can be trivially obtained from the phase shift values, it makes no sense to vary them in the phased array antenna. Hence, in this method, the dataset is generated by varying only the phase shift rather than the phases of individual array elements as done previously.

Specifically, to generate the dataset, the phase of antenna 1 is fixed to 0° as before, but now the phase-shift, is varied from 0° to 360° with phase increments of $\Delta\theta$. This gives a total of $(360°/\Delta\theta + 1)$ entries for the phase shift as well as the dataset. Here, it is worth comparing the dataset size with the previous approaches, e.g. [18]. Assuming the phase increments of $\Delta\theta$ for each array element from 0° to 360°, the dataset size can be calculated as $(360°/\Delta\theta + 1)^{N-1}$ which grows exponentially with the array size N. Hence, the dataset size of the proposed method is independent of the array size N and is reduced by a factor of $(360°/\Delta\theta + 1)^{N-2}$ which is almost an order N reduction. As an example, if we use a constant phase increment of $\Delta\theta = 2°$, the proposed method yields a dataset size of 181 dimensions. As compared to the other two methods the dataset created is extremely small and it takes only a few seconds to train the DNN model. The phases of individual array elements can be simply found from the phase shift values by using:

$$\vartheta_n = n\Delta\theta, \quad n = 0,1,2\ldots N-1.$$

After the dataset [22] has been generated, it is fed to the DNN model which learns the relationship between the input radiation pattern and the output phases. Once the model is trained, a desired radiation pattern is fed as the input to the DNN and the output phases are extracted.

5. Verification Inputs to DNN

To validate the DNN model's performance, two different methods of validation of the neural network are used:

- Validation using ideal square wave;
- Validation using optimal beam scanning.

Note that, unlike the optimal beam scanning, the ideal square wave pattern is unrealistic and cannot be generated by any antenna array. The DNN model processes the given input patterns and generates phases based on the learned relationships between the two. It is important to note that the DNN model is specifically trained using linearly scaled patterns, so the inputs were provided in linear scale. To facilitate comparison and analysis, the output patterns are converted into the decibel (dB) scale.

5.1 Validation using Ideal Square Pattern

To validate the performance of the trained DNN model, a known ideal square wave pattern is generated as shown in Fig. 3. This pattern is specifically designed to have maximum amplitude within the range of ±10° around the desired scan angle (70° here). By feeding this ideal square wave pattern into the trained DNN model, the corresponding output phases are extracted.

The extracted phases are then independently verified using established electromagnetic simulation tools such as MATLAB and CST. These tools allowed for the generation of radiation patterns based on the extracted phases, which

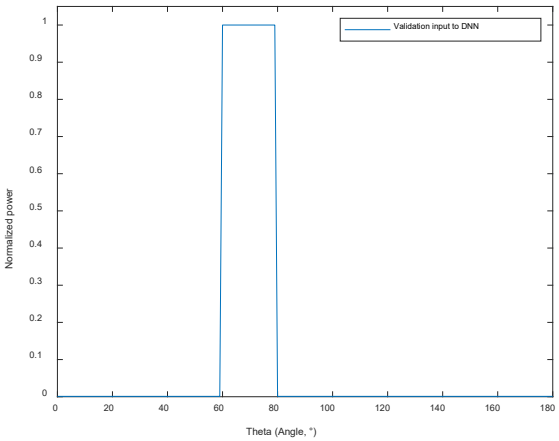

Fig. 3. Validation using square input.

are then compared to the ideal square wave pattern. The objective of this verification process is to ensure that the resulting radiation pattern exhibits the desired behavior, with the main beam confined within the boundaries of the ideal square wave.

5.2 Validation using Optimal Pattern

The optimal method of beam scanning employs a realistic radiation pattern and serves as a benchmark for evaluating the performance of the DNN model. This method determines the progressive phase shift β, required for achieving a specific desired scan angle, θ_0 by using (5). Figure 4 shows a radiation pattern generated by this method with the main beam directed at 70°. This pattern serves as an additional input to the DNN model for verification purposes.

6. Numerical Results and Discussion

We present numerical results for an 8-element phased array antenna to validate the performance of the proposed DNN method by using an ideal square wave pattern and an optimal pattern, discussed in the previous section, as reference inputs to DNN. The patterns synthesized by DNN will

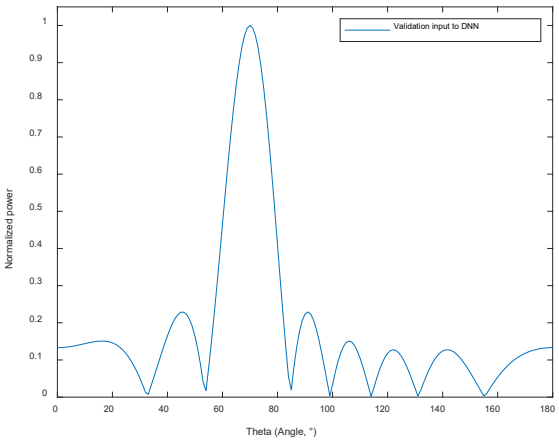

Fig. 4. Validation using optimal pattern.

be compared with the reference patterns to ascertain the efficacy of the proposed methods.

For training purposes, the datasets generated by using three proposed methods, are each split into 80% training and 20% reserved for testing. Each of the training methods is validated on two arbitrary scan angles, 70° and 120°, and outputs are verified using CST and MATLAB.

Figures 5–7 show training and validation loss curves for each of the three proposed methods. Method 1 exhibited a loss of 0.0471 and the testing loss was 0.0503. For Method 2, the training and testing losses were 0.0165 and 0.0187, and for Method 3 it was just 0.0006 and 0.0008, respectively. The convergence of curves in these figures indicates that the DNN model was able to learn the radiation patterns using datasets generated by each method.

6.1 Validation of Proposed Methods

After training the DNN model using the dataset generated by Method 1, an ideal square wave and an optimal array pattern are given as inputs to the DNN and output phases are extracted. These output phases are used to obtain the output radiation patterns in both MATLAB and CST simulator.

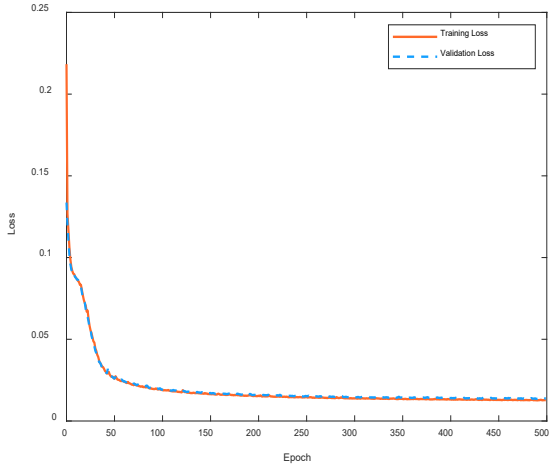

Fig. 5. Loss curve of the proposed Method 1.

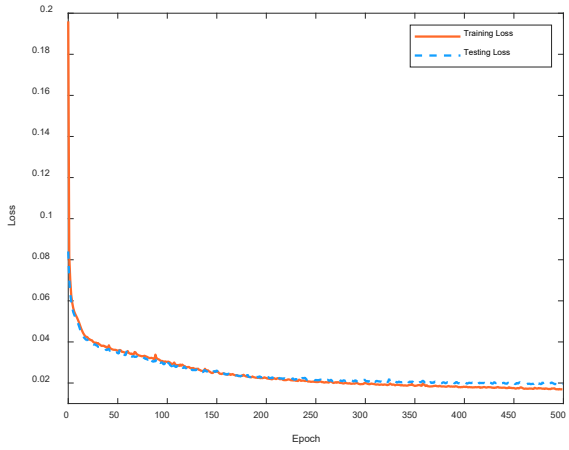

Fig. 6. Loss curve of the proposed Method 2.

Figures 8 and 9 present the overlaid results of all three methods using ideal square wave as input where the main beam was set to 120°. The solid blue line represents the ideal square pattern, which any antenna array cannot generate. The dotted red line shows the results of Method 1, and the dashed line and dash-dotted line show results for Method 2

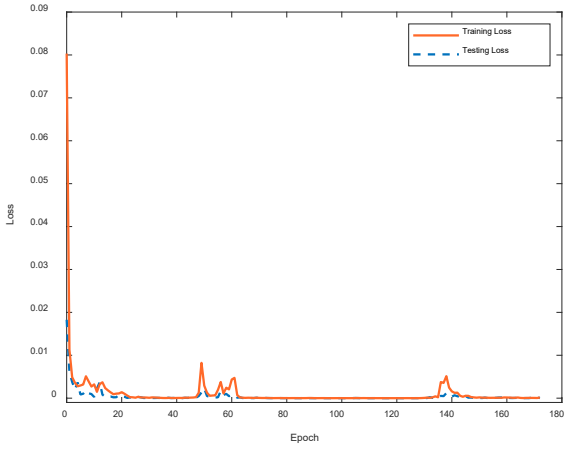

Fig. 7. Loss curve of the proposed Method 3.

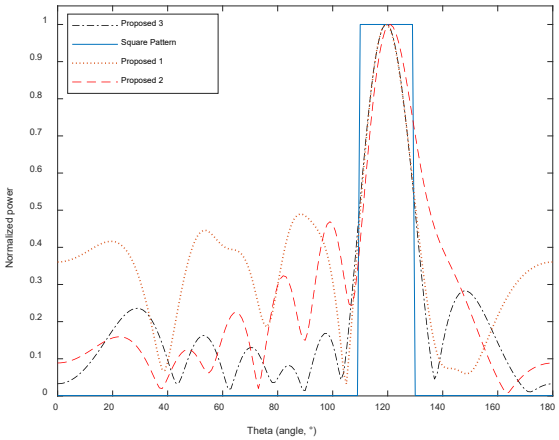

Fig. 8. MATLAB array pattern plots for square wave input at 120° to the DNN model.

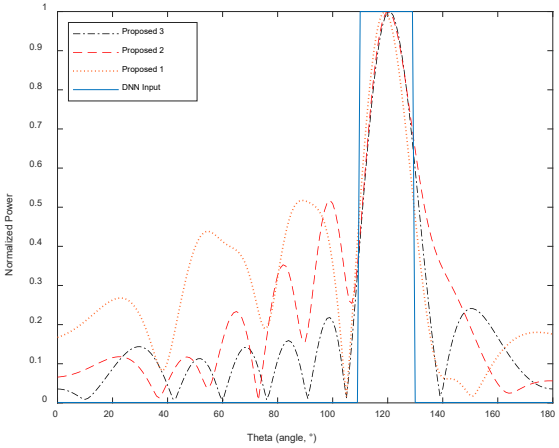

Fig. 9. CST array pattern plots for square wave input at 120° to the DNN model.

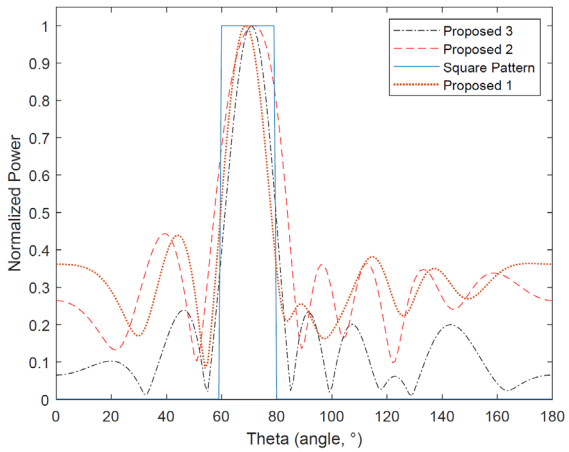

Fig. 10. MATLAB array pattern plots for square wave input at 70° to the DNN model.

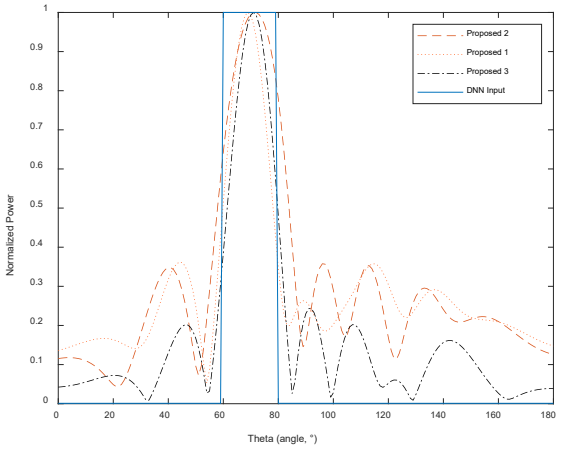

Fig. 11. CST array pattern plots for square wave input at 70° to the DNN model.

and Method 3, respectively. These results show that all the proposed methods can achieve the desired pattern with their main beams confined to square wave input.

Similar results are obtained when the main beam is shifted to 70° as shown in Figs. 10 and 11. As expected, Method 3 achieves the best performance as the sidelobes are lower than the other methods. Of these methods, Method 1, which is a slight modification of the existing method [18], has the worst performance and also uses the largest dataset size.

The optimal patterns are also given as the input to the DNN, and the output phases obtained by using each of the proposed methods are verified. The optimal radiation pattern was generated using (1)–(5) with scanning of the main beam at 120° and 70°. The results for all methods are shown overlaid in Figs. 12–15 using both MATLAB and CST.

As observed in Figs. 12–15, the patterns of the DNN obtained phases using Method 1 do not follow the optimal pattern, especially in the sidelobes where the deviation is more pronounced. It is because of the restricted phase range (0° to 180°) considered in the dataset for the training in Method 1. Nevertheless, the output radiation pattern does follow the main beam of the desired pattern.

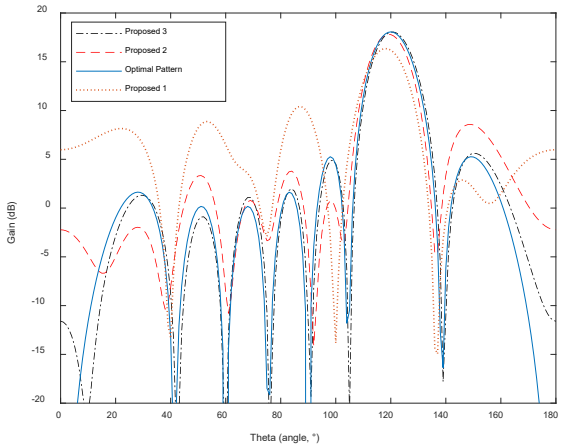

Fig. 12. MATLAB array pattern plots for optimal input at 120° to the DNN model.

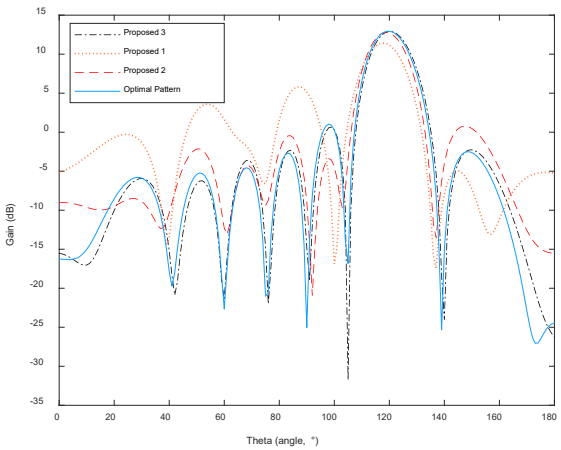

Fig. 13. CST array pattern plots for optimal input at 120° to the DNN model.

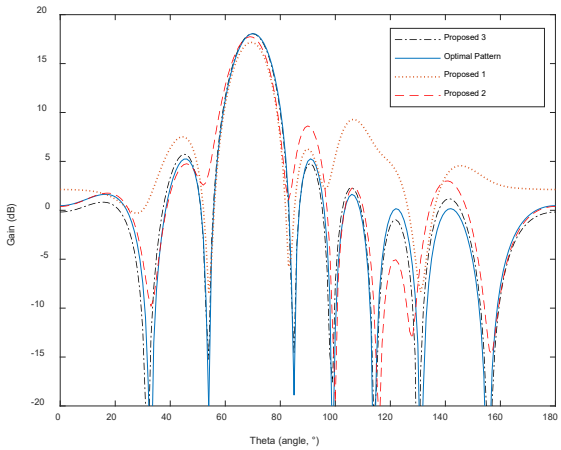

Fig. 14. MATLAB array pattern plots for optimal input at 70° to the DNN model.

As observed in these figures, Method 2 which uses randomly generated phases with strong correlation, produced remarkably good results as both the ideal patterns and the optimal patterns are successfully followed by the DNN. This is due to the inclusion of the full range of phases from 0° to 360° during dataset generation which was lacking in Method 1.

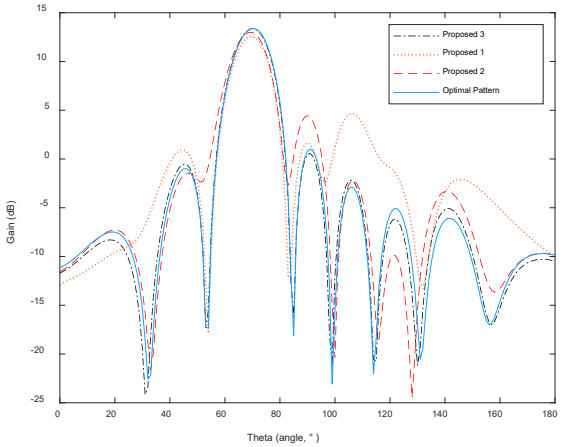

Fig. 15. CST array pattern plots for optimal input at 70° to the DNN model.

By allowing a wider spectrum of phase combinations used in Method 2, DNN can better capture the complex relationships and nuances of antenna array radiation patterns. The results presented in Figs. 12–15 confirm the ability of DNN to accurately synthesize realistic radiation patterns that are consistent with desired specifications. Hence, this method of randomly generated phases proves to be effective for achieving beam scanning.

Note that the dataset generated using Method 3 yielded significantly better results than the other two methods as evident from Figs. 12–15. Not only do the radiation patterns synthesized by DNN using this method follow the desired pattern almost perfectly, but that do with a significantly smaller dataset of 181 points. The reason is that the dataset generated with Method 3 is most consistent with the phased-array antenna structure considered in this paper. Unlike other methods, the dataset in this case exploited the constant-phase shift between array elements, resulting in phases of individual elements being equally apart. Hence, the DNN was able to quickly learn the inherent relationships between the excitation phases and the resulting radiation patterns.

The results are also summarized in Tab. 1, which shows the phases of the optimal array pattern and the learned phases using the DNN model along with the dataset size of each method. The phases of the 8-element array obtained using Method 3 are quite close to the original phases of the input patterns, which shows the superiority of Method 3 over other methods.

The results obtained highlight the potential advantages of practical implementation of DNN in real-world applications. Because the reduced or small dataset size and fast training time make it possible to deploy trained DNNs in time-sensitive applications or environments where computational resources are limited.

7. Conclusion

A deep neural machine learning network is designed for the pattern synthesis of 8-element AESA antennas for which the previous approaches are not applicable due to the prohibitively large datasets. Three methods for generating the dataset are proposed to address this problem, which results in tractable dataset size for an 8-element array. The first method, which is based on existing approaches proved to be useful only for main beam scanning and not the overall radiation pattern. The second method uses random but correlated phases to build a dataset of tractable size. When validated, the patterns created by the output phases were close to the reference patterns. Lastly, we devised another novel approach by exploiting the constant-phase characteristic of the phase-array antenna, where the dataset size was tremendously reduced. This proves to be simple and most efficient as it can be easily adapted to phased-array antennas of higher orders. We conclude that this is the best method for phase-array antennas as it outperforms other methods in terms of both the dataset size and pattern synthesis. The future work is the natural extension to amplitude-phase weights estimation to control more array parameters (side lobe level and nulls).

Input pattern	**Reference phases using equation (5)**	**Phases (DNN) using Method 1** (dataset size: 279,936)	**Phases (DNN) using Method 2** (dataset size: 200,000)	**Phases (DNN) using Method 3** (dataset size: 181)
Square wave at 120°	-	0, 187.2, 175.2, 66.8, 7, 174.8, 150.7, 104.7	0, 273.8, 151.5, 54.7, 300.1, 208.1, 142.4, 123.5	0, 244.3, 535.8, 810.2, 1032.8, 1312.5, 1560, 1847.1
Square wave at 70°	-	0, 80.5, 168.7, 170.6, –148.3, 7.2, –0.5, 134.2	0, 60, 132, 170, 238.9, 291.7, 357.9, 237.5	0, 58.9, 108.9, 192.3, 225.6, 297, 358, 413
Optimal pattern at 120°	0, –90, –180, –270, –360, –450, –540, –630	0, 153.9, 129.6, 60.2, –7.54, –158.5, 162.5, 86	0, –118.2, 152.5, 50, –5, –91.9, 163.5, 90	0, 270.1, 533.9, 803.7, 1071, 1344.5, 1610.5, 1878.4
Optimal pattern at 70°	0, 61.5, 123, 184.5, 246, 307.5, 369, 430.5	0, 38.6, 154.8, 174.5, –166.1, –20.1, 13.1, 85.5	0, 63.4, 128.9, –138.8, –78.6, –8.3, 23.7, 77.6	0, 62.3, 125.4, 191.4, 245.7, 310.7, 372.3, 435.2

Tab. 1. Output phases (degree) of the machine learning-based DNN model for 8-element AESA antennas.

Acknowledgments

This work is funded by the HEC-NRPU, Pakistan via Project No. 20-14696/NRPU/R&D/HEC/2021.

References

[1] MAILLOUX, R. J. *Phased Array Antenna Handbook.* 2nd ed. London (UK): Artech House, 2017. ISBN: 1-58053-689-1

[2] BROWN, A. D. *Active Electronically Scanned Arrays: Fundamentals and Applications*. 1st ed. Wiley-IEEE Press, 2022. DOI: 10.1002/9781119749097

[3] LISI, M. Specification, verification, and calibration of Active Electronically Scanned Array (AESA) antennas. In *4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*. Alkhobar (Saudi Arabia), 2021, p. 1–6. DOI: 10.1109/ISAECT53699.2021.9668352

[4] BROOKNER, E. Advances and breakthroughs in radars and phased-arrays. In *2016 CIE International Conference on Radar (RADAR)*. Guangzhou (China), 2016, p. 1–9. DOI: 10.1109/RADAR.2016.8059284

[5] SCHULPEN, R., JOHANNSEN, U., PIRES, S. C., et al. Design of a phased-array antenna for 5G base station applications in the 3.4-3.8 GHz band. In *12th European Conference on Antennas and Propagation (EuCAP).* London (UK), 2018, p. 1–5. DOI: 10.1049/cp.2018.1102

[6] BIANCO, S., NAPOLETANO, P., RAIMONDI, A., et al. AESA adaptive beamforming using deep learning. In *2020 IEEE Radar Conference (RadarConf20)*. Florence (Italy), 2020, p. 1–6. DOI: 10.1109/RadarConf2043947.2020.9266516

[7] MERAD, L., BENDIMERAD, F. T., MERIAH, S. M., et al. Neural networks for synthesis and optimization of antenna arrays. *Radioengineering*, 2007, vol. 16, no. 1, p. 23–30. ISSN: 1210-2512

[8] KIM, Y. Application of machine learning to antenna design and radar signal processing: A review. In *International Symposium on Antennas and Propagation (ISAP).* Busan (South Korea), 2018, p. 1–2. ISBN: 978-89-5708-304-8

[9] MISILMANI, H. M. E., NAOUS, T. Machine learning in antenna design: An overview on machine learning concept and algorithms. In *International Conference on High Performance Computing & Simulation (HPCS).* Dublin (Ireland), 2019, p. 600–607. DOI: 10.1109/HPCS48598.2019.9188224

[10] AKINSOLU, M. O., MISTRY, K. K., LIU, B., et al. Machine learning-assisted antenna design optimization: A review and the state-of-the-art. In *14th European Conference on Antennas and Propagation (EuCAP).* Copenhagen (Denmark), 2020, p. 1–5. DOI: 10.23919/EuCAP48036.2020.9135936

[11] SHAN, T., LI, M., XU, S., et al. Synthesis of reflectarray based on deep learning technique. In *Proceedings of the Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC)*. Xuzhou (China), 2018, p. 1–2. DOI: 10.1109/CSQRWC.2018.8454981

[12] LOVATO, R., GONG, X. Phased antenna array beamforming using convolutional neural networks. In *Proceedings of the IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*. Atlanta (GA, USA), 2019, p. 1247–1248. DOI: 10.1109/APUSNCURSINRSM.2019.8888573

[13] CHEN, Q., MA, H., LI, E.-P. Failure diagnosis of microstrip antenna array based on convolutional neural network. In *Proceedings of the IEEE Asia-Pacific Microwave Conference (APMC).* Singapore, 2019, p. 90–92. DOI: 10.1109/APMC46564.2019.9038656

[14] ERRICOLO, D., CHEN, P.-Y., ROZHKOVA, A., et al. Machine learning in electromagnetics: A review and some perspectives for future research. In *Proceedings of the International Conference on Electromagnetics in Advanced Applications (ICEAA)*. Granada (Spain), 2019, p. 1377–1380. DOI: 10.1109/ICEAA.2019.8879110

[15] JOUNG, J. Machine learning-based antenna selection in wireless communications. *IEEE Communications Letters*, 2016, vol. 20, no. 11, p. 2241–2244. DOI: 10.1109/LCOMM.2016.2594776

[16] ZHU, W., ZHANG, M. A deep learning architecture for broadband DOA estimation. In *IEEE 19th International Conference on Communication Technology (ICCT)*. Xi'an (China), 2019, p. 244–247. DOI: 10.1109/ICCT46805.2019.8947053

[17] BATZOLIS, E., VROCHIDOU, E., PAPAKOSTAS, G. A. Machine learning in embedded systems: Limitations, solutions and future challenges. In *IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*. Las Vegas (NV, USA), 2023, p. 345–350. DOI: 10.1109/CCWC57344.2023.10099348

[18] KIM, J. H., CHOI, S. W. A deep learning-based approach for radiation pattern synthesis of an array antenna. *IEEE Access*, 2020, vol. 8, p. 226059–226063. DOI: 10.1109/ACCESS.2020.3045464

[19] DI BARBA, P., JANUSZKIEWICZ, L. Linear antenna array modeling with deep neural networks. *International Journal of Applied Electromagnetics and Mechanics*, 2023 vol. 73, no. 4, p. 303–320. DOI: 10.3233/JAE-230086

[20] KASSIR, H. A., ZAHARIS, Z. D., LAZARIDIS, P. I. Antenna array beamforming based on deep learning neural network architectures. In *3rd URSI Atlantic and Asia Pacific Radio Science Meeting (AT-AP-RASC).* Gran Canaria (Spain), 2022, p. 1–4. DOI: 10.23919/AT-AP-RASC54737.2022.9814201

[21] BALANIS, C. A. *Antenna Theory: Analysis and Design.* 4th ed. NJ (USA): Wiley, 2016. ISBN: 978-1-118-64206-1

[22] Dataset. [Online] Cited 2024-03-26. Available at: https://www.cuiatd.edu.pk/electrical-computer-engineering/research-project-antenna-array-failure-correction-for-phased-array-radar/dataset/

About the Authors …

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