IEEE Open Journal of Vehicular Technology

Received 28 February 2024; accepted 15 April 2024. Date of publication 18 April 2024;
date of current version 10 May 2024. The review of this article was coordinated by Editor Koichi Adachi.

Digital Object Identifier 10.1109/OJVT.2024.3390833

A Novel Three-Dimensional Direction-of-Arrival Estimation Approach Using a Deep Convolutional Neural Network

CONSTANTINOS M. MYLONAKIS AND ZAHARIAS D. ZAHARIS (Senior Member, IEEE)

School of Electrical, Computer Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

CORRESPONDING AUTHOR: CONSTANTINOS M. MYLONAKIS (e-mail: mylonakis@ece.auth.gr)

This work was supported by the European Union through the Horizon 2020 Marie Skłodowska-Curie Research and Innovation Staff Exchange "Research Collaboration and Mobility for Beyond 5G Future Wireless Networks (RECOMBINE)" under Grant 872857.

ABSTRACT This article aims to constitute a noteworthy contribution to the domain of direction-of-arrival (DoA) estimation through the application of deep learning algorithms. We approach the DoA estimation challenge as a binary classification task, employing a novel grid in the output layer and a deep convolutional neural network (DCNN) as the classifier. The input of the DCNN is the correlation matrix of signals received by a 4×4 uniformly spaced patch antenna array. The proposed model's performance is evaluated based on its capacity to predict angles of arrival from any direction in a three-dimensional space, encompassing azimuth angles within the interval $[0^\circ, 360^\circ)$ and polar angles within $[0^\circ, 60^\circ]$. We aim to optimize the utilization of spatial information and create a robust, precise, and efficient DoA estimator. To address this, we conduct comprehensive testing in diverse scenarios, encompassing the simultaneous reception of multiple signals across a wide range of signal-to-noise ratio values. Both mean absolute error and root mean squared error are calculated to assess the performance of the DCNN. Rigorous comparison with conventional and state-of-the-art endeavors emphasizes the proposed model's efficacy.

INDEX TERMS Direction-of-arrival (DoA) estimation, convolutional neural network (CNN), deep learning (DL), binary classification, antenna array analysis and synthesis, spatial signal processing.

I. INTRODUCTION

Direction-of-arrival (DoA) estimation has been one of the most important problems in array signal processing for the past few decades [1], [2], [3]. In the pursuit of enhanced wireless communication, multiple-input multiple-output (MIMO) [4], [5], [6] antennas have proven to be a transformative technological breakthrough. MIMO systems utilize multiple antennas for both transmission and reception, exploiting spatial diversity to improve data rates and overall system performance. The development of "smart antennas" [7], [8], [9], [10], [11] has complemented MIMO technology, as these antennas can dynamically adjust their radiation patterns based on the estimated DoAs of incoming signals. By accurately estimating DoAs, smart antennas can optimize their main lobes towards the desired users and nullify interference from other directions, thus enhancing signal reception and reducing signal degradation caused by multipath propagation and fading. Researchers are actively developing advanced techniques to further bolster signal enhancement, e.g., electromagnetic skins, which are novel structures that can be integrated into antenna systems to manipulate radiation and scattering parameters [12], [13], [14], [15], [16], [17], [18]. Precise DoA estimation is crucial for smart antennas in MIMO systems, playing a pivotal role in achieving spatial signal processing, beamforming, and link reliability. As a consequence, applications reliant on precise DoA estimation, e.g., vehicular communications [19], [20], can experience significant improvements and thrive in performance through the advancement of research in this domain.

There are many conventional high resolution DoA estimation algorithms, such as multiple signal classification (MUSIC) [21], estimation of signal parameters via rotational invariance techniques (ESPRIT) [22] and Capon [23]. In ideal scenarios, these spatial spectrum estimation algorithms

demonstrate strong performance in acquiring the signal and noise subspaces through eigendecomposition of the signal's covariance matrix. However, these methods have certain limitations, particularly in non-uniform and low signal-to-noise ratio (SNR) environments. In such conditions, the accuracy and reliability of DoA estimation may be compromised. Another major disadvantage of conventional methods stems from their computationally intensive nature. These methods necessitate real-time complex calculations, placing a significant demand on computational resources and, consequently, requiring substantial hardware capabilities. The processing time for predicting DoAs is considerably prolonged, which can be impractical or prohibitive in certain time-sensitive applications. The complexity of these algorithms can lead to increased power consumption, limiting their applicability in resource-constrained environments. The slow prediction speed can adversely affect the overall system performance in dynamic environments, where DoAs of signals may rapidly change.

To overcome these challenges, various alternative approaches have been explored, with machine learning (ML) standing out as a particularly promising option. ML algorithms have the ability to learn and adapt from data, making them highly versatile and capable of handling complex and dynamic environments. Unlike conventional methods that rely on predefined mathematical models, they can discover patterns in the data, hence enabling more accurate and robust predictions. ML algorithms contribute to the advancement of research aimed at optimizing DoA estimation time by providing quicker and more efficient solutions. Neural networks (NNs) not only reduce computational complexity, but also enable swift predictions once the initial training phase is completed, thus eliminating the requirement for intricate real-time computations. Moreover, the ML approach enables the handle of a wide range of input data formats, allowing for seamless integration with different types of sensors and signal sources. Convolutional neural networks (CNNs) particularly demonstrate their effectiveness when challenged to learn and adapt to data for tasks such as object detection, image recognition, and semantic segmentation [24], [25], [26]. It is reasonable to expect that CNNs can also excel in DoA estimation when appropriately applied for this purpose.

By leveraging deep learning (DL) techniques and fine-tuning CNNs, this research aims to enhance the current state of DoA estimators and bridge the existing gap in the literature concerning three-dimensional (3D) space research. We are pursuing binary classification to achieve DoA estimation. To accomplish this, we employ a grid in the fully connected (FC) output layer (classification layer) of the proposed DCNN, which consists of $2N$ subgrids, where N is the total number of incoming signals, meaning that each subgrid corresponds to a specific angle of arrival (AoA) of each signal. This approach enables the network to discriminate between different angles. To address the limited research on planar arrays and 3D space, we harness the power of DL techniques to achieve DoA estimation for a patch antenna array consisting of 4×4 uniformly spaced elements. The key contributions of this paper are listed below:

1) We introduce a novel approach in the output layer, employing a grid for binary classification. The proposed grid systematically divides the 3D spatial domain, facilitating structured classification of angles. This approach demonstrates superiority in DoA estimation accuracy.
2) We conduct hyperparameter tuning for hidden layers, strategically minimizing the model's complexity. We fine-tune and optimize the proposed architecture of the hidden layers specifically for three incoming signals. Subsequently, we establish that this optimized configuration maintains its effectiveness across a range of 1-10 signals by making appropriate adjustments to the input and output layers.
3) We analyze model complexity and benchmark it against state-of-the-art NNs and conventional algorithms, focusing on the time required for prediction.
4) We validate the model's accuracy across various scenarios, including different SNR environments and varying numbers of simultaneously incoming signals. Our results conclusively demonstrate the establishment of a noise-independent, low-complexity, and precision-focused DoA estimator.

The remaining sections of this paper are organized as follows: In Section II, we survey prior work, thus offering a succinct overview of existing literature. In Section III, we lay the groundwork by introducing the basic DoA estimation system and its underlying principles as well as the signal model. In Section IV, we delve into the data model, elucidating the signal representations and preprocessing techniques essential for training and evaluating the DCNN. Section V is dedicated to describing the DCNN architecture, where we carefully discuss the design choices for the parameters and hyperparameters of the model. In Section VI, we analyze the training process of the DCNN and present the obtained results. In Section VII, we concentrate on validating the performance of the DCNN. We highlight the model's capability to sustain performance regardless of the presence of noise and its proficiency in managing simultaneously received signals. Moreover, we emphasize its computational efficiency, particularly in terms of the time required for estimation. Finally, in Section VIII, we conclude the paper by summarizing our key findings and presenting the implications of our results.

II. PRIOR WORK

Several academic endeavors have been instrumental in guiding the course of DoA estimation research. A pioneering work is presented in [27], where a neural network-based smart antenna designed for multiple source tracking is introduced. This work serves as a foundational contribution, paving the way for subsequent advancements in smart antenna technology through the integration of NNs. The efficacy of convolutional recurrent neural networks (CRNNs) within

challenging acoustic environments is illuminated in [28], while scholarly undertakings exemplified by [29] and [30] direct attention to the robustness, noise-resistance and applicability of CNNs in the domains of broadband DoA estimation and source counting. Huang et al.'s research [31] delves into the domain of high resolution channel estimation and DoA estimation within massive MIMO systems, offering insights into the applications of DL for enhanced system performance. [32] contributes to the field by addressing the challenges posed by array imperfections, with a focus on developing a DoA estimation method based on deep neural networks (DNNs) with inherent robustness to imperfections in antenna arrays. Wu et al.'s study [33] introduces a DCNN that demonstrates the potential of DL in leveraging sparse knowledge for accurate estimations. Moreover, Massa et al.'s comprehensive review [34] surveys the broader landscape of applying DNNs to electromagnetics and antennas, offering a panoramic understanding of the evolving field. Lastly, [35] presents a ML approach to DoA estimation in noisy environments, addressing a critical challenge in real-world applications. Various endeavors [36], [37], [38], [39] have, also, focused on minimizing computational overhead and processing time of DoA estimation.

We should note that while most existing bibliography focuses on the case of linear antenna arrays, only a few delve into deeper configurations of antenna arrays, such as circular or planar arrays, in an effort to approach the DoA estimation challenge in the three-dimensional space. Zaharis et al. [40] make a notable contribution to the field by suggesting a modification to conventional beamforming methods. This modification is specifically designed for realistic linear antenna arrays, placing a strong emphasis on practical applicability and spatial signal processing. [41] introduces a discretization-free sparse and parametric approach for linear array signal processing, establishing its statistical reliability in the number of snapshots under uncorrelated sources. The application and evolving significance of CNNs for DoA estimation in linear antenna arrays is further explored in [42]. Lota et al. [43] present an exploration of 5G uniform linear arrays, employing beamforming and spatial multiplexing for outdoor urban communication at various GHz frequencies, showcasing a robust two-level approach. Researchers in [44], [45] leverage deep networks for DoA estimation in low SNR scenarios, demonstrating the adaptability of DL techniques in antenna arrays, whereas in [46] there is a focus on near-field DoA estimation in MIMO systems by a complex ResNet framework.

In terms of implementations within 3D space, [47] proposes a joint DoA estimation and phase calibration approach for uniform rectangular arrays, offering a comprehensive solution for improved array processing precision. Wei et al. [48] contribute insights into optimizing planar array configurations for direction of arrival estimation, while Wang et al. [49] showcase advancements in positioning technologies by introducing a multitarget active backscattering positioning system with superresolution time series post-processing. A multiport interferometer-enabled 3D DoA estimation system is introduced in [50], which enhances the accuracy of existing estimators. Finally, Xu et al. [51] present an application-oriented study on DoA-based positioning, employing uniform circular arrays, thus providing insights into the practical utilization of circular array configurations for positioning in communication technologies.

Distinguished from most previous works, our research underscores a strategic focus on enriching the representation of three-dimensional space, encompassing a diverse range of angles, which is crucial in the landscape of 5G and beyond 5G (B5G) networks. This inclusivity spans the elevation and azimuth planes, ensuring a comprehensive and adaptable framework for modern communication systems. Integral to our methodology is the deployment of a 16-element planar antenna array, a pivotal enabler for our advanced signal processing procedures. With an emphasis on precision, our estimator is meticulously crafted to exhibit notable noise independence, which becomes a critical attribute in dynamic and often noisy environments. Our approach operates under the assumption that the precise number of incoming signals is known, as we do not engage in its calculation. Although the completeness of the receiver block requires extensive exploration in various areas, our emphasis lies in refining the estimation process. The proposed model's proficiency extends to scenarios characterized by the simultaneous reception of signals, an inherent trait in real-world applications. For example, in the context of vehicular communications, where signal interference and simultaneous reception are commonplace, our research stands out as particularly pertinent. These distinctive capabilities position our research as a valuable asset for addressing the evolving demands of contemporary communication systems.

III. SIGNAL MODEL

The incoming signals are described by respective angle pairs denoted as (θ_i, ϕ_i). Let's consider an antenna array that receives N monochromatic signals, denoted as $s_i(t)$ $(i = 1, \ldots, N)$. Alternatively, we can represent these signals in a sample form, where $s_i(k)$ corresponds to a signal sample of order k. For each sample $s_i(k)$, a corresponding modulation function $g_i(k)$ exists. These modulation functions encompass all the information transferred by the incoming signals and vary depending on the multiplexing technique employed for broadband transmission (e.g., OFDMA). The incoming signals originate from specific AoAs and can be represented in the spherical coordinate system by a radial unit vector (see Fig. 1)

$$\mathbf{v}_i = \cos\phi_i \sin\theta_i \mathbf{x}_0 + \sin\phi_i \cos\theta_i \mathbf{y}_0 + \cos\theta_i \mathbf{z}_0, \qquad (1)$$

where $\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0$ are the respective unit vectors of x, y and z-axes. Apart from the modulation function, each signal is

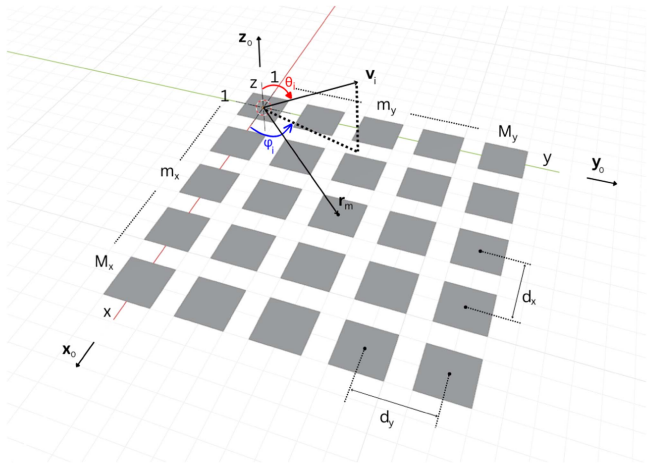

FIGURE 1. Three-dimensional plot of a $M_x \times M_y$ antenna array, and an incoming signal described by a radial unit vector that corresponds to angle pair (θ_i, ϕ_i).

associated with a "steering vector"

$$\alpha_n = \begin{bmatrix} \exp(j\beta \mathbf{r}_1 \mathbf{v}_i) \\ \exp(j\beta \mathbf{r}_2 \mathbf{v}_i) \\ \vdots \\ \exp(j\beta \mathbf{r}_M \mathbf{v}_i) \end{bmatrix}, \tag{2}$$

where β is the propagation constant and

$$\begin{aligned} \mathbf{r}_m &= (m_x - 1)d_x \mathbf{x}_0 + (m_y - 1)d_y \mathbf{y}_0, \\ m_x &= 1, \ldots, M_x \quad \text{and} \quad m_y = 1, \ldots, M_y, \end{aligned} \tag{3}$$

is the position vector of the mth array element, with $m = 1, \ldots, M$ and $M = M_x M_y$ denoting the total number of the array elements [52], [53], [54]. Based on the steering vectors of all incoming signals, an $M \times N$ "steering matrix" is defined as

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \exp(j\beta \mathbf{r}_1 \mathbf{v}_1) & \cdots & \exp(j\beta \mathbf{r}_1 \mathbf{v}_N) \\ \vdots & \ddots & \vdots \\ \exp(j\beta \mathbf{r}_M \mathbf{v}_1) & \cdots & \exp(j\beta \mathbf{r}_M \mathbf{v}_N) \end{bmatrix} \\ &= \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_N \end{bmatrix}. \end{aligned} \tag{4}$$

According to the above equations, we can define signals x_m $(m = 1, \ldots, M)$ induced at the inputs of the antenna array elements in a vector form as

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) & x_2(k) & \cdots & x_M(k) \end{bmatrix}^T, \tag{5}$$

where T indicates the transpose operation. This vector can be calculated according to the following expression:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{g}(k) + \mathbf{n}(k) = \mathbf{x}_s(k) + \mathbf{n}(k), \tag{6}$$

where $\mathbf{x}_s(k)$ represents the component of the input vector solely attributed to incoming signals,

$$\mathbf{g}(k) = \begin{bmatrix} g_1(k) & g_2(k) & \cdots & g_N(k) \end{bmatrix}^T \tag{7}$$

is the modulating vector containing the modulation functions of all incoming signals, and

$$\mathbf{n}(k) = \begin{bmatrix} n_1(k) & n_2(k) & \cdots & n_M(k) \end{bmatrix}^T \tag{8}$$

is the vector of noise samples entering the inputs of the antenna array elements. According to the process followed by several DoA estimation algorithms, each input signal x_m is multiplied by the complex conjugate value of a weight w_m. Consequently, the resulting output signal is given by the equation

$$y(k) = \sum_{m=1}^{M} w_m x_m(k), \tag{9}$$

which can also be expressed as

$$y(k) = \mathbf{w}^H \mathbf{x}(k), \tag{10}$$

where

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_M \end{bmatrix}^T \tag{11}$$

is the vector of complex weights and index H indicates the conjugate transpose operation. Equation (10) demonstrates that the output is expressed as the inner product of vectors $\mathbf{w}$ and $\mathbf{x}(k)$. Then, the mean output power is expressed as

$$\hat{P}_y = \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w}, \tag{12}$$

where

$$\mathbf{R}_{xx} = E[\mathbf{x}(k)\mathbf{x}^H(k)] \tag{13}$$

is an $M \times M$ hermitian matrix known as the correlation matrix of the antenna array input signals and $E[.]$ represents the expected value. Assuming the source signals s_n are uncorrelated with each other and with the noise signals, (13) simplifies to

$$\mathbf{R}_{xx} = \mathbf{A}\mathbf{R}_{gg}\mathbf{A}^H + \mathbf{R}_{nn} = \mathbf{R}_{ss} + \mathbf{R}_{nn}, \tag{14}$$

where $\mathbf{R}_{ss}$ is the correlation matrix of the components of the input signals attributed solely to the incoming signals,

$$\mathbf{R}_{gg} = E[\mathbf{g}(k)\mathbf{g}^H(k)] \tag{15}$$

is the correlation matrix of the modulation functions of the incoming signals, and

$$\mathbf{R}_{nn} = E[\mathbf{n}(k)\mathbf{n}^H(k)] \tag{16}$$

is the correlation matrix of the noise signals. Now, assuming the noise signals have zero mean value and variance σ^2, (16) becomes

$$\mathbf{R}_{nn} = \sigma^2 \mathbf{I}_{M \times M}, \tag{17}$$

where $\mathbf{I}_{M \times M}$ is the $M \times M$ identity matrix.

IV. DATA MODEL

The fundamental step for formulating the problem is to create a dataset that includes the necessary information and serves as the basis of our simulation. This dataset is used as the input of the DCNN and comprises multiple sets of incoming signals, each corresponding to various noise levels ranging from

−10 to 10 dB. The intentional incorporation of varied noise levels was implemented to authentically replicate real-world conditions. For each set of incoming signals we record the $\mathbf{R}_{xx}$ matrix which is created. Alongside this matrix, we maintain an additional $S \times N$ matrix, where S represents the total number of distinct angle sets, and N signifies the total number of signals per set. This $S \times N$ matrix contains the angles of arrival (AoAs) of the signals, denoted as (θ_i, ϕ_i), $i = 1, \ldots, N$, and assumes a pivotal role in performance validation, facilitating a comparison between the actual AoAs and those predicted by the DCNN.

Assuming a 4×4 planar antenna array as the receiver of the signals, $\mathbf{R}_{xx}$ becomes a 16×16 matrix, with elements representing complex numbers. Neural networks are generally not well-suited to directly process complex numbers, so we need to split the matrix into real and imaginary parts [55]. This splitting operation produces two sub-matrices; one containing the real values and the other containing the imaginary values. Moreover, we incorporate an additional 16×16 matrix to preserve the essential phase information inherent in the complex values of each element within the original matrix. This process allows us to effectively handle complex data while leveraging the benefits of the DCNN. As a result of this data manipulation, the final dimensions of the dataset become $S \times 16 \times 16 \times 3$.

A sufficiently large set of angle pairs is required to ensure diversity and variations from record to record. Our analysis involves the examination of 1 to 10 simultaneously incoming signals. The dataset will encompass 10^4 records for each possible number of incoming signals, resulting in a total of 10^5 records ($S = 10^5$). In our analysis, each azimuth angle lies within the range $[0°, 360°)$, whereas each polar angle belongs to the range $[0°, 60°]$. As we mentioned in Section II., these angle ranges reflect realistic conditions of incoming signals received by antenna arrays in 5G and B5G applications. The AoAs within the dataset undergo a precision adjustment, where they are rounded to the nearest tenth decimal digit. This ensures a standardized level of granularity in the representation of spatial information. Assuming, for example, a total of number of three incoming signals ($N = 3$), a possible set of angles (θ_i, ϕ_i), $i = 1, 2, 3$, would be:

$$\Big[(30.6°, 128.2°) \quad (9.3°, 89.1°) \quad (58.5°, 12.8°)\Big].$$

Matrices $\mathbf{R}_{gg}$ and $\mathbf{R}_{nn}$ are derived from identity matrices multiplied by a power level, whose value is determined by the respective SNR. For power levels P_g and $P_n = \sigma^2$, respectively, the matrices are given as

$$\mathbf{R}_{gg} = P_g \mathbf{I}_{N \times N} \tag{18}$$

and

$$\mathbf{R}_{nn} = P_n \mathbf{I}_{M \times M}. \tag{19}$$

Using this information, the correlation matrix is constructed based on (14), which allows the detection of a number of signals either equal to or less than the total number of elements in the antenna array.

V. DCNN ARCHITECTURE

A. LOSS AND ACCURACY FUNCTIONS

Binary cross-entropy serves as the appropriate loss function, and binary accuracy emerges as the fitting metric when treating each angle prediction as a binary classification task [56]. Binary cross-entropy, also known as log loss, measures the difference between the predicted probabilities and the true labels, assessing the performance of the model's predictions, and is given as

$$\text{Log Loss} = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}), \tag{20}$$

where y represents the actual label of the binary classification problem and $\hat{y}$ denotes the predicted probability of the positive class. On the other hand, binary accuracy calculates the accuracy of the model in predicting the correct class for each angle, considering a threshold of 0.5 to assign the class, and can be expressed as

$$\text{Binary Accuracy} = \frac{\text{Correctly predicted records}}{\text{Total number of records}}. \tag{21}$$

However, these functions might not fully capture the nuances of a model's behavior, especially in scenarios with imbalanced datasets like in our case, where most neurons are not expressing a DoA of an incoming signal. In such scenarios, binary accuracy can be misleading. Let's assume one class is predominant. Then, a model that always predicts this class could achieve high accuracy according to binary accuracy metric, even though it fails to correctly classify instances from the minority class. In these cases, metrics like precision, recall, and F1-score are more reliable. Precision represents the proportion of correctly predicted positive instances among all predicted positives, whereas recall quantifies the proportion of correctly predicted positive instances among all actual positives. These metrics offer an in-depth evaluation of the model's performance across individual classes. Hence, they effectively address challenges related to class imbalance. F1-score, which is a combination of precision and recall, offers a single value that balances the trade-off between these two metrics by using the harmonic mean of them and is, therefore, given as

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{22}$$

B. HYPERPARAMETER TUNING AND ARCHITECTURE OVERVIEW

In the pursuit of achieving optimal performance for our DCNN, we carefully consider a range of hyperparameters. These hyperparameters encompass crucial aspects, such as the number of layers, the number of filters in each layer, kernel size, kernel stride, dropout rate, regularizer rate, and learning rate. The selection of the appropriate activation function and optimizer is equally important. To determine the most effective values for these hyperparameters, we employ the

TABLE 1. F1-Score and Number of Parameters vs Number of Filters, for 2 Convolutional Layers

Number of filters	F1-Score	Number of parameters
128	98.78%	15,068,080
256	99.46%	24,706,608
512	99.04%	44,376,880
1024	98.89%	85,290,288

TABLE 2. F1-Score and Number of Parameters vs Number of Filters, for 3 Convolutional Layers

Number of filters	F1-Score	Number of parameters
128	98.38%	12,250,672
256	99.99%	19,202,864
512	99.74%	33,893,680
1024	98.70%	66,421,040

TABLE 3. F1-Score and Number of Parameters vs Number of Filters, for 4 Convolutional Layers

Number of filters	F1-Score	Number of parameters
128	97.79%	9,957,552
256	98.40%	14,747,696
512	98.51%	25,507,632
1024	97.96%	51,746,096

TABLE 4. F1-Score and Number of Parameters vs Number of Filters, for 5 Convolutional Layers

Number of filters	F1-Score	Number of parameters
128	97.22%	8,188,720
256	97.85%	11,341,104
512	98.12%	19,218,736
1024	97.77%	41,265,456

grid search optimization technique. This meticulous approach involves systematically exploring various combinations of hyperparameter values, aiming to pinpoint the specific configuration that not only maximizes accuracy, but also ensures the robustness of the model. Within the context of the complexity/performance trade-off in NNs, it is crucial to clarify that a higher level of complexity typically leads to enhanced performance. This advantage, however, is accompanied by the trade-off of deploying more resources and accommodating longer time responses. Conversely, opting for lower complexity might result in quicker responses and the need for fewer resources, but it could potentially come at the expense of compromising overall performance. Therefore, through optimization and hyperparameter tuning, our goal is to strike a balance, achieving the highest precision in our model's predictions, while maintaining minimal complexity.

The balance between information and complexity comes into focus when examining the R_{xx} matrix, which serves as the input of our proposed NN. An influx of incoming signals enhances the informational aspect, though it also amplifies complexity as the dimensions of the R_{xx} matrix grow. In addressing this trade-off, our deliberate focus is on optimizing performance for a specific scenario involving three incoming signals. Our architectural design undergoes meticulous fine-tuning and optimization for three incoming signals. It is crucial to note that the design is intentionally not individually tailored for each scenario involving different numbers of incoming signals. This deliberate simplification ensures consistency across the hyperparameters of the hidden layers, excluding the input and output layers, regardless of the varying number of incoming signals. The balance and efficiency of this approach will be evident in the results presented in Section VII.

Initially, we focus on meticulously determining the optimal number of layers and filters, given their fundamental role as the cornerstone of any CNN. The examination of Tables 1–4 reveals patterns concerning the impact of these hyperparameters on the feature extraction capacity and accuracy of the model. In instances where the convolutional layers are inadequate in number, the model's capacity for feature extraction becomes constrained. This limitation in feature extraction capacity is consequential, leading to a substantial decrease in overall accuracy. On the contrary, an excess of convolutional layers may give rise to overfitting, resulting in a reduction of parameters and a subsequent decline in accuracy. After a thorough evaluation, it is evident that an optimal configuration involves an architecture consisting of three convolutional layers, each equipped with 256 filters. This choice is substantiated by the understanding that a lower quantity, i.e., 128 filters, results in insufficient feature extraction, and compromises the model's accuracy.

Conversely, an excessive number of filters, i.e, 512 or 1024 filters, introduces redundancy, and leads to reduced overall efficiency. The decision to employ three convolutional layers with 256 filters each, signifies a strategic compromise, striking a balance between robust feature extraction and computational efficiency to enhance performance. This becomes more obvious, through the observation of a decline in F1-Score values when deviating from the proposed configuration. We should note that while the F1-Score may consistently register values higher than 97%, this should not be misleading. It is crucial to keep in mind the class imbalance, which may result in the model exhibiting high accuracy in predicting non-DoA estimations, although it may be comparatively weak in classifying the actual DoAs – which is the primary objective. The proposed architecture achieves an impressive F1-Score of 99.99%, highlighting its noteworthy performance. This remarkable level indicates the model's effective handling and mitigation of issues associated with uneven distribution among classes.

Regarding the kernel size, we explore different options and assess their impact on the model's ability to capture meaningful information from the correlation matrix. Through the grid

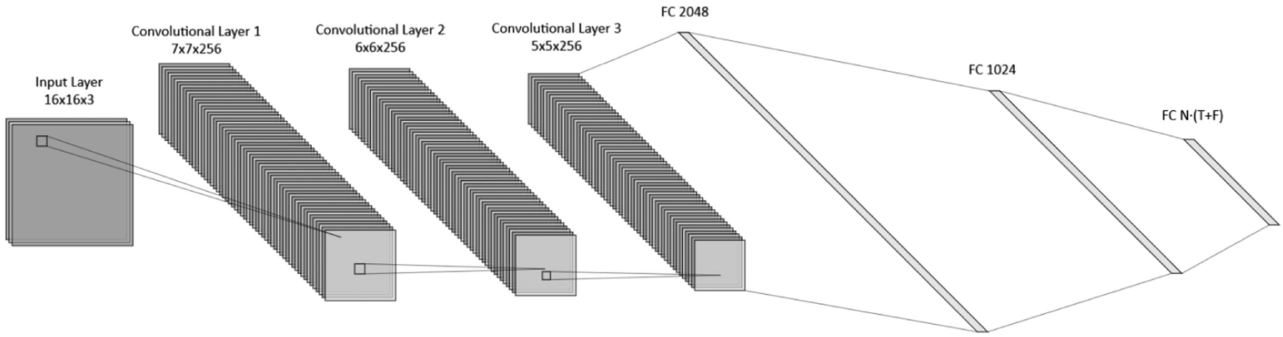

FIGURE 2. Depiction of the DCNN architecture, consisting of the input layer, 3 convolutional layers, and 3 fully connected layers, with the last one serving as the classification layer.

search optimization, the 3×3 kernel size for the first convolutional layer and the subsequent 2×2 kernel size for the next two layers proves to be effective in capturing local features and spatial relationships within the input data. To improve computational efficiency, even further, without compromising performance, we introduce a kernel stride with a value of 2 for the first convolutional layer. This enables skipping certain convolution operations, effectively reducing the spatial dimensions of the feature maps. The subsequent two convolutional layers use a default stride value of 1, hence ensuring a more detailed analysis of the features. The dropout rate and the regularization rate of the L2 regularizer are set at 0.2 and 0.001, respectively, aiming to prevent unnecessary network overload and enhance the precision of the model. These values, have also emerged as the optimal outcomes from the grid search process. Furthermore, the Adam optimizer has been identified as the most effective choice for optimizing our model's performance. For the learning rate, we utilize the "ReduceLROnPlateau" function provided by Keras. Although the initial value for the learning rate is set at 0.001, this function monitors the accuracy of the model during the training phase and adjusts the learning rate when accuracy stabilizes for a certain number of epochs. After each convolutional layer, a batch normalization layer is applied to the output. Batch normalization normalizes the activations, stabilizes the training process, and accelerates convergence. By reducing internal covariate shift, it enables the use of higher learning rates, thus leading to faster training processes. It, also, acts as a regularization technique, improving the model's generalization performance. Since the initial input dimension is smaller than typical pooling layer applications, pooling layers are not utilized.

After passing through the last convolutional layer, the resulting output is directed to FC layers, which are preceded by a flattening operation to convert the output into a vector format. Three FC layers are employed in the proposed DCNN architecture. The first two have 2048 and 1024 neurons, respectively. Dropout layers, with a dropout rate of 0.2, are added after these FC layers, to enhance training speed and limit the risk of overfitting. As for the activation function, rectified linear unit (ReLU) is applied in every convolutional and FC layer apart from the output layer. ReLU introduces non-linearity to the network, allowing it to learn complex patterns. It efficiently solves the vanishing gradient problem that can occur in deep networks, thus ensuring effective propagation of gradients during backpropagation. The last FC layer serves as the classification layer of our DCNN. The dimensions of this layer depend on the number of AoAs requiring estimation. Fig. 2 illustrates the proposed DCNN architecture. In the subsequent section we will delve into the specifics of the network's output and the interpretation of the results.

C. CLASSIFICATION LAYER

For the classification process, we establish a mapping of each unique angle ranging in $\theta \in [0^\circ, 60^\circ]$ and $\varphi \in [0^\circ, 360^\circ)$ to a finite number of classes represented as 0 or 1. If a neuron of the output layer corresponds to an angle that is not predicted as the desired DoA, it will be classified as 0. On the other hand, if the neuron represents the DoA of the DCNN's prediction, it will be classified as 1. This mapping should be performed for each neuron of the grid.

The total number of neurons in the NN's output layer is another critical trade-off between accuracy and complexity of the DCNN, and it determines the grid resolution, which divides the continuous angle range into discrete values. Increasing the number of neurons allows for a finer grid resolution, which can potentially lead to higher accuracy in predicting AoAs. However, as the number of neurons increases, so does the total number of parameters in the network. Each additional neuron introduces more weights and biases that need to be learned during the training process. This leads to a more complex model with higher computational costs and memory requirements. It is crucial to consider that after a certain point, increasing the grid resolution may not result in significant improvements in accuracy. In fact, apart from excessive computational costs, the network may become overfit to the training data. Excessively high grid resolution can introduce noise and small variations in the input data. These minor fluctuations can cause the model to become sensitive to irrelevant details, thus leading to reduced robustness and poor performance on its predictions. We will delve deeper into this

matter in Section VI, where we define the grid resolution of the output layer and compare the outcomes of each scenario.

The output layer of the proposed DCNN employs the sigmoid activation function, which is suitable for the binary classification implementation. The sigmoid function performs a transformative role by mapping input values to a confined range between 0 and 1, a characteristic that aligns seamlessly with our objective of binary classification, distinguishing between two classes–specifically, 0 and 1. This activation function proves invaluable in endowing our DCNN with the capacity to generate probability scores for each neuron in the grid, offering a nuanced perspective on the likelihood of a neuron accurately representing the desired DoA. Values near 0 indicate a low probability and values approaching 1 signal a high likelihood, the sigmoid function renders the DCNN's output interpretable in terms of confidence levels. This interpretability becomes particularly pertinent as it allows us to establish precise thresholds on the predicted probabilities, facilitating meticulous binary decisions for every neuron within the grid. By embracing the sigmoid activation function, our DCNN not only navigates the complexities of DoA estimation but also introduces a level of transparency and control over the classification outputs. The output layer is designed to have this format:

$$\Big[(\theta_1, \varphi_1), \quad (\theta_2, \varphi_2), \ldots, \quad (\theta_N, \varphi_N)\Big]$$

To enhance clarity, let's assume a scenario where we use a resolution step of 1° for the grid and we consider the case of 2 incoming signals ($N = 2$) to comprehend the structure of the output layer. At this point, we should remind that the angle ranges are $\theta \in [0^\circ, 60^\circ]$ and $\varphi \in [0^\circ, 360^\circ)$. This means that in this scenario, the output layer will consist of 842 neurons. This occurs because the model predicts the angle pairs (θ_1, ϕ_1) and (θ_2, ϕ_2), hence the total grid of 842 neurons is divided into four subgrids. The first subgrid contains 61 neurons and predicts θ_1, the second contains 360 neurons and predicts ϕ_1, the third contains 61 neurons and predicts θ_2, and the fourth contains 360 neurons and predicts ϕ_2. The process remains the same for more incoming signals, with appropriate adjustments to the output layer dimensions. In general, the output layer dimensions can be expressed as $N(T + F)$, where T represents the total number of neurons for the polar angles and F represents the total number of neurons for the azimuth angles (T and F depend on the resolution step).

In the case of two incoming signals, Algorithm 1 is introduced to outline the data processing strategy for searching nodes within the output layer of the proposed neural network. The output layer is divided into four subgrids, and the algorithm efficiently traverses each subgrid, inspecting whether individual nodes carry the value of 1, indicative of classification as an AoA. Noteworthy is the algorithm's provision for adjusting return values based on the subgrid, ensuring accurate mapping to the corresponding θ or ϕ angle within the 3D space. This refined methodology preserves the algorithm's efficacy in pinpointing nodes that fulfill the specified criteria.

Algorithm 1: Post-Processing for Predicting AoAs in the Case of Two Incoming Signals.

Input: Output layer with 842 nodes
Output: Nodes with the value of 1
Divide the output layer into four subgrids
for $j \leftarrow 1$ to 4 **do**
  Determine the range of nodes for the jth subgrid
  if $j = 1$ **then**
    Set $start\_index$ to 1 and end_index to 61
  else if $j = 2$ **then**
    Set $start\_index$ to 62 and end_index to 421
  else if $j = 3$ **then**
    Set $start\_index$ to 422 and end_index to 483
  else
    Set $start\_index$ to 484 and end_index to 842
  end if
  for $i \leftarrow start\_index$ to end_index **do**
    Check if node i is classified as '1'
    if Condition is met **then**
      if $j = 1$ **then**
        Output: Return $i - 1$
      else if $j = 2$ **then**
        Output: Return $i - 62$
      else if $j = 3$ **then**
        Output: Return $i - 422$
      else
        Output: Return $i - 484$
      end if
    end if
  end for
end for

VI. TRAINING PHASE

A. TRAINING RESULTS

In Figs. 3 and 4, loss and accuracy of the NN are presented for both training and validation datasets. Prior to this analysis, the entire dataset was divided, allocating 90% for training and 10% for validation purposes. After completing the network's training with 120 epochs, noteworthy observations can be made, regarding the accuracy and loss metrics for both datasets. The accuracy for the training dataset reaches a high value of 0.9995 (99.95%), and the corresponding loss is 7.25×10^{-4}. The F1-score also demonstrates results close to 99.98%, further affirming the performance of the model and indicating that the network has successfully learned from the training data and can accurately predict the majority of records.

The validation dataset exhibits even higher accuracy, with values close to 99.99%, and loss values of 2.95×10^{-4}. This observation is remarkable, as it indicates that the network generalizes exceptionally well to unknown data. The loss value of 2.95×10^{-4} further supports the notion that the network's predictions closely match the true labels for the majority of validation records. Between 80 and 120 epochs, both loss and

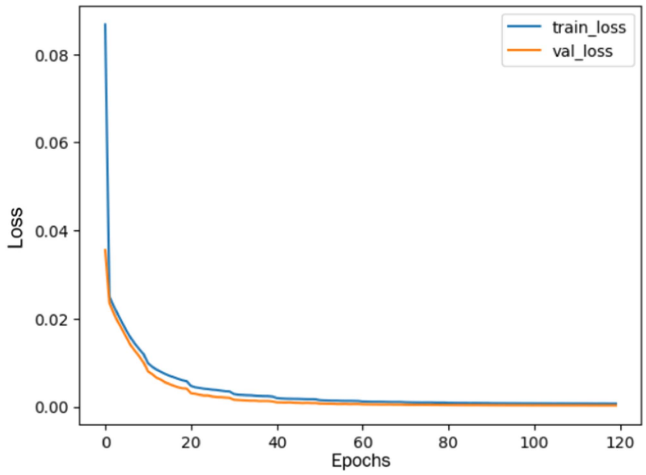

FIGURE 3. Training and validation loss with respect to the epochs of training.

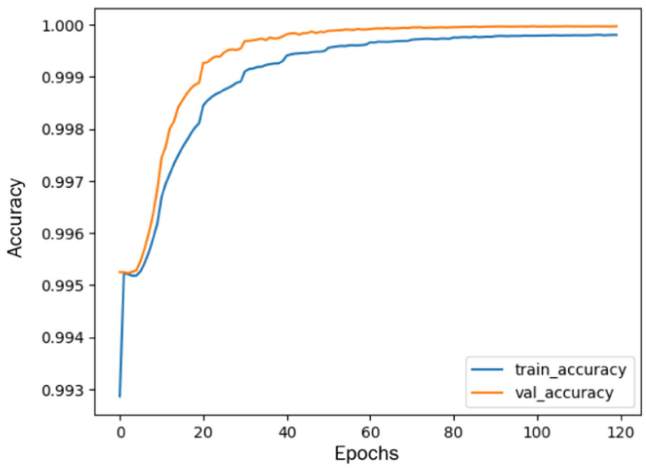

FIGURE 4. Training and validation accuracy with respect to the epochs of training.

accuracy metrics reach a point where they seem to stabilize. Beyond this point, there are no significant improvements in the model's performance. Therefore, in the subsequent analysis, we train the model only for 80 epochs in order to save computational resources.

B. EVALUATION METRICS

To evaluate the NN's performance in estimating AoAs, we employ two key metrics: the root mean squared error (RMSE) and the mean absolute error (MAE). Unlike accuracy and loss functions presented in Section V, which assess the correctness of binary classification, RMSE and MAE enable the measurement of the error between the predicted and actual AoAs in degrees.

RMSE is advantageous for its sensitivity to larger errors, as it exhibits them through squaring. This characteristic becomes crucial when underscoring the importance of precise predictions, particularly in our endeavor, where the prohibition of extreme errors in predicted AoAs is a critical requirement. The squaring operation in RMSE ensures that larger errors have a more pronounced impact on the overall evaluation, allowing for a focused assessment of the model's performance in handling significant deviations. On the other hand, MAE treats all errors equally, providing an average absolute deviation without magnifying the impact of outliers. This makes MAE more robust to extreme values, ensuring that the evaluation is not disproportionately influenced by outliers. We compute both RMSE and MAE, ensuring a comprehensive assessment of the model's performance by including all relevant graphs. This approach offers a thorough overview of how well the model predicts and captures errors across various scenarios. Here, RMSE is defined as

$$\text{RMSE}=\sqrt{\frac{1}{2NL}\sum_{i=1}^{L}\sum_{j=1}^{N}\left[\left(\theta_{ij}-\hat{\theta}_{ij}\right)^2+\left(\phi_{ij}-\hat{\phi}_{ij}\right)^2\right]}, \tag{23}$$

where N is the total number of incoming signals per scenario, $2N$ is the total number of predictions made (for polar and azimuth angles), L is the total number of records (scenarios), θ_{ij} and ϕ_{ij} represent the actual AoA values (polar and azimuth angles, respectively), whereas $\hat{\theta}_{ij}$ and $\hat{\phi}_{ij}$ represent the predicted AoA values. In this section, L comprises 2500 training records, whereas in Section VII, it will encompass 2500 validation records. MAE can be calculated as

$$\text{MAE}=\frac{1}{2NL}\sum_{i=1}^{L}\sum_{j=1}^{N}\left(\left|\theta_{ij}-\hat{\theta}_{ij}\right|+\left|\phi_{ij}-\hat{\phi}_{ij}\right|\right). \tag{24}$$

Our objective is to minimize these metrics, thereby reducing the disparity between the actual and predicted AoAs.

C. ERROR EVALUATION THROUGHOUT TRAINING

Our analysis is initially performed with output data discretized at 1° intervals. Given the discrete nature of the output grid, it becomes evident that discretization introduces a baseline error into the model's performance. In this context, the calculated RMSE due to discretization is approximately 0.23°, and the corresponding MAE is calculated to be 0.25°. These values represent the inherent errors associated with the discretization process and serve as a foundational reference for assessing the overall performance of our model.

Figs. 5 and 6 illustrate the overall RMSE and MAE, respectively, taking into account the cases of 2, 3, and 4 simultaneous signals. MAE starts at around 0.65° - 0.8° and decreases dramatically as the NN continues its training iterations. Notably, even from the first epoch, MAE is already below 1°, indicating promising results. The error of interest is the one after the NN's training is completed, which has even lower values. After the completion of 80 epochs of training, MAE is calculated to be 0.271° for the case of 2 incoming signals, 0.262° for 3 incoming signals, and 0.268° for 4 incoming signals. Moreover, RMSE also exhibits promising trends, registering at less than 2 degrees for each case involving 2, 3, and 4 incoming signals. Upon the completion of the training process, the final RMSE values underscore the efficacy of the model.

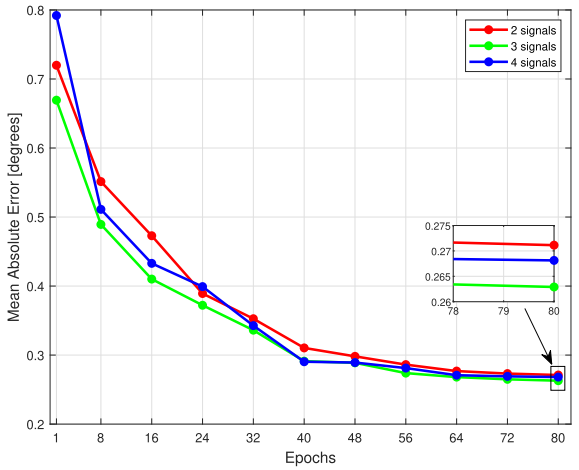

FIGURE 5. MAE evolution during training, using a grid resolution of 1°.

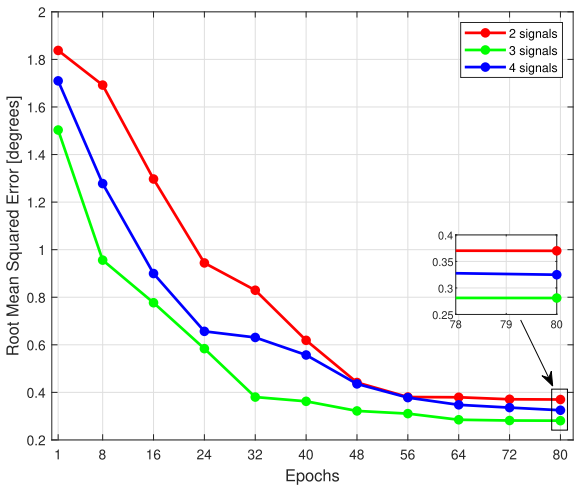

FIGURE 6. RMSE evolution during training, using a grid resolution of 1°.

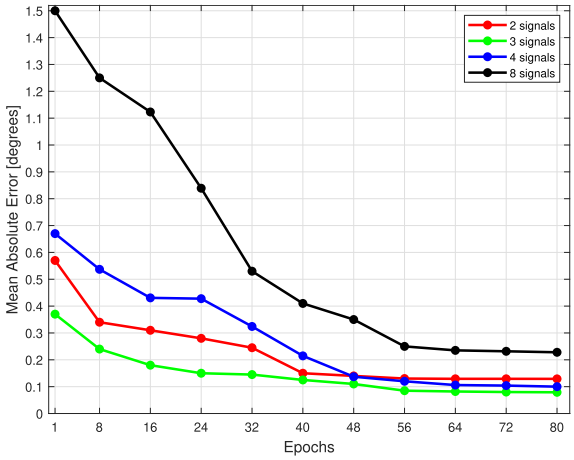

FIGURE 7. MAE evolution during training, using a grid resolution of 0.25°.

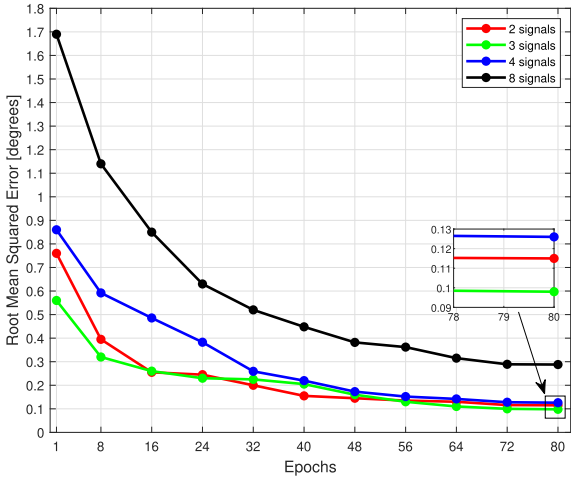

FIGURE 8. RMSE evolution during training, using a grid resolution of 0.25°.

Specifically, in the scenario of 3 incoming signals, RMSE reaches 0.256°, whereas in scenarios of 3 and 4 incoming signals, RMSE is 0.33° and 0.362°, respectively.

It is evident that the NN exhibits high accuracy in estimating the arrival directions of signals, regardless of the number of incoming signals. The error approximates the mean error introduced by the grid's discretization, which is particularly encouraging for our analysis. The minimal increase in error noticed for the case of 2 and 4 incoming signals, compared to the one of 3 signals, can be attributed to increased analysis complexity due to certain hyperparameter optimization trade-offs. Nevertheless, this increase remains negligible, and overall, the proposed DCNN demonstrates pleasing performance in analyzing incoming data.

D. REDUCED GRID RESOLUTION

Having achieved promising results with a resolution of 1°, it is reasonable to explore smaller resolution step values to minimize the error introduced by discretization and, thereby, improve the overall accuracy. For this purpose, we discretize the grid with a step of 0.25° and assess the NN's performance under this more detailed resolution. Consequently, the format of the output data, representing the angles encoded by the neurons, will be:

$$\begin{bmatrix} \dots & 31 & 31.25 & 31.5 & 31.75 & 32 & \dots \end{bmatrix}$$

and the minimum achievable error, i.e., the error introduced by the discretization, will be approximately 0.067° for MAE and 0.122° for RMSE. Under this finer resolution, RMSE is reduced in half and MAE in one-fourth of the values observed in the previous analysis of 1° resolution. Taking into consideration this error and calculating the mean MAE and RMSE of all records for every epoch, as shown in Figs. 7 and 8, we observe that errors in the first epochs of training are relatively high, but decrease and become approximately equal to the introduced errors due to discretization after the training's completion.

For 3 incoming signals the final MAE value is approximately 0.08°, whereas the final RMSE is around 0.098°. Achieving prediction accuracy less than one tenth of a degree

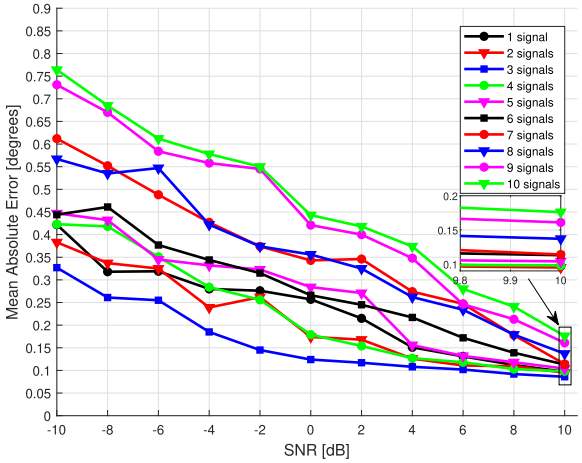

FIGURE 9. MAE vs SNR, for a range of 1 to 10 incoming signals.

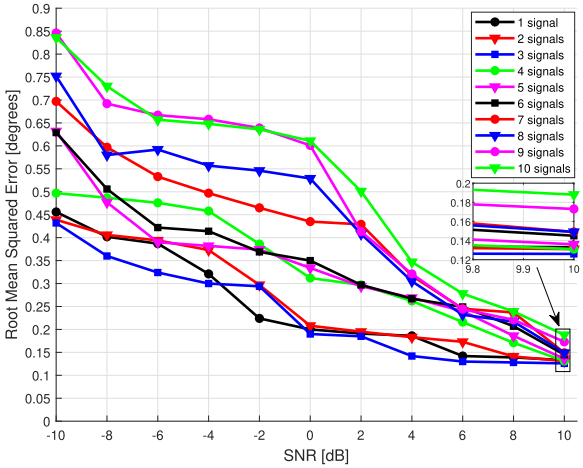

FIGURE 10. RMSE vs SNR, for a range of 1 to 10 incoming signals.

leaves us highly satisfied in terms of performance. To contrast with the errors observed in scenarios with a low number of incoming signals (2-4), we also assess the error in the scenario involving 8 incoming signals. We observe a higher error value in the early training epochs of the NN, which gradually decreases and reaches values of 0.228° for MAE and 0.29° for RMSE, by the end of the training process. Although error values are higher compared to the scenario of 2 to 4 incoming signals, they still remain at a low level.

VII. PERFORMANCE VALIDATION

After thorough consideration, we have determined that a grid resolution of 0.25° strikes a balance in the trade-off between precision and computational efficiency. This choice serves as our final decision for discretization and will be utilized in all subsequent analyses of the validation process. Critical evaluation of the proposed NN's capabilities is its performance with respect to the SNR levels. As illustrated in Fig. 9, MAE exhibits a decreasing trend as the SNR increases. For SNR equal to 10 dB a minimum MAE value of 0.098° is achieved. This value is obtained for the case of 3 incoming signals, which is the number of signals the DCNN architecture has been fine-tuned to handle. As previously stated, we have meticulously optimized and fine-tuned our architecture, specifically targeting scenarios with three incoming signals. While it may initially seem counterintuitive that the error for 3 incoming signals is lower than that for 1 or 2, the explanation lies in the consistent hyperparameter settings throughout the hidden layers (we exclude input and output layers). These settings, contributing to the distinctive behavior observed across scenarios with 1, 2, 3, or 4 incoming signals, also play a crucial role in sustaining consistently high accuracy and low error values, even when dealing with up to 10 simultaneous incoming signals.

However, even when the number of incoming signals is other than 3, MAE remains low, not exceeding an increase greater than 0.08° compared to the case of 3 incoming signals. The highest MAE value observed is 0.175°, which occurs when 10 incoming signals are received simultaneously. In an environment with high noise, there is a slight reduction in the estimation accuracy, although, the DCNN can still estimate DoAs with precision. We observe that MAE for 1 to 6 incoming signals is significantly better ($< 0.5°$) at low SNR values, in contrast to MAE obtained for 7 to 10 incoming signals, where it is greater than 0.5° but always less than 0.8°.

Similarly, in the case of RMSE illustrated in Fig. 10, we observe a diminishing trend with increasing SNR. At an SNR of 10 dB, RMSE attains a minimum value of 0.126° for the scenario involving 3 incoming signals. Irrespective of the number of incoming signals, RMSE consistently maintains a relatively low value, exhibiting an increase not exceeding 0.1° compared to the scenario of 3 incoming signals. The highest observed RMSE value in low-noise environments is not greater than 0.2°. In high-noise environments, there is a marginal decline in estimation accuracy; nevertheless, the DCNN reliably delivers accurate AoAs, consistently producing estimations below 0.86° error.

In Fig. 11, a thorough examination of performance relative to SNR underscores the pronounced superiority of the proposed DCNN over existing approaches. In comparison to the DNN proposed by G.Tang et al. [57], our DCNN showcases a discernible advantage, manifesting in a notable reduction of RMSE by several degrees. Likewise, when pitted against the traditional 2D MUSIC method, the DCNN reveals a significant edge, exemplified by a substantial decrease in RMSE.

Moreover, in high SNR scenarios, the proposed DCNN demonstrates a slight superiority over the long short-term memory neural network (LSTM NN) presented in [58], outperforming it by a fractional degree. This marginal lead becomes more prominent at lower SNR levels, where our DCNN consistently displays a more substantial enhancement over the LSTM NN, underscoring its heightened resilience and accuracy in challenging, noisy conditions.

Next, the DCNN undergoes a validation process aimed at estimating AoAs for 1 to 10 incoming signals across randomly assigned SNR values within the range of [−10 dB, 10 dB].

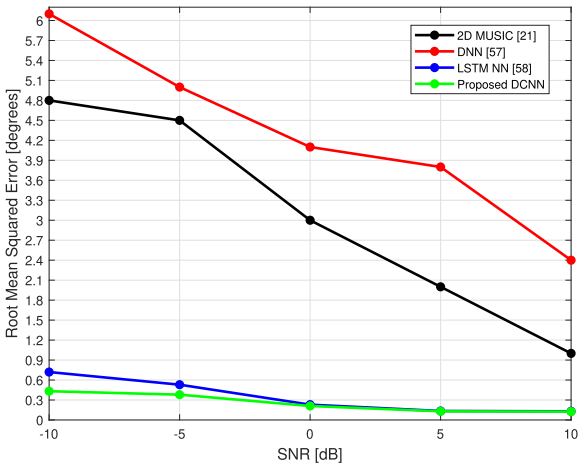

FIGURE 11. RMSE vs SNR, comparison of the proposed DCNN with state-of-the-art methods.

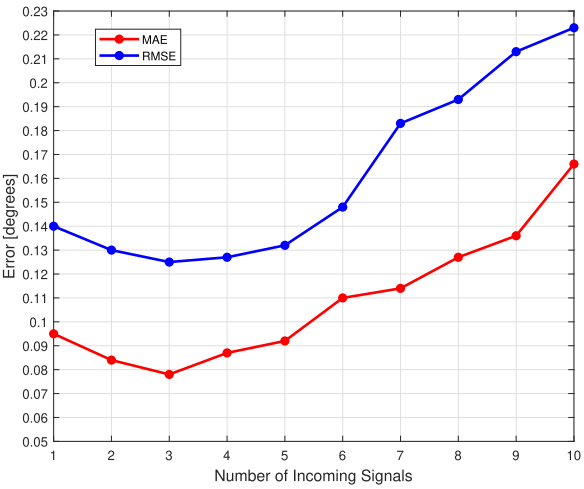

FIGURE 12. RMSE vs the number of incoming signals in a random-noise environment.

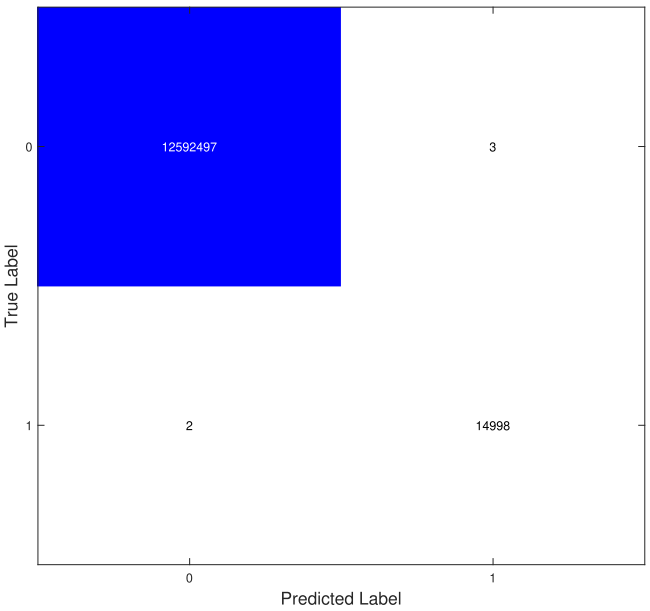

FIGURE 13. Comparison of real and predicted DoAs, using a confusion matrix for the case of 3 incoming signals and 2500 validation records.

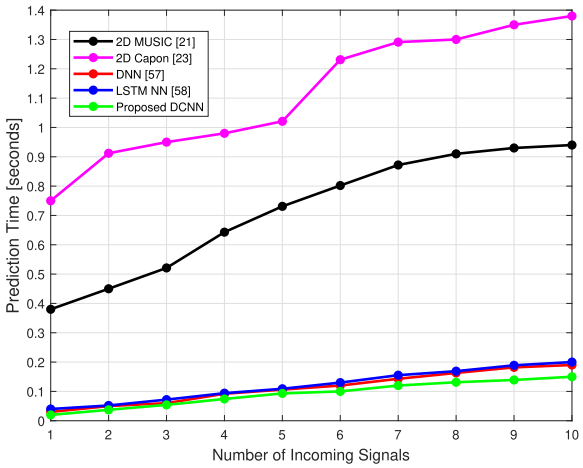

FIGURE 14. Time required by the proposed DCNN to estimate DoAs, compared to conventional and state-of-the-art algorithms.

This approach was adopted to emulate real-life scenarios, where SNR exhibits dynamic variations rather than being constant, and ensure that the network's performance remains robust in the presence of noise. The results of this simulation are shown in Fig. 12. A small, yet noticeable, improvement of approximately one hundredth of a degree is observed. The most significant achievement is the stability in performance. Even when SNR is equal to 10 dB, the error values displayed in Fig. 10 do not reach the outstanding 0.078° for MAE or the introduced limit of 0.125° for RMSE, achieved when considering a random-noise environment for 3 incoming signals, as shown in Fig. 12. As far as 1, 2, and more than 3 incoming signals are concerned, we still observe a satisfying performance by the DCNN, with error values not exceeding 0.223°, which is only 0.098° above the mean error achieved for the case of 3 signals. It is noteworthy, that for 1 to 5 signals MAE is less than a tenth of a degree.

To further illustrate the functionality of the network we examine the confusion matrix for the case of 3 incoming signals. In Fig. 13, we observe that in a validation dataset which consists of 2500 different sets, thus resulting in a total of 12,607,500 neurons for classification, the NN makes only 5 wrong predictions (neurons' classification error probability equal to $3.96 \times 10^{-5}\%$). This clarifies the reason why the overall error calculated in our analyses converges and approaches the mean error introduced by the grid's discretization.

Another equally important factor for evaluating the model is the processing time required by the DCNN to estimate AoAs. Beyond accuracy and performance, response time is a significant criterion for real-time applications, as it affects

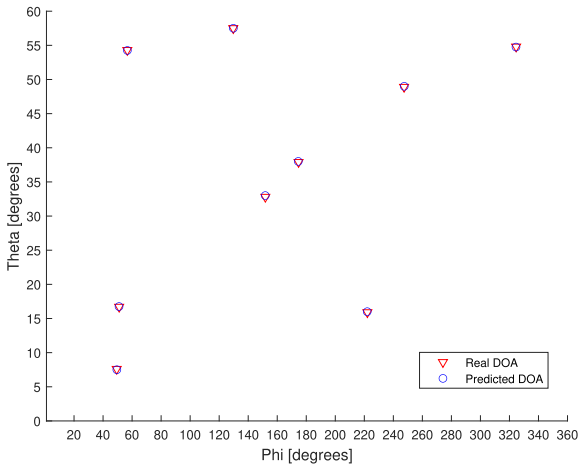

FIGURE 15. Two-dimensional visualization of actual and predicted AoAs for three records with three incoming signals per case.

the system's ability to respond promptly to channel changes. We conducted a simulation to compare the proposed DCNN approach with two conventional DoA estimation methods: 2D MUSIC and 2D Capon, as well as two state-of-the-art approaches: DNN [57] and LSTM NN [58].

Prior to examining the outcomes, we delve into a comprehensive analysis of the complexity associated with traditional algorithms versus the CNN approach, and articulate our expectations concerning prediction times. MUSIC involves computing R_{xx} and performing eigenvalue decomposition (EVD) on this matrix. The computational complexity of EVD, dominated by matrix multiplication and inversion operations, is $O(N^3)$. Capon shares similarities with MUSIC but introduces a weighted covariance matrix. Both MUSIC and Capon exhibit cubic complexities ($O(N^3)$) due to matrix operations, making them computationally demanding, especially for antenna arrays composed of many sensors. The significance of these complexities is pivotal as they directly impact the prediction process, rendering the algorithms computationally intensive and, consequently, sluggish in terms of prediction times.

In contrast, the complexity of a CNN architecture predominantly influences the training phase. Once a CNN is meticulously trained on a diverse and representative dataset, it gains the ability to swiftly and accurately make predictions. A comprehensive analysis in the CNN's complexity is indicative. In our proposed DCNN architecture, there are three convolutional layers, each employing K filters of dimensions $F_q \times F_q$, $q = 1, 2, 3$, with a stride of D, where q denotes each of the three convolutional layers. Since pooling layers are not applied, the complexity of each convolutional layer with an input of size $N \times M$ is equal to $O(\frac{N \times M \times K \times F_q \times F_q}{D^2})$, where O denotes big-O notation. Following the convolutional layers, our network includes three FC layers, where each layer has A_k neurons. The computational complexity of each FC layer is $O(A_k \times B_k)$, where B_k is the number of neurons in the previous layer. Combining the computational complexities arising from the three convolutional and three fully connected stages provides a complete perspective on the overall computational complexity of the proposed DCNN.

To conclude, CNNs exhibit computational complexities primarily tied to the convolution operations within their architecture. Unlike traditional DoA estimation algorithms, which carry cubic complexities, the operations in CNNs are typically more efficient and closer to squared complexities. The convolutional layers in CNNs are designed to capture hierarchical spatial features in data, and their time complexity is influenced by factors like input size, filter dimensions, and the number of filters. It is, also, essential to note that training an NN is computationally more intensive than inference. This distinction arises because the training process involves both the forward propagation of data through the network and the intricate backpropagation of errors, leading to repeated adjustments of the network's weights.

The superiority of ML algorithms can be established through the results of our analysis, which are depicted in Fig. 14. For 1 to 5 incoming signals, the DCNN demonstrates a response time of a few tenths of milliseconds (ms), whereas for 6 to 10 incoming signals, it takes a few hundred ms.

However, even in the case of 10 incoming signals, the DCNN's response time remains below 240 ms. The response time of the DCNN exhibits a relatively low dependence on the number of incoming signals, ranging from 1 to 10. Comparatively, the conventional 2D MUSIC method requires approximately 5 times longer time for prediction, and the 2D Capon method takes even longer, up to 7 times longer than the proposed DCNN. Both the DNN and the LSTM NN demonstrate relatively low response times, closely resembling those of the proposed DCNN. These results prove that the machine learning approach exhibits superior performance on response times. It is worth noting that these results were calculated using a AMD Ryzen 7 4800H CPU, since the metric of "seconds" for computational complexity is highly dependent on the computational resources employed.

Fig. 15 visualizes the results of the DCNN's performance by plotting the real and predicted DoA values in a two-dimensional space for three random cases with three incoming signals per case.

VIII. CONCLUSION

The presented deep convolutional neural network has shown exceptional effectiveness in DoA estimation. The proposed DCNN model achieves an impressive 99.8% accuracy in predicting DoAs with an error of less than 0.23° for 1 to 10 incoming signals in random-noise environments. Our model exhibits remarkable resilience to high noise levels. Even with increasing noise levels, the error only experiences a minimal increase, remaining below 1° for up to 5 simultaneously incoming signals. Due to the meticulous optimization of both parameters and hyperparameters, the proposed architecture showcases exceptionally low time responses. The noise robustness and the fast processing times of the model are crucial for real-world applications, as they ensure reliable, accurate,

and efficient direction finding, even in challenging and noisy environments. The DCNN's capabilities pave the way for its successful integration into diverse practical applications, such as wireless communication systems, vehicular, radar, sonar, and many others, where accurate and real-time DoA estimation is crucial for optimal performance.

REFERENCES

[1] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, Jul. 1996.

[2] A. Farina, F. Gini, and M. Greco, "DOA estimation by exploiting the amplitude modulation induced by antenna scanning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 4, pp. 1276–1286, Oct. 2002.

[3] M. Pardini, F. Lombardini, and F. Gini, "The hybrid Cramér–Rao bound on broadside DOA estimation of extended sources in presence of array errors," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1726–1730, Apr. 2008.

[4] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[5] W. Fan, X. C. B. Carreño Bautista de Lisbona, F. Sun, J. Ø. Nielsen, M. B. Knudsen, and G. F. Pedersen, "Emulating spatial characteristics of MIMO channels for OTA testing," *IEEE Trans. Antennas Propag.*, vol. 61, no. 8, pp. 4306–4314, Aug. 2013.

[6] W. Fan, F. Zhang, and Z. Wang, "Over-the-air testing of 5G communication systems: Validation of the test environment in simple-sectored multiprobe anechoic chamber setups," *IEEE Antennas Propag. Mag.*, vol. 63, no. 1, pp. 40–50, Feb. 2021.

[7] M. Pastorino and A. Randazzo, "The SVM-Based smart antenna for estimation of the directions of arrival of electromagnetic waves," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 6, pp. 1918–1925, Dec. 2006.

[8] M. Chryssomallis, "Smart antennas," *IEEE Antennas Propag. Mag.*, vol. 42, no. 3, pp. 129–136, Jun. 2000.

[9] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tut.*, vol. 18, no. 3, pp. 1617–1655, Third Quarter 2016.

[10] L. Dai et al., "Reconfigurable intelligent surface-based wireless communications: Antenna design, prototyping, and experimental results," *IEEE Access*, vol. 8, pp. 45913–45923, 2020.

[11] M. Pastorino and A. Randazzo, "A smart antenna system for direction of arrival estimation based on a support vector regression," *IEEE Trans. Antennas Propag.*, vol. 53, no. 7, pp. 2161–2168, Jul. 2005.

[12] A. Benoni, M. Salucci, G. Oliveri, P. Rocca, B. Li, and A. Massa, "Planning of EM skins for improved quality-of-service in urban areas," *IEEE Trans. Antennas Propag.*, vol. 70, no. 10, pp. 8849–8862, Oct. 2022.

[13] G. Oliveri, M. Salucci, and A. Massa, "Generalized analysis and unified design of EM skins," *IEEE Trans. Antennas Propag.*, vol. 71, no. 8, pp. 6579–6592, Aug. 2023.

[14] L. Tosi, P. Rocca, N. Anselmi, and A. Massa, "Array-antenna power-pattern analysis through quantum computing," *IEEE Trans. Antennas Propag.*, vol. 71, no. 4, pp. 3251–3259, Apr. 2023.

[15] N. Anselmi, P. Rocca, S. Feuchtinger, B. Biscontini, A. M. Barrera, and A. Massa, "Optimal capacity-driven design of aperiodic clustered phased arrays for multi-user MIMO communication systems," *IEEE Trans. Antennas Propag.*, vol. 70, no. 7, pp. 5491–5505, Jul. 2022.

[16] A. Benoni, P. Rocca, N. Anselmi, and A. Massa, "Hilbert-ordering based clustering of complex-excitations linear arrays," *IEEE Trans. Antennas Propag.*, vol. 70, no. 8, pp. 6751–6762, Aug. 2022.

[17] A. Benoni, F. Capra, M. Salucci, and A. Massa, "Towards real-world indoor smart electromagnetic environments - A large-scale experimental demonstration," *IEEE Trans. Antennas Propag.*, vol. 71, no. 11, pp. 8450–8463, Nov. 2023.

[18] M. Salucci, A. Benoni, G. Oliveri, P. Rocca, B. Li, and A. Massa, "A multihop strategy for the planning of EM skins in a smart electromagnetic environment," *IEEE Trans. Antennas Propag.*, vol. 71, no. 3, pp. 2758–2767, Mar. 2023.

[19] S. Yuan, F. Fioranelli, and A. Yarovoy, "Vehicular motion-based DOA estimation with a limited amount of snapshots for automotive MIMO radar," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 6, pp. 7611–7625, Dec. 2023.

[20] W. Zhai, X. Wang, M. S. Greco, and F. Gini, "Reinforcement learning based integrated sensing and communication for automotive MIMO radar," in *Proc. IEEE Radar Conf.*, 2023, pp. 1–6.

[21] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[22] R. R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[23] P. Handel, P. Stoica, and T. Soderstrom, "Capon method for DOA estimation: Accuracy and robustness aspects," in *Proc. IEEE Winter Workshop Nonlinear Digit. Signal Process.*, 1993, pp. P.7.1–P7.5.

[24] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations, Comput. Biol. Learn. Soc.*, 2015, pp. 1–14.

[27] A. H. El Zooghby, C. G. Christodoulou, and M. Georgiopoulos, "A neural network-based smart antenna for multiple source tracking," *IEEE Trans. Antennas Propag.*, vol. 48, no. 5, pp. 768–776, May 2000.

[28] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 1, pp. 34–48, Mar. 2019.

[29] S. Chakrabarty and E. A. P. Habets, "Broadband doa estimation using convolutional neural networks trained with noise signals," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2017, pp. 136–140.

[30] T. N. T. Nguyen, W. -S. Gan, R. Ranjan, and D. L. Jones, "Robust source counting and DOA estimation using spatial pseudo-spectrum and convolutional neural network," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2626–2637, 2020.

[31] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.

[32] Z.-M. Liu, C. Zhang, and P. S. Yu, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7315–7327, Dec. 2018.

[33] L. Wu, Z.-M. Liu, and Z.-T. Huang, "Deep convolution network for direction of arrival estimation with sparse prior," *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1688–1692, Nov. 2019.

[34] A. Massa, D. Marcantonio, X. Chen, M. Li, and M. Salucci, "DNNs as applied to electromagnetics, antennas, and propagation–A review," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2225–2229, Nov. 2019.

[35] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 2814–2818.

[36] C. Qin, J. A. Zhang, X. Huang, K. Wu, and Y. J. Guo, "Fast angle-of-arrival estimation via virtual subarrays in analog antenna array," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6425–6439, Oct. 2020.

[37] K. Wu, W. Ni, T. Su, R. P. Liu, and Y. J. Guo, "Recent breakthroughs on angle-of-arrival estimation for millimeter-wave high-speed railway communication," *IEEE Commun. Mag.*, vol. 57, no. 9, pp. 57–63, Sep. 2019.

[38] G. Wang, J. Xin, N. Zheng, and A. Sano, "Computationally efficient subspace-based method for two-dimensional direction estimation with L-shaped array," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3197–3212, Jul. 2011.

[39] M. Agatonović, Z. Stanković, and N. Dončov, "Application of artificial neural networks for efficient high-resolution 2D DOA estimation," *Radioengineering*, vol. 21, pp. 1038–1045, 2012.

[40] Z. D. Zaharis, I. P. Gravas, P. I. Lazaridis, T. V. Yioultsis, C. S. Antonopoulos, and T. D. Xenos, "An effective modification of conventional beamforming methods suitable for realistic linear antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 68, no. 7, pp. 5269–5279, Jul. 2020.

[41] Z. Yang, L. Xie, and C. Zhang, "A discretization-free sparse and parametric approach for linear array signal processing," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 4959–4973, Oct. 2014.
[42] G. Kokkinis, Z. D. Zaharis, P. I. Lazaridis, and N. V. Kantartzis, "Direction of arrival estimation applied to antenna arrays using convolutional neural networks," in *Proc. 3rd URSI Atlantic Asia Pacific Radio Sci. Meeting*, 2022, pp. 1–4.
[43] J. Lota, S. Sun, T. S. Rappaport, and A. Demosthenous, "5G uniform linear arrays with beamforming and spatial multiplexing at 28, 37, 64, and 71 GHz for outdoor urban communication: A two-level approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9972–9985, Nov. 2017.
[44] G. K. Papageorgiou, M. Sellathurai, and Y. C. Eldar, "Deep networks for direction-of-arrival estimation in low SNR," *IEEE Trans. Signal Process.*, vol. 69, pp. 3714–3729, 2021.
[45] J. Yu and Y. Wang, "Deep learning-based multipath DoAs estimation method for mmWave massive MIMO systems in low SNR," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7480–7490, Jun. 2023.
[46] Y. Cao, T. Lv, Z. Lin, P. Huang, and F. Lin, "Complex ResNet aided DoA estimation for near-field MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11139–11151, Oct. 2020.
[47] P. Heidenreich, A. M. Zoubir, and M. Rubsamen, "Joint 2-D DOA estimation and phase calibration for uniform rectangular arrays," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4683–4693, Sep. 2012.
[48] L. Wei, Q. Li, and G. Wu, "Direction of arrival estimation with uniform planar array," in *Proc. IEEE 86th Veh. Technol. Conf.*, 2017, pp. 1–5.
[49] K. Wang, J.-F. Gu, F. Ren, and K. Wu, "A multitarget active backscattering 2-D positioning system with superresolution time series post-processing technique," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 5, pp. 1751–1766, May 2017.
[50] J. Moghaddasi, T. Djerafi, and K. Wu, "Multiport interferometer-enabled 2-D angle of arrival (AOA) estimation system," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 5, pp. 1767–1779, May 2017.
[51] K. C. Xu, Y. J. Guo, X. Huang, and E. Dutkiewicz, "DoA based positioning employing uniform circular arrays," in *Proc. 11th Int. Symp. Commun. Inf. Technol.*, 2011, pp. 328–332.
[52] F. Gross, *Smart Antennas for Wireless Communications*. New York, NY, USA: McGraw-Hill Professional, 2005.
[53] L. C. Godara, *Smart Antennas*. Boca Raton, FL, USA: CRC, 2004.
[54] Z. D. Zaharis, I. P. Gravas, P. I. Lazaridis, T. V. Yioultsis, and T. D. Xenos, "Improved beamforming in 3D space applied to realistic planar antenna arrays by using the embedded element patterns," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6145–6157, Jun. 2022.
[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[56] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, Aug. 2016.
[57] G. Tang, X. Gao, Z. Chen, Y. Zhang, H. Zhong, and M. Li, "Deep neural network based multiple targets DOA estimation for millimeter-wave radar," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov.*, 2019, pp. 433–438.
[58] T. Han-Trong, N. N. Duc, H. T. Van, and H. Pham-Viet, "Direction of arrival estimation for coherent signals' method based on LSTM neural network," *Appl. Comput. Intell. Soft Comput.*, vol. 2022, 2022, Art. no. 4032419.

CONSTANTINOS M. MYLONAKIS anticipates to receive his integrated master's degree in electrical and computer engineering at Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2024. He specializes in telecommunications and his current research interests include topics of antennas and propagation, wireless communications, machine learning and signal processing.

ZAHARIAS D. ZAHARIS (Senior Member, IEEE) received the B.Sc. degree in physics, the M.Sc. degree in electronics, and the Ph.D. degree in antennas and propagation modeling for mobile communications, and the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1987, 1994, 2000, and 2011, respectively. From 2002 to 2013, he was with the Administration of the Telecommunications Network, Aristotle University of Thessaloniki. Since 2013, he has been with the School of Electrical and Computer Engineering, Aristotle University of Thessaloniki. He has been involved in several European research projects. He serves as a Principal Investigator for the Horizon 2020 MOTOR5G and Horizon 2020 RECOMBINE projects. Additionally, he holds the role of Coordinator for the Horizon Europe 6G-ICARUS and Horizon Europe ISAC-NEWTON projects. He is the author of 92 scientific journal articles, 119 international conference papers, one national patent, five book chapters, and one textbook. His research interests include design and optimization of antennas and microwave circuits, signal processing on smart antennas, development of evolutionary optimization algorithms, and neural networks. He is a Member of the Technical Chamber of Greece. Recently, he was elected Chair of the Electron Devices/Microwave Theory and Techniques/Antennas and Propagation Joint Chapter of the IEEE Greece Section. He is an Associate Editor for IEEE ACCESS.