

Received October 21, 2021, accepted November 15, 2021, date of publication November 22, 2021, date of current version December 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3129856

# Simulated Annealing Assisted Sparse Array Selection Utilizing Deep Learning

STEVEN WANDALE<sup>1</sup>, (Graduate Student Member, IEEE), AND  
KOICHI ICHIGE<sup>1</sup>, (Member, IEEE)

Department of Electrical and Computer Engineering, Graduate School of Engineering Sciences, Yokohama National University, Hodogaya-ku, Yokohama 240-8501, Japan

Corresponding author: Koichi Ichige (koichi@ynu.ac.jp)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) under Grant-in-Aid for Scientific Research 20K04500.

**ABSTRACT** This paper proposes simulated annealing (SA) assisted deep learning (DL) based sparse array selection approach. Conventional DL-based antenna selectors are primarily data-driven techniques. As a result, the required dataset is generated by listing all possible combinations of selecting  $M$  sensors given  $N$  uniform array, which is computationally expensive. A simulated annealing algorithm is proposed to assist dataset generation as an initializer to circumvent the above limitation. The SA algorithm sequentially samples and optimizes the subarrays that constitute the training data samples while retaining specific array characteristics. Hence, it simplifies the dataset annotation as most array configurations generated contain desired properties, thereby reducing the computation complexity of the overall data annotation processes. Therefore, the initializer reduces computation costs related to data generation considerably. Simulation examples show that using the dataset generated by the proposed method improves the DL-based array selector's accuracy compared to the one generated by the conventional random sampler. Moreover, the realized sparse arrays show better sparse array configuration characteristics and enhanced DOA estimation performance.

**INDEX TERMS** Antenna selection, direction-of-arrival estimation, deep learning, simulated annealing, sparse arrays.

## I. INTRODUCTION

The design of optimum sparse arrays via machine learning (ML) for direction-of-arrival (DOA) and beamforming (BF) has recently received tremendous attention [1]–[15]. This is the case as ML-based arrays design approaches have low computation complexity compared to their combinatorial and convex optimization counterparts [6]–[9]. For instance, [10] proposed a support vector machine (SVM) in connection with an artificial neural network (ANN) to predict sparse linear array configurations for adaptive beamforming. The approach employs sensing environmental features extracted from capon-beamformer to select the array that maximizes the adaptive beamforming signal-to-interference-plus-noise ratio (SINR). Furthermore, a deep learning-based antenna selection approach was proposed in [11] to predict planar or two-dimensional (2D) sparse arrays using covariance matrix as input.

Structurally, the ML-based selectors are designed so that once the optimal sensors are selected, any DOA estimation

technique can be applied for DOA estimation. Typical methods include subspace methods, such as MUSIC [12], and compressive sensing methods, such as orthogonal matching pursuit (OMP) [13], [14]. As such, further application of [11] to compressive sensing techniques and multiple sources scenarios was presented in [11]. Furthermore, transfer learning over the source and the geometry domains were also evaluated over the exact formulation of [10] in [12]. Finally, [17], [18] presents the application of the technique to linear arrays and performance analysis of the same under various DOA estimation situations.

A closer comparison of [11], [15]–[19] shows that these DL-based methods were proposed based on the benchmark work in [11]. Since the main limitation of the benchmark work in [11] is the use of enumeration and exhaustive search algorithms to generate and label the dataset, then [15]–[19] exhibit the same limitation. Typically, antenna selection problems involving 2D arrays yield combinatorial solutions of the order of  $10^6$ , and higher [18]. As such, enumeration of all combinations and annotating such a large solution set is costly and often limits the size of the 2D array per problem [16]. To overcome the limitations above, in [11],

The associate editor coordinating the review of this manuscript and approving it for publication was Hasan S. Mir.

[15], a random sampling approach was suggested to replace the enumeration method. The technique involves random enumeration of a portion of all combinations instead of using all the possible combinations. However, the method often leads to sub-optimal solutions as the randomly sampled batches may not statistically represent the distribution of subarrays in the actual combination set [19].

Notwithstanding the limitation above, the ML-based optimization approaches are still desirable for a wide range of applications in wireless communication, aside from the design of sparse arrays. For example, the ML-based techniques are becoming instrumental in resource allocation in the unmanned aerial vehicle (UAV) [4], [5], beam selection, and management in MIMO systems. Such is the case because once trained; the model requires less matrix multiplication time to converge to a near-optimal solution than combinatorial or convex optimization-based algorithms [6]–[9]. Moreover, the ML-based techniques are robust to uncertainties, and they can transfer features between two models or tasks [16], [20].

On the other hand, most conventional antenna selection techniques employ heuristic or population-based optimization methods. Although practical, they are computationally expensive and prone to local minima [6]. However, they have been proven effective when dealing with a small, well-defined set of problems. The commonly used methods include simulated annealing (SA) and genetic algorithm (GA) [21]. Therefore, to circumvent the use of enumeration of all possible combinations, we propose an SA-based algorithm to assist in the generation of the training dataset. Although, in specific settings, GA can outperform the SA algorithm. In this work, we opted for SA since, unlike the GA algorithm, it starts with only one solution and tries to enhance it. Hence, it simplifies the selection of instances at the outset [21].

This paper presents simulated annealing assisted DL-based antenna selection approach for 2D sparse array selection. The proposed technique is a hybrid two-stage approach for the generation of the dataset, and sparse array selection using features extracted from DOA estimation environment [19]. The first stage involves the generation of training samples where the  $M$ -element 2D subarrays associated with the antenna selection problem are sequentially and randomly sampled. Then, the sampled 2D subarrays are optimized to spread the sensors while maintaining the maximum aperture. The second stage consists of training data labeling and sparse array selection processes. The stage employs a simple search algorithm to sift through the subarrays rendered in stage one for best subarrays to designate class labels or ground truth. Then, the realized dataset is used to train a convolutional neural network (CNN) model for sparse subarray multi-classification purposes [11]. Numerical results show that the proposed method improves the DL-based antenna selector's accuracy and reduces computation costs related to training data generation and annotation. Furthermore, the results show that the rendered 2D sparse arrays have improved DOA estimation resolution compared to the parent 2D array and other 2D sparse arrays.

In general, the main contributions of this paper are summarized as follows.

- i) We proposed a deep-learning-based antenna selection approach with a simulated annealing initializer. The proposed hybrid approach has improved sparse array estimation accuracy and reduced computation complexity compared to the conventional deep learning-based methods and traditional SA-based antenna optimization approach.
- ii) Furthermore, detailed theoretical and numerical simulation results are presented to demonstrate the superiority of the proposed antenna selection method in terms of realized antenna characteristics, DOA estimation performance, and computation complexity.

The remainder of this paper is outlined as follows. In section II, we consider preliminaries of antenna selection and corresponding conventional dataset generation approach. The proposed simulated annealing-based antenna selection technique is discussed in section III. Then, numerical simulation experiments are carried out to test the advantages of the proposed antenna selection approach under various scenarios in section IV. Finally, section V concludes the paper.

Throughout the paper, we use lower-case and upper-case bold characters to denote vectors and matrices, respectively, i.e.,  $\mathbf{I}_K$  represents the  $K \times K$  identity matrix. Operators  $(\cdot)^T$  and  $(\cdot)^H$  stand for transpose and the conjugate transpose of a vector or matrix in that order. And,  $\text{vec}(\cdot)$  denotes vectorization operator and  $\text{diag}(\cdot)$  represents a diagonal matrix. Moreover,  $\odot$  and  $E[\cdot]$  denote the Khatri-Rao product and statistical expectation operator.

## II. PRELIMINARIES

In this section, we briefly introduce the conventional deep learning-based antenna selection approach.

### A. PROBLEM FORMULATION

The problem of selecting a subarray with  $M$ -elements from a uniform array with  $N$ -elements yields  $Q$  possible combinations where

$$Q = \binom{N}{M} = \frac{N!}{M!(N-M)!}. \quad (1)$$

From a machine learning perspective, (1) is considered as the number of possible classes. Assuming that  $\mathcal{H}$  contains all the possible classes in (1), and that  $h \in \mathcal{H}$  is a sparse subarray associated with sensor position set  $\mathcal{Z}_h = [z_1, z_2, z_3, \dots, z_M]$ . Then, all the sparse subarrays in  $\mathcal{H}$  can be expressed as

$$\mathcal{H} = [\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3, \dots, \mathcal{Z}_Q]. \quad (2)$$

Following (2), [11] proposed a convolutional neural network (CNN) model to classify the best sparse subarray configuration that offers better DOA estimation performance. Since CNN is a data-driven approach, the core component of the method in [11] is the well-labeled dataset. Therefore, per [11] the annotated dataset can be generated from  $\mathcal{H}$

using a two-step process. Firstly, a set of target DOA are sampled, and corresponding received signal realizations for all  $h \in \mathcal{H}$  are generated. Secondly, using a performance metric, the subarray configurations with the best performance values are designated as labels [11], [13]–[15]. We review the procedures above in the subsequent sections.

## B. REALIZATION OF TRAINING DATASET

In this section, we briefly review the procedures for generating and annotating the training dataset. In general, the process involves the following key steps:

### 1) REALIZATION OF SUBARRAY CONFIGURATIONS

The first step involves realization of sample subarray configurations according to (1) and (2).

### 2) REALIZATION OF RECEIVED SIGNALS

Next, the received signals for each sample subarray realized in step I are calculated. Therefore, suppose that  $K$  uncorrelated narrowband sources from directions  $\Theta_1, \Theta_2, \dots, \Theta_K$  are impinging on  $h$ -th subarray with  $M$ -elements where  $\Theta_k = (\theta_k, \phi_k)$  such that  $\theta_k$  and  $\phi_k$  are  $k$ -th elevation and azimuth angles for  $k = 1, 2, \dots, K$ . Then, the received signal at  $h$ -th subarray can be defined as

$$\mathbf{x}_h(t) = \mathbf{A}_h(\Theta)\mathbf{s}_h(t) + \mathbf{n}_h(t), \quad (3)$$

where  $\mathbf{A}_h(\Theta)$ ,  $\mathbf{s}_h(t)$  and  $\mathbf{n}_h(t)$  denote the  $h$ -th subarray manifold, received signal vector and noise vector respectively. Assuming that the source and the noise vectors are statistically independent and uncorrelated [17]–[19], the corresponding covariance matrix can be expressed as

$$\begin{aligned} \mathbf{R}_h &= E[\mathbf{x}_h(t)\mathbf{x}_h^H(t)] \\ &= \mathbf{A}_h(\Theta)\mathbf{R}_s\mathbf{A}_h^H(\Theta) + \sigma_n^2\mathbf{I}_M, \end{aligned} \quad (4)$$

where  $\mathbf{R}_s = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2)$ ,  $\sigma_k^2$  and  $\sigma_n^2$  are signal and noise powers respectively.

### 3) COMPUTATION OF PERFORMANCE METRIC VALUES

Here, the performance of each subarray is determined using a specific performance metric. Like [11], this work assumes CRB as a performance metric. Therefore, from (4), the partial derivative of  $\mathbf{A}_h(\Theta)$  with respect to  $\theta$  and  $\phi$  can be expressed as  $\mathbf{b}(\theta_i) = \frac{\partial}{\partial \theta_i}\mathbf{A}_h(\theta_i)$  and  $\mathbf{b}(\phi_i) = \frac{\partial}{\partial \phi_i}\mathbf{A}_h(\phi_i)$  respectively for  $i = 1, 2, \dots, K$  such that  $\mathbf{B}_\theta = [\mathbf{b}(\theta_1), \mathbf{b}(\theta_2), \dots, \mathbf{b}(\theta_D)]$  and  $\mathbf{B}_\phi = [\mathbf{b}(\phi_1), \mathbf{b}(\phi_2), \dots, \mathbf{b}(\phi_D)]$  respectively. As such, the CRB of the  $h$ -th subarray with respect to  $\theta$  and  $\phi$  can be defined as

$$\mathcal{C}_{\theta,h} = \frac{\sigma_n^2}{2T} \Re \left[ \mathcal{I}_\theta \odot (\mathbf{R}_s \mathbf{A}_h^H(\Theta) \mathbf{R}_h^{-1} \mathbf{A}_h(\Theta) \mathbf{R}_s)^T \right], \quad (5)$$

$$\mathcal{C}_{\phi,h} = \frac{\sigma_n^2}{2T} \Re \left[ \mathcal{I}_\phi \odot (\mathbf{R}_s \mathbf{A}_h^H(\Theta) \mathbf{R}_h^{-1} \mathbf{A}_h(\Theta) \mathbf{R}_s)^T \right], \quad (6)$$

where

$$\begin{aligned} \mathcal{I}_\theta &= \mathbf{B}_\theta^H \left[ \mathbf{I} - \mathbf{A}_h(\Theta)(\mathbf{A}_h^H(\Theta)\mathbf{A}_h(\Theta))^{-1}\mathbf{A}_h^H(\Theta) \right] \mathbf{B}_\theta, \\ \mathcal{I}_\phi &= \mathbf{B}_\phi^H \left[ \mathbf{I} - \mathbf{A}_h(\Theta)(\mathbf{A}_h^H(\Theta)\mathbf{A}_h(\Theta))^{-1}\mathbf{A}_h^H(\Theta) \right] \mathbf{B}_\phi, \end{aligned}$$

in that order [10], [17]. Hence, the CRB of the  $h$ -th subarray can be expressed as

$$\mathcal{C}(\Theta, \mathcal{Z}_h) = \frac{1}{\sqrt{2}} \left[ \mathcal{C}_{\theta,h}^2 + \mathcal{C}_{\phi,h}^2 \right]^{\frac{1}{2}}, \quad (7)$$

assuming that the signal-to-noise ratio (SNR) is defined as  $10 \log_{10}(\sigma_s^2/\sigma_n^2)$ . Thus, using (7) one can determine the CRB of DOA estimation of any  $h \in \mathcal{H}$  given information related to the DOAs of signals [19].

### 4) SELECTION OF THE BEST SUBARRAYS (OR LABELS)

Lastly, we comb through the calculated CRB values and select the subarrays with the best performance. Therefore, through a simple search method, the best subarrays which minimize (7) are singled out as labels. Thus, set  $\mathcal{U}$  is constructed which consists of subarrays that minimize the following problem

$$u = \arg \min_{h=1,2,3,\dots,Q} \mathcal{C}(\Theta, \mathcal{Z}_h), \quad (8)$$

for  $u = 1, 2, 3, \dots, |\mathcal{U}|$ . As observed in [11], [19], the set  $\mathcal{U}$  is much smaller than  $\mathcal{H}$  due to the similarities in array configurations and responses to various DOAs.

Note that the input data is  $N \times N \times 3$  real-valued matrices  $\{\mathbf{H}\}_{i=1}^3$  whose  $(i, j)$ -th entry consists of  $[\mathbf{H}_1]_{i,j} = \angle[\hat{\mathbf{R}}]_{i,j}$ ,  $[\mathbf{H}_2]_{i,j} = \Re[\hat{\mathbf{R}}]_{i,j}$  and  $[\mathbf{H}_3]_{i,j} = \Im[\hat{\mathbf{R}}]_{i,j}$  denoting the phase, real and imaginary components of a sample covariance matrix  $\hat{\mathbf{R}}$  [10]. Hence, the input-output data pairs are computed as  $(\mathbf{H}, u)$  where  $u \in \mathcal{U}$  is the output label denoting the best subarray sensor indices given  $\hat{\mathbf{R}}$  as input [11], [19].

As mentioned in the introductory section, the enumeration of  $\mathcal{H}$  coupled with optimization task in (8) is computationally expensive [11] bearing the fact that  $|\mathcal{U}| \ll |\mathcal{H}|$ . Therefore, in the subsequent section, we introduce an alternative approach to realizing sample subarray configurations and the labels. This approach requires less expensive computation costs than the use of  $\mathcal{H}$ .

## III. PROPOSED SIMULATED ANNEALING BASED TRAINING DATA GENERATION APPROACH

In this section, we introduce a simulated annealing-based algorithm as an initialization step to the DL-based selector. The primary purpose of the algorithm is to generate the training dataset instead of enumeration of the whole  $\mathcal{H}$  set as in (1) or a partition of  $\mathcal{H}$  as in [11].

### A. PROPOSED SA-BASED APPROACH

Simulated annealing has been used countless times to design optimum arrays [21]. The approach was used in [19] to optimize sensor positions of the hourglass array while minimizing the mutual coupling between the sensors. Thus, given an initial array as  $\mathcal{Z}_{init}$ , the optimized array  $\mathcal{Z}_{SA}$  can be obtained by minimizing

$$\mathcal{Z}_{SA} = \arg \min_{\mathcal{Z}_{init}} k_o = \sum_{i=1}^M \sum_{j=i+1}^M \frac{1}{\|m_i - m_j\|_2},$$

$$\text{such that } \|m_i - m_j\|_2 \leq B, \quad (9)$$

where  $m_i, m_j \in \mathcal{Z}_{init}$ ,  $M$  is the number of sensors,  $\|\cdot\|_2$  is the  $l_2$ -norm of a vector and  $B$  is the mutual coupling coefficient upper-bound. Note that, apart from the distribution of sensor positions for a 2D sparse arrays, the SA-based optimization method can be turned to operate under the fixed physical aperture [19], [21].

This work utilizes the SA algorithm as an initialization step for the deep learning model. Mainly, the SA algorithm is used to generate sparse subarrays with large physical apertures and well-distributed sensors instead of an enumeration approach. Specifically, the SA optimization stage follows the steps below:

- (a) Firstly, an initial random  $M$ -element 2D sparse subarray is generated out of a full  $N$ -element 2D array i.e  $\mathcal{Z}_{init}$ .
- (b) Then, at each iteration it perturbrates the  $\mathcal{Z}_{init}$  while maintaining the corner sensors, i.e.,  $M \setminus \mathcal{Z}_\psi$ . Here,  $\mathcal{Z}_\psi$  defined as

$$\mathcal{Z}_\psi = [(0, 0), (0, M_y - 1), (M_x - 1, 0), \dots, (M_x - 1, M_y - 1)] \quad (10)$$

contains all corner sensors. Also, the number of permitted missing virtual sensors  $\eta$  in the difference coarray (DCA) of  $\mathcal{Z}_{init}$  is set zero [19]. Using the current temperature  $\beta$ , we defined the acceptance probability function  $\rho(\Delta\kappa, \beta)$  as

$$\rho(\Delta\kappa, \beta) = \begin{cases} e^{-\Delta\kappa/\beta}, & \text{if } \Delta\kappa > 0 \\ 1, & \text{otherwise} \end{cases} \quad (11)$$

such that  $\Delta\kappa = \kappa_n - \kappa_{n-1}$  where  $\kappa_n$  and  $\kappa_{n-1}$  denote objective functions of the new and the previous solutions. Thus, if the new  $\kappa$  is smaller than the preceding one, the solution is accepted.

Finally, at the end of the iteration the temperature is decreased to block poor solutions from being accepted [21]. This is done using a cooling schedule which is determined by a factor  $\alpha$ . Therefore, using  $\beta_o$ , at  $i$ -th iteration the temperature becomes

$$\beta = \beta_o \cdot \alpha^i. \quad (12)$$

And, the temperature is reduced at each iteration until the algorithm converges. Note that the smaller the  $\Delta\kappa$ , the higher the temperature. Furthermore, by cooling down the temperature slowly, it slows down the convergence rate. Therefore, it is essential to select a higher value of  $\beta_o$  to escapes from the local minimum and an optimal value of  $\alpha$  to increase chances of obtaining a global optimum [6], [19], [21].

- (c) Lastly, steps (a)–(b) are repeated until a reasonable number of subarrays are realized to construct a solution set  $\mathcal{H}_{sa}$ .

Figure 1 summarizes the above steps into a generalized flow diagram. Note that this SA-based approach can be extended to any planar array configuration, and the same applies to the whole SA-based initialization method [19], [21]. Following labeling approach in step (4) of section (II-B), we select the best subarrays from  $\mathcal{H}_{sa}$  with the lowest CRB values as labels.

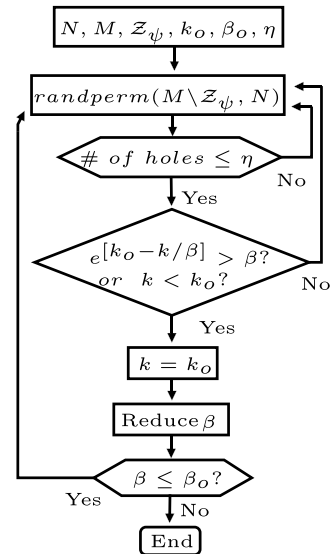


FIGURE 1. Improved simulated annealing algorithm for random 2D sparse subarray generation and optimization.

The above steps are summarized in Algorithm 1. In Algorithm 1, the inputs are as follows: the total number of given antennas  $N$ , the number of antennas to be selected  $M$ , the number of snapshots  $T$ , the number of different DOA angles  $K$ , the number of signals and noise realizations  $P$  and the SNR. Moreover, the elements of  $\mathcal{V}$  in step 5 are selected from  $\mathcal{H}_{sa}$  which is calculated using the proposed SA-based optimization method as shown in Fig. 1 rather than the enumeration of the entire combinations  $\mathcal{H}$ . The SNR used for calculation of the covariance matrices in step 4 is denoted as  $\text{SNR}_{\text{TRAIN}}$ .

**Algorithm 1** Proposed Training Dataset Generation Method

**Input:** Given  $N, M, T, K, P$  and  $\text{SNR}_{\text{TRAIN}}$

**Output:** Training data  $\mathcal{D}_{\text{TRAIN}}$

- 1: Generate  $\mathcal{H}_{sa}$  as shown in Fig. 1.
- 2: Sample  $K$  DoA angles  $\{\Theta_k\}_{k=1}^K$ .
- 3: Compute  $P$  different realizations of subarray output,  $\{X_k^i\}_{i=1}^P$  for  $k = 1, \dots, K$   
 $X_k^i = [x_k^i(1), x_k^i(2), \dots, x_k^i(T)]$ ,  
 where  $x_k^i(t) = \mathbf{a}(k)\mathbf{s}^{(i)}(t) + \mathbf{n}^{(i)}(t)$ ,  
 $\mathbf{s}^{(i)}(t) \sim \mathcal{CN}(0, \sigma_s^2 \mathbf{I})$  and  $\mathbf{n}^{(i)}(t) \sim \mathcal{CN}(0, \sigma_n^2)$
- 4: Calculate sample covariance matrix  $\hat{\mathbf{R}}$  and  $M \times M$  covariance matrices  $R_{h_s}^{(i,k)}$  for  $h_s = 1, 2, \dots, \bar{\mathcal{H}}_{sa}$ .
- 5: Compute  $\mathcal{C}(\Theta, \mathcal{Z}_{h_s})$  for all  $h_s \in \mathcal{H}_{sa}$  and select subarrays following (8).
- 6: Create input-output data pairs as  $(\hat{\mathbf{R}}^{(i,k)}, u_k^{(i)})$  for  $k = 1, 2, \dots, K$  and for  $i = 1, 2, \dots, P$ .
- 7: Connect the input-output pairs to form the training dataset as

$$\mathcal{D}_{\text{TRAIN}} = \left[ (\hat{\mathbf{R}}^{(1,1)}, u_1^{(1)}), (\hat{\mathbf{R}}^{(2,1)}, u_1^{(2)}), \dots, (\hat{\mathbf{R}}^{(P,1)}, u_1^{(P)}), (\hat{\mathbf{R}}^{(1,2)}, u_2^{(1)}), \dots, (\hat{\mathbf{R}}^{(P,K)}, u_K^{(P)}) \right]$$

where the size of the training dataset is  $\mathcal{R} = PK$ .

## B. APPLICATION TO 2D DOA ESTIMATION

Following the selection of 2D sparse arrays, the received signal on the selected sparse array can be expressed as

$$\bar{\mathbf{x}}(t) = \bar{\mathbf{A}}(\Theta)\mathbf{s}(t) + \bar{\mathbf{n}}(t), \quad (13)$$

such that  $\bar{\mathbf{A}}(\Theta)$  is the sub-matrix of a  $N \times N$  covariance matrix  $\mathbf{A}(\Theta)$  whose entries are constructed based on the sensor locations in the selected sparse array [11], [19]. Then, a new covariance matrix is computed as

$$\begin{aligned} \mathbf{R}_{\bar{\mathbf{x}}} &= E[\bar{\mathbf{x}}(t)\bar{\mathbf{x}}^H(t)] \\ &= \bar{\mathbf{A}}(\Theta)\mathbf{R}_s\bar{\mathbf{A}}^H(\Theta) + \sigma_n^2\mathbf{I}_M. \end{aligned} \quad (14)$$

By vectorizing (14), we obtain the following coarray model

$$\mathbf{z} = \mathbf{A}_c\mathbf{p} + \sigma_n^2\mathbf{r}, \quad (15)$$

where  $\mathbf{A}_c = \bar{\mathbf{A}}^* \odot \bar{\mathbf{A}}$ ,  $\mathbf{p} = [p_1, p_2, \dots, p_K]^T$  and  $\mathbf{r} = \text{vec}(\mathbf{I}_M)$  [20]. Here,  $\mathbf{A}_c$  is the extended array manifold whose sensor locations are given by a difference coarray (DCA) which can be expressed as  $\mathcal{Z}_{co} = \{m_1 - m_2 | m_1, m_2 \in \mathcal{Z}\}$ . By carefully deleting the repeated rows,  $\mathbf{A}_c$  yields a new array manifold representing a virtual uniform rectangular array (URA) structure with enhanced degrees of freedom [21]–[24]. Therefore, if the signal model in (13) is used directly, the proposed method can estimate  $K < M$  sources only. However, if the coarray signal model as in (15) is exploited, the proposed method can estimate more sources than the number of sensors, i.e.,  $K > M$  [21], [23].

## IV. NUMERICAL EXAMPLES

This section examines the performance of the proposed SA-assisted DL-based antenna selection approach for 2D sparse array selection. We train and test the CNN model using the data generated by the conventional method [19] and the proposed method. The problem of selecting 16 sensors ( $M$ ) out of a 42-sensor URA is considered throughout the section. Later, we evaluate the realized enhanced DL-based 2D sparse array in terms of array structure characteristics and root-mean-square-error (RMSE) of 2D DOA estimation.

In the subsequent paragraphs, we employ subscripts  $(\cdot)_{\text{TRAIN}}$  and  $(\cdot)_{\text{TEST}}$  to indicate parameters used for training and testing modes, respectively. Moreover, based on [11] the training data is obtained by sampling the DOA space with  $K_{\theta, \phi}$  directions whereas the DOAs in the test data are randomly selected. Also, for performance comparison purposes we consider (a) the conventional DL-based 2D sparse array [11], (b) a 16-element SA-optimized sparse array [21], (c) a 16-element URA and (d) the original 42-element URA [1]. Note that all the 2D sparse arrays consist of  $M = 16$  sensors, and the same aperture except for the parent 42-element URA.

### A. CNN ARCHITECTURE

For objective comparison, we adopt a general CNN structure consisting of 8 layers as in [11]. In general terms, the first layer (1<sup>st</sup> layer) accepts the 2D input and the last

TABLE 1. Realized training data for various techniques.

$(N, M)$	Sets	Data Generation Techniques	
		Conventional ( $\mathcal{H}$ )	Proposed ( $\mathcal{H}_{sa}$ )
(42, 16)	Samples	10000	4000
	Labels	32	37

output layer (8<sup>th</sup> layer) is a classification layer with  $h$  units where a softmax function is used to obtain the probability distribution of the classes [19]. The second (2<sup>nd</sup> layer) and the fourth (4<sup>th</sup> layer) layers are max-pooling layers with  $2 \times 2$  kernel to reduce the dimension whereas the third (3<sup>rd</sup> layer) and the fifth (5<sup>th</sup> layer) layers are convolutional layers with 64 filters of size  $2 \times 2$ .

Finally, the seventh (7<sup>th</sup> layer) and the eighth (8<sup>th</sup> layer) layers are fully connected layers with 1024 units. Note, the rectified linear units (ReLU) are used after each convolutional and fully connected layers such that  $\text{ReLU}(x) = \max(x, 0)$  [11]. During the training phase: 90 % and 10 % of the data are allocated for training and validation purposes. The stochastic gradient descent with momentum (SGD) is used with a learning rate of 0.03, and a mini-batch of 500 for 50 epochs [20].

### B. TRAINING DATA GENERATION

In this section, using a URA with 42 sensors, we generate two distinct training datasets using the conventional method [11] and the proposed method. For the former, we randomly sample 10000 subarrays, whereas, for the latter, we employ the proposed SA-based optimization method. To realize classes  $\mathcal{H}_{sa}$  using proposed SA-based algorithm, we assume the following parameters:  $\eta = 0$ ,  $\beta = 1000$ ,  $\beta_o = 0.0001$  and  $\kappa_o$  is as in (9). We sample  $K_\phi = 120$  DOAs uniformly within the range of  $[0^\circ, 360^\circ)$  whereas  $K_\theta$  was fixed at  $90^\circ$ . Furthermore, for each dataset, we assume 10 dB SNR<sub>TRAIN</sub> and 100 snapshots.

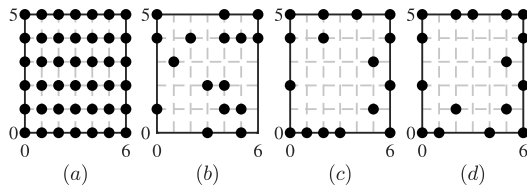
Table 1 shows the realized data samples and their corresponding labels. As shown in Table 1, the number of labels generated by the proposed method almost tally with those realized using the conventional method despite using a small number of samples as compared to the conventional approach. Thus, the proposed initialization step achieves large label samples from a small predefined generated dataset compared to the conventional method. Thus, the proposed approach considerably reduces computation costs.

### C. ESTIMATION PERFORMANCE AND ACCURACY

This section evaluates the estimation accuracy of the proposed method compared to the conventional approach using the CNN models trained using the datasets generated in the previous section. In the first example, we test the CNN model with the data generated using parameters as shown in the second column of Table 2 to predict sparse 2D arrays. Figure 2 shows the array configuration of predicted 2D sparse arrays. Particularly, Fig. 2 (a) illustrates the parent 42-element URA whereas Fig. 2 (b)–(c) show the realized

**TABLE 2.** Simulation parameters for estimation performance evaluation.

Parameters	Simulation Examples		
	Example #1	Example #2	Example #3
$(N, M)$	(42, 16)		
# of trials, $\mathcal{T}$	100		
$T_{\text{TEST}}, T_{\text{TRAIN}}$	100 snapshots		
$(K_\theta, K_\phi)_{\text{TRAIN}}$ [deg.]	120		
$(K_\theta, K_\phi)_{\text{TEST}}$ [deg.]	2		
SNR <sub>TRAIN</sub> [dB]	10		
SNR <sub>TEST</sub> [dB]	5	-20 to 10	



**FIGURE 2.** The array configurations of (a) parent 42–sensor URA, (b) the conventional DL-based array (16–sensors), (c) the conventional SA-based array (16–sensors) and (d) the proposed DL-based array (16–sensors). Note that the dots denote physical sensors.

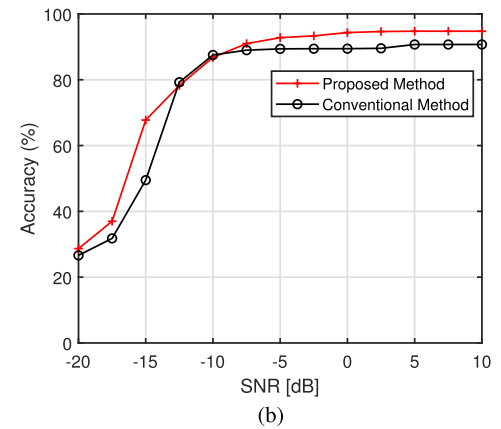
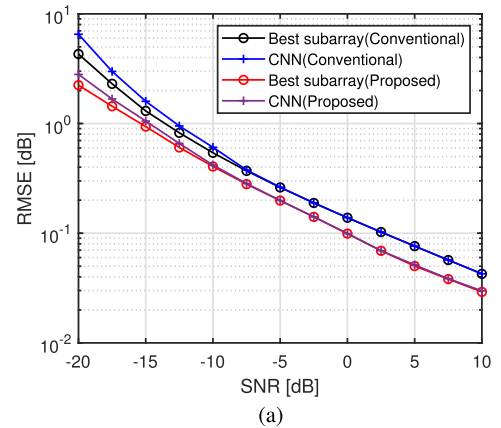
16-element DL-based 2D sparse array using the conventional and the proposed method respectively. It can be observed that the proposed method yields a sparse array with a larger aperture compared to the conventional DL-based sparse array.

In the second example, we evaluate the proposed method’s DOA estimation performance compared to the conventional method using parameters as shown in the third column of Table 2. The realized sparse arrays from CNN are fed to a MULTiple Signal Classification (MUSIC) algorithm [11] for DOA estimation. In this case, the SNR<sub>TEST</sub> is varied from -20 dB to 10 dB over 100 number of trials. Figure 3 (a) shows the RMSE of DOA estimation as function of SNR<sub>TEST</sub>. Namely,

$$\text{RMSE} = \sqrt{\frac{1}{\mathcal{T}K} \sum_{i=1}^{\mathcal{T}} \sum_{k=1}^K (\tilde{\phi}_k^i - \phi_k)^2}, \quad (16)$$

where  $\tilde{\phi}_k^i$  and  $\phi_k$  denote the estimated and true  $k$ -th DOAs in the  $i$ -th trial, respectively. Note that the best subarray as indicated Fig. 3 (a) represents the subarray with the lowest CRLB value or the label. As a result, we compare the best subarray’s DOA estimation performance with the predicted arrays, i.e., CNN generated sparse arrays by both the conventional and the proposed method. Moreover, it can be observed that the proposed method follows and converges quickly with the best subarray performance compared to the conventional method.

In the third example, we evaluate the proposed method’s sensor selection accuracy compared to the conventional method using parameters as shown in the fourth column of Table 2. The realized sparse arrays from CNN are compared to the best subarrays or labels to evaluate the classification performance [12]. Similarly, during the testing stage the SNR<sub>TEST</sub> is varied from -20 dB to 10 dB over 100 number of trials. Figure 3 (b) shows the accuracy of sensor selection



**FIGURE 3.** Comparison of (a) DOA estimation of the conventional and proposed method for different array configurations, and (b) performance of the conventional and proposed method for different SNR<sub>TEST</sub> values.

as a function of SNR<sub>TEST</sub>, i.e.,

$$\text{Accuracy} = \frac{\mathcal{D}}{\mathcal{F}} \times 100 \%, \quad (17)$$

where  $\mathcal{D}$  is the total number of input data in which the model identifies the best subarray correctly  $\mathcal{F}$  times [16]. As a result, we observed that the proposed method has more than 90% accurate for SNR<sub>TEST</sub>  $\geq$  -8 dB when the network is trained by the dataset with SNR<sub>TRAIN</sub>  $\geq$  10 dB. Compared to the conventional method, it has less than 90% accurate for SNR<sub>TEST</sub>  $\geq$  -8 dB when the model is trained with the same parameters. Hence, the proposed method shows improved performance compared to the conventional method.

#### D. 2D DOA ESTIMATION PERFORMANCE

In this section, we evaluate the performance of the realized 2D sparse array in section IV-B using the proposed method in comparison to the parent 42–sensor URA and other 2D sparse arrays. In particular, we explore the behavior of the RMSE as a function of SNR and the number of snapshots. Here, the RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{\mathcal{T}K} \sum_{i=1}^{\mathcal{T}} \sum_{k=1}^K [(\tilde{\phi}_k^i - \phi_k) + (\tilde{\theta}_k^i - \theta_k)]^2}, \quad (18)$$

TABLE 3. 2D DOA estimation simulation parameters.

Parameters	Simulation Examples	
	Example #4	Example #5
# of sources $K$	9	
# of trials, $\mathcal{T}$	1000	
# of snapshots	500	10 to 1000
SNR [dB]	-20 to 10	0
DOAs $(\theta, \phi)$ [deg.]	(10, 255), (45, 300), (20, 345), (50, 210), (70, 0), (0, 30), (60, 165), (5, 120), (30, 79)	
Estimator	2D-ESPRIT	

where  $\tilde{\phi}_k^i, \tilde{\theta}_k^i$  and  $\phi_k, \theta_k$  denote the estimated and true  $k$ -th DOAs in the  $i$ -th trial, respectively.

Table 3 lists the parameters used to compute RMSE with respect to SNR (Example #4) and the number of snapshots (Example #5), respectively. Moreover, the 2D-ESPRIT algorithm is used to estimate the sources [19], [24]. However, if the DCA of the realized 2D arrays has holes in the coarray, the resulting virtual 2D array becomes irregular. Therefore, the spatial-smoothing DOA estimation method such as 2D-ESPRIT cannot be applied. This is the case as the 2D-ESPRIT algorithm requires a URA array structure for spatial smoothing pre-processes [24]. As a result, a nuclear norm minimization (NNM) approach is applied to fill the holes [21] to restore a standard 2D configuration.

Figure 4 shows the DOA estimation performance of the realized DL-based 2D sparse array compared to the parent URA and other 2D sparse arrays. It can be observed in Fig. 4(a) that the URA with 42 sensors has better performance overall due to the large physical aperture. In contrast, the performance of the DL-based 2D sparse array realized using the proposed method has better than a URA with 16 sensors and slightly lower than that of the parent URA. Besides, the proposed 2D sparse array performs better than the conventional SA-based sparse array. Moreover, the conventional DL-based performed poorly as compared to both the proposed array and 16-sensor URA.

In Fig. 4(b), we can observe a similar trend where the performance of the proposed array is bounded by the parent URA and the 16-sensor URA as the number of snapshots increases. The parent URA has the lowest RMSE values, whereas the 16-sensor URA has higher RMSE values than the proposed array. Also, the conventional DL-based array performs better than the 16-sensor URA in lower numbers of snapshots. However, the RMSE values degrade as the number of snapshots increases. The examples demonstrate that the proposed method can thin 2D arrays cognitively to a manageable size without considerable loss of DOA estimation resolution. Moreover, compared to other conventional methods such as SA, the realized sparse arrays exhibit enhanced DOA estimation performance.

E. COMPUTATION COMPLEXITY

The enumeration method used to generate  $\mathcal{H}$  has a computation complexity of  $\mathcal{O}(\frac{N!}{M!(N-M)!}M)$ . On the other hand, the SA-based algorithm requires  $\mathcal{O}(M)$  to generate

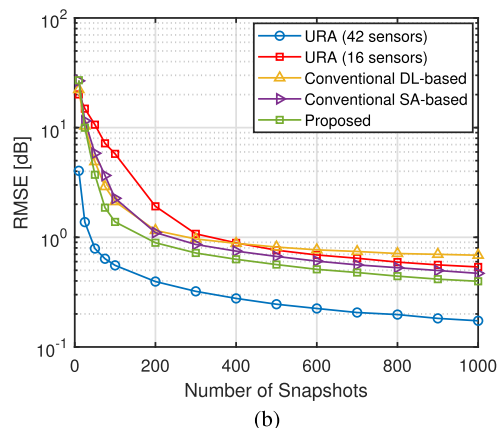
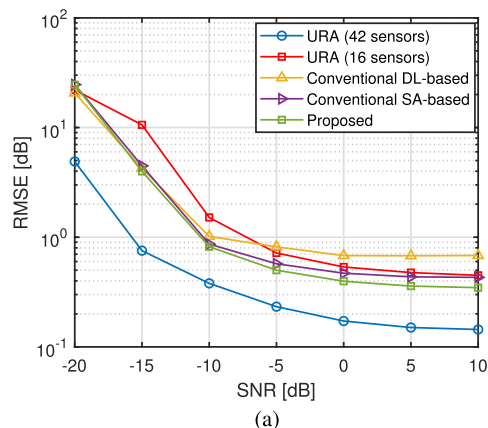


FIGURE 4. RMSE performance versus (a) SNR and (b) number of snapshots for the realized DL-based sparse array compared to URA, conventional DL-based array and SA-based array.

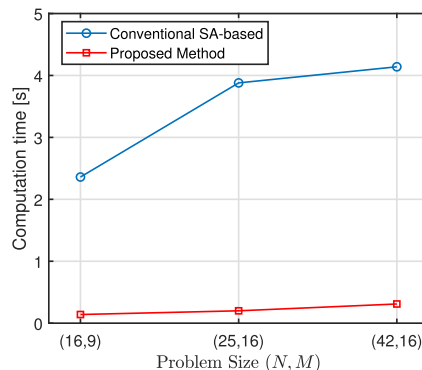


FIGURE 5. A summary of computation time of the DL-based antenna selection approach against the conventional SA-based optimization technique..

$M$ -element array and almost  $\mathcal{O}(MN^2)$  to optimize the randomly generated 2D array. Thus, the computation complexity can be approximated as  $\mathcal{O}(MN^2 + M)$ . Although the SA-based algorithm has to be run several times, i.e.,  $\mathcal{J}$ , to realized a specific number of samples. The product of  $\mathcal{J}$  and  $\mathcal{O}(MN^2 + M)$  leads to  $\mathcal{O}(\mathcal{J}(MN^2 + M))$ , which is still reasonable compared to the computation complexity of the conventional method. Table 4 summarizes the computation complexity of the two methods above [16].

**TABLE 4. A summary of computation complexity of the DL-based antenna selection approach.**

Method	Computation complexity
Conventional	$\mathcal{O}((N!/M!(N-M)!)M)$
Proposed	$\mathcal{O}(\mathcal{J}(MN^2 + M))$

In Fig. 5, we compare the computation time required for the proposed DL-based approach and the SA-based optimization method to estimate a single optimal 2D sparse array per 100 iterations. We run the models in MATLAB using a PC with Intel(R) Core (TM)-i5 at 2.60 GHz with 4GB RAM. As indicated in the table, the proposed method requires less computation time to yield a 2D sparse array given a  $N$ -element 2D parent array than the computation time required for the SA-based optimization technique to optimize an  $M$ -element 2D sparse array [21].

## V. CONCLUSION

The paper presented a novel two-stage DL-based sparse array selection approach. The first stage is an initialization step that uses the SA algorithm to generate sparse array configurations with key target features. As a result, it enables data labels to be realized using few data samples, which reduces computational costs and time. The final dataset is then used to train a CNN model in the second stage. Simulation results indicate that the proposed method yields sparse arrays with enhanced DOA estimation performance. Moreover, the results showed that the proposed approach realizes many labels from small data samples. More importantly, the CNN model requires less computation time to converge at an optimum solution (almost 10 times less) than the conventional SA-based optimization approach upon training and deployment.

## REFERENCES

- [1] T. E. Tuncer and B. Friedlander, *Classical and Modern Direction of Arrival Estimation*. New York, NY, USA: Academic, 2009.
- [2] H. L. Van Trees, *Optimum Array Processing, Detection, Estimation and Modulation: Part IV*. Hoboken, NJ, USA: Wiley, 2002.
- [3] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
- [4] L. Sun, L. Wan, and X. Wang, "Learning-based resource allocation strategy for industrial IoT in UAV-enabled MEC systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5031–5040, Jul. 2021.
- [5] L. Wan, L. Sun, K. Liu, X. Wang, Q. Lin, and T. Zhu, "Autonomous vehicle source enumeration exploiting non-cooperative UAV in software defined internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3603–3615, Jun. 2021.
- [6] G. Doblinger, "Optimized design of interpolated array and sparse array wideband beamformers," in *Proc. 16th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2008, pp. 614–1520.
- [7] V. Roy, S. P. Chepuri, and G. Leus, "Sparsity-enforcing sensor selection for DOA estimation," in *Proc. 5th IEEE Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, Dec. 2013, pp. 340–343.
- [8] J. F. Hopperstad and S. Holm, "The coarray of sparse arrays with minimum sidelobe level," in *Proc. 3rd IEEE Nordic Signal Process. Symp. (NORSIG)*, Jun. 1998, pp. 137–140.
- [9] X. Wang, E. A. Aboutanos, and M. G. Amin, "Adaptive array thinning for enhanced DOA estimation," *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 799–803, Jul. 2015.
- [10] X. Wang, P. Wang, and X. Wang, "Adaptive sparse array reconfiguration based on machine learning algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1159–1163.
- [11] A. M. Elbir, K. V. Mishra, and Y. C. Eldar, "Cognitive radar antenna selection via deep learning," *IET Radar, Sonar Navigat.*, vol. 13, pp. 871–880, Jun. 2019.
- [12] P. Stoica and N. Arye, "MUSIC, maximum likelihood, and Cramér–Rao bound," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 37, no. 5, pp. 720–741, May 1989.
- [13] K. Aghababaiyan, V. Shah-Mansouri, and B. Maham, "High-precision OMP-based direction of arrival estimation scheme for hybrid non-uniform array," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 354–357, Feb. 2020.
- [14] K. Aghababaiyan, R. G. Zefreh, and V. Shah-Mansouri, "3D-OMP and 3D-FOMP algorithms for DOA estimation," *Phys. Commun.*, vol. 31, pp. 87–95, Dec. 2018.
- [15] A. M. Elbir, S. Mulleti, R. Cohen, R. Fu, and Y. C. Eldar, "Deep-sparse array cognitive radar," in *Proc. 13th Int. Conf. Sampling Theory Appl. (SampTA)*, Jul. 2019, pp. 1–5.
- [16] A. M. Elbir and K. V. Mishra, "Sparse array selection across arbitrary sensor geometries with deep transfer learning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 1, pp. 255–264, Mar. 2021.
- [17] S. Wandale and K. Ichige, "On the DOA estimation performance of optimum arrays based on deep learning," in *Proc. IEEE 11th Sensor Array Multichannel Signal Process. Workshop (SAM)*, Jun. 2020, pp. 1–5.
- [18] S. Wandale and K. Ichige, "Design of sparse arrays via deep learning for enhanced DOA estimation," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 1, pp. 1–13, Dec. 2021.
- [19] S. Wandale and K. Ichige, "2D sparse array selection via deep learning," in *Proc. Int. Symp. Antennas Propag. (ISAP)*, Jan. 2021, pp. 445–446.
- [20] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [21] S. Nakamura, S. Iwazaki, and K. Ichige, "An optimum 2D sparse array configuration with reduced mutual coupling," in *Proc. Int. Symp. Antennas Propag. (ISAP)*, 2018, pp. 617–618.
- [22] S. Nakamura, S. Iwazaki, and K. Ichige, "Optimum 2D sparse array and its interpolation via nuclear norm minimization," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.
- [23] S. Nakamura, S. Iwazaki, and K. Ichige, "Optimization and hole interpolation of 2-D sparse arrays for accurate direction-of-arrival estimation," *IEICE Trans. Commun.*, vol. E104.B, no. 4, pp. 401–409, Apr. 2021.
- [24] C.-L. Liu and P. P. Vaidyanathan, "Hourglass arrays and other novel 2-D sparse arrays with reduced mutual coupling," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3369–3383, Jul. 2017.



**STEVEN WANDALE** (Graduate Student Member, IEEE) received the B.Ed. degree in physics from the University of Malawi, Chancellor College, in 2017, and the M.E. degree from Yokohama National University, in 2020, where he is currently pursuing the Ph.D. degree with the Division of Physics, Electrical and Computer Engineering, Graduate School of Engineering. His research interests include array signal processing, wireless communications, and machine learning.



**KOICHI ICHIGE** (Member, IEEE) received the B.E., M.E., and Dr.Eng. degrees in electronics and computer engineering from the University of Tsukuba, Japan, in 1994, 1996, and 1999, respectively. He became a Research Associate with the Department of Electrical and Computer Engineering, Yokohama National University, Japan, in 1999, where he is currently a Professor. He was a Visiting Researcher with the Swiss Federal Institute of Technology Lausanne (EPFL),

Switzerland, from 2001 to 2002. His research interests include digital signal processing, approximation theory, and their applications to image processing and mobile communication. He served as an Associate Editor of IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, from 2004 to 2008, an Associate Editor of *Journal of Circuits, Systems and Computers* (JCSC), from 2012 to 2014, an Associate Editor of *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* (IEICE-EA), from 2015 to 2018, and an Editor of *IEICE-EA*, from 2019 to 2021. He is a member of IEICE.

...