Deep-LaRGE: Higher-Order SVD and Deep Learning for Model Order Selection in MIMO OFDM Systems

Brenda Vilas Boas[1,2], *Wolfgang Zirwas*[1]
[1]Nokia, Germany
brenda.vilas_boas@nokia.com, wolfgang.zirwas@nokia.com

Martin Haardt[2]
[2]Ilmenau University of Technology, Germany
martin.haardt@tu-ilmenau.de

***Abstract*—Despite the large volume of research on the field of model order selection, finding the correct rank number can still be challenging. Propagation environments with many scatters may generate channel multipath components (MPCs) which are closely spaced. This clustering of MPCs in addition to noise makes the model order selection task difficult for wireless channels which can directly impact user equipment (UE) throughput, e.g., wrong lower rank approximation for channel estimation via Unitary ESPRIT. In this paper, we exploit the multidimensional characteristics of MIMO orthogonal frequency division multiplexing (OFDM) systems and propose an artificial intelligence and machine learning (AI/ML) method capable of determining the number of MPCs with a higher accuracy than state of the art methods in almost coherent scenarios. Moreover, our results show that our proposed AI/ML method has an enhanced reliability as the threshold for signal singular value selection is 80 %.**

***Index Terms*—Source number detection, MIMO, deep learning, pattern recognition.**

I. Introduction

Model order estimation has been an active research field since four decades, especially due to its importance for parameter estimation algorithms and difficulty in coherent scenarios, when there is a high correlation between the signal sources. Many algorithms for estimating the model order rely on information theoretic criteria (ITC), such as the Akaike information criterion (AIC) [1], and the minimum description length (MDL) [2]. However, those methods often fail when the number of measurements is limited.

The exponential fitting test (EFT) [3] was proposed to overcome this problem by looking into the gap between the signal and the noise eigenvalues. The EFT finds a theoretical profile of the noise eigenvalues and recursively tests whether there is a mismatch on the observed eigenvalue and the theoretical one. A mismatch greater than a threshold indicates the presence of a source. Moreover, the EFT as well as AIC and MDL were extended to their higher dimensional version in [4]. Recently, the linear regression of global eigenvalues (LaRGE) method [5] has proposed to use the higher-order singular values computed by the higher-order singular value decomposition (HOSVD) [6] to construct the global eigenvalues [4]. Those are further used to fit a linear curve to the noise eigenvalues, a relative prediction error above a defined threshold defines the model order. LaRGE is similar to EFT and modified EFT (M-EFT) [4]; however, it does not require to compute the probability of false alarm.

Lately, the increasing availability of computational capabilities has drawn attention to artificial intelligence and machine learning (AI/ML), especially, deep neural networks (NNs) techniques. The model order selection problem has been tackled by AI/ML methods in [7], [8]. The work in [7] proposes a NN for model order selection as a multiclass classification problem. The NN in [7] takes the channel covariance matrix as input and is trained to output the model order. However, the proposed NN architecture has a large number of training parameters. In [8], the source number detection is modeled as a regression (ERNet) as well as multiclass classification (ECNet) task. Their NNs are trained to output the model order from the knowledge of the eigenvalues of the channel covariance matrix. Nonetheless, [8] does not take advantage of the multidimensional characteristics of MIMO systems, as the method just consider the eigenvalues in the spatial domain.

Motivated by the multidimensional characteristics of MIMO orthogonal frequency division multiplexing (OFDM) systems, we propose a new AI/ML method, here called Deep-LaRGE, which tackles the model order estimation problem as a multi-label classification task, where each higher-order singular value is modeled as a Bernoulli variable. Furthermore, we propose to use higher-order singular values as input to our AI/ML to enhance the classification performance for radio channels with closely spaced multipath components (MPCs). Moreover, our proposed NN architecture can be easily adapted for channel models with higher dimensions, and more MPCs. As baseline methods, we compare to LaRGE [5] as a non-AI/ML method, and to the ECNet [8] as a an AI/ML-method. Our results shows that our AI/ML method has an increased accuracy and can better distinguish closely spaced MPCs. To the best of the authors' knowledge, Deep-LaRGE is the first AI/ML algorithm for model order selection that considers each output neuron as a Bernoulli variable which corresponds to the presence of at least one higher-order singular value at its input. Furthermore, our proposed AI/ML method is part of a larger framework for channel estimation and prediction which we have published in [9].

In this paper, Section II presents our system overview and the wireless channel model, Section III introduces our proposed method, Section IV presents our results, and Section V concludes our paper.

Regarding the notation, a, $\mathbf{a}$, $\mathbf{A}$ and $\boldsymbol{\mathcal{A}}$ represents, respectively, scalars, column vectors, matrices and D-dimensional tensors. The superscripts ^{T}, ^{H}, * denote, respectively, transposition, Hermitian transposition, and complex conjugation. For a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \times \ldots M_D}$, M_d refers to the tensor dimension on the d^{th} mode. A d-mode unfolding of a tensor is written as $[\boldsymbol{\mathcal{A}}]_{(d)} \in \mathbb{C}^{M_d \times M_{d+1} \ldots M_D M_1 \ldots M_{d-1}}$ where all d-mode vectors are aligned as columns of a matrix. The d-mode vectors of $\boldsymbol{\mathcal{A}}$ are obtained by varying the d^{th} index from 1 to M_d and keeping all other indices fixed. Moreover, $\boldsymbol{\mathcal{A}} \times_d \mathbf{U}$ is the d-mode product between a D-way tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \cdots \times M_D}$ and a matrix $\mathbf{U} \in \mathbb{C}^{J \times M_d}$. The d-mode product is computed by multiplying $\mathbf{U}$ with all d-mode vectors of $\boldsymbol{\mathcal{A}}$. In addition, $\boldsymbol{\mathcal{A}} \sqcup_d \boldsymbol{\mathcal{B}}$ denotes the concatenation of $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ among the d^{th} mode. The concatenation $\sqcup_d$ operation also applies to matrices.

II. MIMO OFDM Scenario

As our scenario we consider environments with many scatterers, i.e., urban macro and urban micro, in frequency bands below 6 GHz. In a MIMO OFDM system, the base station (BS) and the user equipment (UE) are equipped with uniform linear arrays (ULAs) with M_T and M_R antennas, respectively. Moreover, there are N_{sub} sub-carriers. In a fixed time slot, the MIMO channel at each sub-carrier $\mathbf{H}_{n_{\text{sub}}} \in \mathbb{C}^{M_R \times M_T}$ is modeled as

$$\mathbf{H}_{n_{\text{sub}}} = \sum_{i=1}^{L} \alpha_i e^{-j2\pi\left(f_c + \frac{(n_{\text{sub}}-1)}{N_{\text{sub}}}\right)\tau_i} \mathbf{a}_R(\theta_i)\mathbf{a}_T(\phi_i)^H, \quad (1)$$

where f_c is the carrier frequency, n_{sub} is the sub-carrier index, L is the number of MPCs, τ_i, α_i, θ_i, and ϕ_i are, respectively, the delay, complex amplitude, direction of arrival (DoA), and direction of departure (DoD) of the i^{th} MPC. The ULA steering vector at the receiver side $\mathbf{a}_R$ is modeled as

$$\mathbf{a}_R(\theta_i) = [1, e^{j\mu_i}, e^{j2\mu_i}, \ldots e^{j(M_R-1)\mu_i}]^T, \quad (2)$$

where $\mu_i = \frac{2\pi}{\lambda}\Delta_d \cos\theta_i$, is the spatial frequency and $\Delta_d = \frac{\lambda}{2}$ is the spacing between the antenna elements. The ULA steering at the transmitter side $\mathbf{a}_T$ is modeled in a similar way.

Moreover, we assume that the transmitter uses a fixed grid of beams (GoB) [10], [11] as beamformers, and that the transmitter beam has already been selected, i.e., following the beam management procedures defined by 3GPP [12]. Therefore, without loss of generality, our channel model can be simplified as a SIMO OFDM wireless channel $\mathbf{H} \in \mathbb{C}^{M_R \times N_{\text{sub}}}$ at the receiver side. The channel at each sub-carrier $\mathbf{h}(n_{\text{sub}})$ is modeled as

$$\mathbf{h}(n_{\text{sub}}) = \sum_{i=1}^{L} \alpha_i e^{-j2\pi\left(f_c + \frac{(n_{\text{sub}}-1)}{N_{\text{sub}}}\right)\tau_i} \mathbf{a}_R(\theta_i) + \mathbf{z}(n_{\text{sub}}), \quad (3)$$

where $\mathbf{h}(n_{\text{sub}})$ is a column of $\mathbf{H} \in \mathbb{C}^{M_R \times N_{\text{sub}}}$ and $\mathbf{z}(n_{\text{sub}}) \in \mathbb{C}^{M_R \times 1}$ is a zero mean circularly symmetric complex Gaussian noise process.

Due to the high correlation between the MPCs, we apply spatial smoothing [13] in the sub-carriers dimension. From the N_{sub} sub-carriers, we take K of them to compute the smoothing. Therefore, the new sub-carriers dimension is $N'_{\text{sub}} = N_{\text{sub}} - K + 1$. The selection matrix for the k^{th} smoothing sub-block is defined as

$$\mathbf{J}_k = \left[\mathbf{0}_{(N'_{\text{sub}},\ k-1)} \quad \mathbf{I}_{N'_{\text{sub}}} \quad \mathbf{0}_{(N'_{\text{sub}},\ K-k)}\right] \in \mathbb{R}^{N'_{\text{sub}} \times N_{\text{sub}}}, \quad (4)$$

and the smoothed channel tensor $\boldsymbol{\mathcal{H}}$ is computed by

$$\boldsymbol{\mathcal{H}} = \left[\mathbf{H}\mathbf{J}_1^T \sqcup_3 \mathbf{H}\mathbf{J}_2^T \ \ldots \ \sqcup_3 \mathbf{H}\mathbf{J}_K^T\right] \in \mathbb{C}^{M_R \times N'_{\text{sub}} \times K}, \quad (5)$$

where the K channel smoothed matrices are concatenated in the third dimension, such that $\boldsymbol{\mathcal{H}} \in \mathbb{C}^{M_R \times N'_{\text{sub}} \times K}$ is our 3-dimensional channel tensor.

III. AI/ML for Model Order Selection

Inspired by LaRGE [5] and Unitary Tensor ESPRIT [6], we propose an AI/ML method designed as a multi-label classification task which observes all the d-mode wireless channel singular values and classifies its multidimensional input as signal or noise singular value. Each output neuron models a Bernoulli variable and has values ranging between 0 and 1. The closer to 1, the more confident the NN is that the neuron represents a signal singular value. Hence, we set a threshold ξ to define the decision region. Therefore, neurons with output above ξ are signal singular values, which are later summed to express the model order. In the following, we present the data pre-processing and the implementation details of our AI/ML architecture and training.

A. Data pre-processing

First, we compute the forward-backward averaged version of the channel tensor $\boldsymbol{\mathcal{H}}$ as [6]

$$\boldsymbol{\mathcal{Y}} \doteq \left[\boldsymbol{\mathcal{H}} \sqcup_3 \left(\boldsymbol{\mathcal{H}}^* \times_1 \boldsymbol{\Pi}_M \times_2 \boldsymbol{\Pi}_{N'_{\text{sub}}} \times_3 \boldsymbol{\Pi}_K\right)\right] \quad (6)$$

where $\boldsymbol{\Pi}_p$ is a $p \times p$ exchange matrix with ones on its anti-diagonal and zeros otherwise, and $\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{M_R \times N'_{\text{sub}} \times 2K}$.

Second, we take advantage of the centro-Hermitian characteristics of the forward-backward averaged tensor $\boldsymbol{\mathcal{Y}}$, as in Unitary Tensor ESPRIT [6], and apply a real data transformation $\boldsymbol{\mathcal{F}} = \varphi(\boldsymbol{\mathcal{Y}}) \in \mathbb{R}^{M_R \times N'_{\text{sub}} \times 2K}$ which is computed as

$$\varphi(\boldsymbol{\mathcal{Y}}) = \boldsymbol{\mathcal{Y}} \times_1 \mathbf{Q}_{M_R}^H \times_2 \mathbf{Q}_{N'_{\text{sub}}}^H \times_3 \mathbf{Q}_{2K}^H, \quad (7)$$

where $\mathbf{Q}_p \in \mathbb{C}^{p \times p}$ is a left-$\boldsymbol{\Pi}$-real matrix, i.e., $\boldsymbol{\Pi}\mathbf{Q}_p^* = \mathbf{Q}_p$. In this way, we avoid to compute covariance matrices, and reduce the complexity by working with real numbers only.

Third, we compute the HOSVD of $\boldsymbol{\mathcal{F}}$ as

$$\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \quad (8)$$

where $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{M_R \times N'_{sub} \times 2K}$ is the core tensor, and $\mathbf{U}_1 \in \mathbb{R}^{M_R \times M_R}$, $\mathbf{U}_2 \in \mathbb{R}^{N'_{sub} \times N'_{sub}}$, $\mathbf{U}_3 \in \mathbb{R}^{2K \times 2K}$ are the unitary matrices of the d-mode singular vectors, in this case $d = 1, 2, 3$. The d-mode singular values are computed by the singular value decomposition (SVD) of the d-mode unfolding of $\boldsymbol{\mathcal{F}}$, as in

$$[\boldsymbol{\mathcal{F}}]_{(d)} = \mathbf{U}_d \, \mathbf{\Sigma}_d \, \mathbf{V}_d^H, \tag{9}$$

where $\mathbf{U}_d \in \mathbb{R}^{M_d \times M_d}$, $\mathbf{V}_d \in \mathbb{R}^{\tilde{M}_d \times \tilde{M}_d}$ are unitary matrices, and $\mathbf{\Sigma}_d \in \mathbb{R}^{M_d \times \tilde{M}_d}$ has the d-mode singular values $\sigma_i^{(d)}$ on its main diagonal, and $\tilde{M}_d = \frac{M_R N'_{sub} 2K}{M_d}$.

As our channel model is 3-dimensional, we compute the SVD in each of the three unfoldings of $\boldsymbol{\mathcal{F}}$, where

$$\begin{aligned} \boldsymbol{\sigma}^{(1)} &= \mathrm{diag}(\mathbf{\Sigma}_1) \in \mathbb{R}^{M_R \times 1}, \\ \boldsymbol{\sigma}^{(2)} &= \mathrm{diag}(\mathbf{\Sigma}_2) \in \mathbb{R}^{N'_{sub} \times 1}, \text{ and} \\ \boldsymbol{\sigma}^{(3)} &= \mathrm{diag}(\mathbf{\Sigma}_3) \in \mathbb{R}^{2K \times 1} \end{aligned} \tag{10}$$

are the d-mode singular value vectors which serve as input to LaRGE [5] for computing the global eigenvalues and estimating the model order.

Inspired by LaRGE, our AI/ML input consists of scaling the singular values by the logarithmic function $\boldsymbol{\sigma}_s^{(d)} = \ln(\boldsymbol{\sigma}^{(d)})$, and reshaping each d-mode vector to size N. When reshaping dimensions, we check the sizes of each $\boldsymbol{\sigma}_s^{(d)}$ vector and decide on a common size for all the three d-mode singular values vectors. If the size of one of the 3 dimensions is much smaller than the other 2 dimensions, we choose N to be of the same order of magnitude as the bigger dimensions. Hence, we extend the size of the smallest d-mode singular value vector by copying its smallest singular value on the extended vector positions. However, if all the d-mode singular value vectors have sizes of a similar order of magnitude, we select $N = \min\{M_R, N'_{sub}, 2K\}$. Therefore, some d-mode singular values are filtered out when $M_d > N$. Finally, the input matrix to our AI/ML method is

$$\mathbf{G} = \left[\boldsymbol{\sigma}_s^{(1)}(1:N) \ \sqcup_2 \ \boldsymbol{\sigma}_s^{(2)}(1:N) \ \sqcup_2 \ \boldsymbol{\sigma}_s^{(3)}(1:N)\right], \tag{11}$$

where $\mathbf{G} \in \mathbb{R}^{N \times 3}$ for a 3-dimensional channel tensor. From an implementation perspective, this AI/ML method can be applied in parallel for every transmitter beam candidate. However, if we can observe a channel tensor with more dimensions, e.g., including an antenna array at the transmitter side as in equation (1) such that $\boldsymbol{\mathcal{H}} \in \mathbb{C}^{M_R \times M_T \times N'_{\mathrm{sub}} \times K}$, we propose to concatenate all the available d-mode singular value vectors along the second dimension and input them to our AI/ML architecture. Similarly, if we can perform measurements of SIMO channels at different carrier frequencies, as in equation (3), the d-mode singular value vectors of each carrier measurement should be concatenated along the second dimension. Therefore, our generic AI/ML input is called $\mathbf{G} \in \mathbb{R}^{N \times D}$, where D is the total number of available d-mode singular value vectors. In Section IV, we show that increasing the number of observable dimensions enhances the classification accuracy.

B. AI/ML architecture and training

We design a NN for the model order selection task as a supervised learning problem for multi-label classification. The input to our NN is the matrix $\mathbf{G} \in \mathbb{R}^{N \times D}$ with all the vectors

TABLE I
DESCRIPTION OF THE NN FOR MODEL ORDER SELECTION OF MULTIDIMENSIONAL DATA.

Layer	N_{filter}	Filter size	Activation
Conv1D	8	3	ReLU
Conv1D	1	3	ReLU
Dense	8	-	ReLU
Dense	8	-	ReLU
Dense	N	-	Sigmoid

TABLE II
COMPARISON OF NN COMPLEXITY.

Baseline NN			Our NN		
Layer	N_{filter}	# parameters	Layer	$(N_{\mathrm{filter}}, \text{filter size})$	# parameters
Dense	F_3	$F_3(ND+1)$	Conv1D	(F_1, Q_1)	$F_1(Q_1 D + 1)$
Dense	F_4	$F_4(F_3+1)$	Conv1D	(F_2, Q_2)	$F_2(Q_2 F_1 + 1)$
Dense	N	$N(F_4+1)$	Dense	F_3	$F_3(O_2 F_2 + 1)$
-	-	-	Dense	F_4	$F_4(F_3+1)$
-	-	-	Dense	N	$N(F_4+1)$
Total	-	$F^2 + F(N(D+1)+2) + N$	Total	-	$F^2 + F(2N + Q(D-1) + 5) + N + 1$

of d-mode singular values. The NN architecture is presented in Table I where the 1-dimensional convolutional layers are introduced to efficiently process the d-mode singular values, see Section III-C. Each neuron in the output layer is modeled as a Bernoulli variable that represents the probability of that neuron being a signal singular value. The output of our NN is a vector with **1**s and **0**s of size $1 \times N$. Each label '1' denotes a signal singular value, while each '0' label represents a noise singular value, e.g., $[1\ 1\ 1\ 0\ 0\ 0\ 0\ 0]$ for $N = 8$ and 3 signal singular values.

The activation function of the output layer is the sigmoid function γ, calculated as

$$\gamma(x) = \frac{1}{1 + e^{-x}}, \tag{12}$$

where x is the input to the layer activation function. For the loss function, used to update the gradient descent algorithm, we select the binary cross-entropy which is computed for each $n^{\mathrm{th}} \in [1, 2, \ldots, N]$ output neuron as

$$\mathcal{L} = \sum_{n=1}^{N} -\left(y_n \log(\tilde{y}_n) + (1 - y_n)\log(1 - \tilde{y}_n)\right) \tag{13}$$

where y_n is the label, and $\tilde{y}_n$ is the predicted score, both for the n^{th} output neuron.

Therefore, we design a multi-label classification NN that maps the multidimensional singular values at the input to signal or noise singular values at the output. Moreover, our NN output includes a reliability measure. As each n^{th} output neuron can have a value between 0 and 1, the NN is more reliable that the n^{th} neuron represents a signal singular value when $\tilde{y}_n$ is closer to 1. Hence, we set a threshold ξ to define the decision region, such that $\tilde{y}_n$ is a signal singular value if $\tilde{y}_n > \xi$. Then, we sum the number of output neurons which are above ξ to get an estimate of the model order.

C. AI/ML computational complexity

For the design of our NN architecture, we target a structure which works efficiently with D-dimensional data. Hence, we

ISBN 978-3-8007-6050-3

proposed the NN architecture in Table I, which avoids a rapid increase on the number of trainable parameter when the number of observable dimensions D increases. Our baseline NN architecture is the ECNet [8], where only two dense inner layers are used for the task of source number detection, and the number of sources is modeled as a multinoulli distribution over the output neurons, such that $\sum_{n=1}^{N} p_n = 1$, where p_n is the probability of the n^{th} neuron being a signal singular value, and the neuron with highest probability corresponds to the estimated number of sources. In addition to this statistical modeling difference, we input $\mathbf{G} \in \mathbb{R}^{N \times D}$ higher-order singular values and output $\mathbf{g} \in \mathbb{R}^{N \times 1}$. In Table II, we present our NN architecture and the baseline NN using generic parameters for accounting the computational complexity of the AI/ML models. If we set the number of filters, or neurons, as $F = F_1 = F_3 = F_4$, $F_2 = 1$, and the size of each convolutional filter to $Q = Q_1 = Q_2$, the output of the second convolutional layer O_2 becomes $O_2 = N - 2Q + 2$. The final number of parameters for our NN is $F^2 + F(2N + Q(D-1) + 5) + N + 1$, while for the baseline NN it is $F^2 + F(N(D+1)+2) + N$. Since $Q \ll N$, the complexity of our proposed NN architecture does not increase as fast as in the architecture with dense layers only. Hence, our NN architecture is more suitable for multidimensional data.

IV. Simulations and Results

For the model order selection task, we simulate 3-dimensional wireless channels with a varying number of MPCs, as modeled in Equation (3). We consider an OFDM system with channel bandwidth of 20 MHz, each sub-carrier spaced by 15 kHz, sampling frequency $f_s = 30.72$ MHz, and one pilot signal every 12 sub-carriers, and $N_{\text{sub}} = 100$ resource blocks. Five channel datasets are generated, with 1 to 5 MPCs, each with 3000 channel samples. The channel gains follow a Rayleigh distribution, and the delays and DoAs are drawn from a uniform distribution. All the MPCs within a dataset have different delay values on the interval $[0.05, 5]\mathrm{T}_s$, where $T_s = 1/f_s$, and DoA values range between $[0^o, 120^o]$. The SIMO system is parameterized by $M = 8$, $\Delta_d = \lambda/2$, and a signal to noise ratio (SNR) of 20 dB. In addition, we consider three different values for the carrier frequency, in order to test if the classification accuracy increases when more d-mode singular values are available. Hence, we have one dataset with varying MPCs at baseband ($f_c = 0$), and a second dataset with varying MPCs with two 3-dimensional channels at downlink ($f_{c_{\text{down}}} = 2.6$ GHz) and uplink ($f_{c_{\text{up}}} = 2.8$ GHz) according to Equation (3).

For a performance comparison, we select the ECNet [8] as the AI/ML baseline. The ECNet was designed to estimate the number of sources using the information of 1-dimensional eigenvalues. Therefore, we take $\mathbf{h}_b \in \mathbb{C}^{1 \times N_{\text{sub}}}$, the SISO version of the channel in Equation (3), and apply smoothing with $K = 50$ and $N'_{\text{sub}} = 51$ on the sub-carriers domain $\mathbf{H}_b \in \mathbb{C}^{N'_{\text{sub}} \times K}$. After that, we compute the forward-backward averaging. Then, we compute the singular values in the delay domain $\mathbf{\Sigma}_b \in \mathbb{R}^{N'_{\text{sub}} \times 2K}$. The input vector to ECNet is

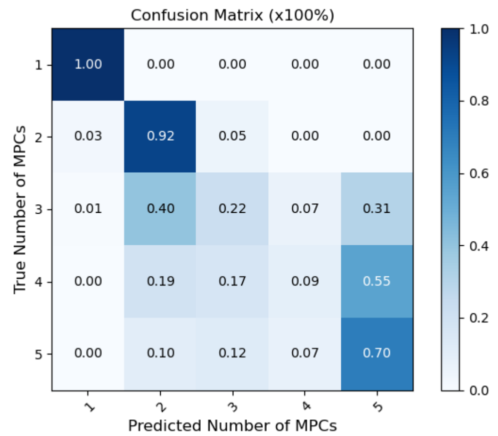

Fig. 1. Classification accuracy per class for 1-dimensional ECNet [8]. The input is $\mathbf{g}^T \in \mathbb{R}^{50 \times 1}$, where $N = 50$ and $f_c = 0$.

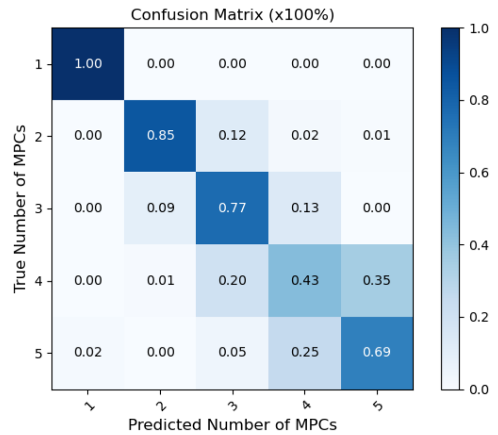

Fig. 2. Classification accuracy per class for ECNet [8]. The input is $\mathbf{G} \in \mathbb{R}^{8 \times 6}$, where $N = 8$, and the higher-order singular values for $f_{c_{\text{down}}} = 2.6$ GHz and $f_{c_{\text{up}}} = 2.8$ GHz are concatenated on the second dimension.

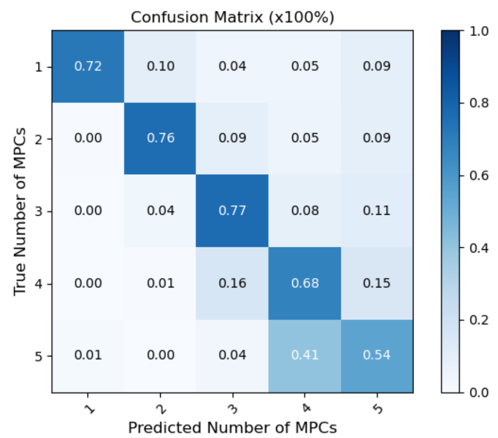

Fig. 3. LaRGE [5] classification accuracy per class for $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{8 \times 51 \times 100}$, the 3-dimensional channel dataset at $f_c = 0$, and $\rho = 0.57$.

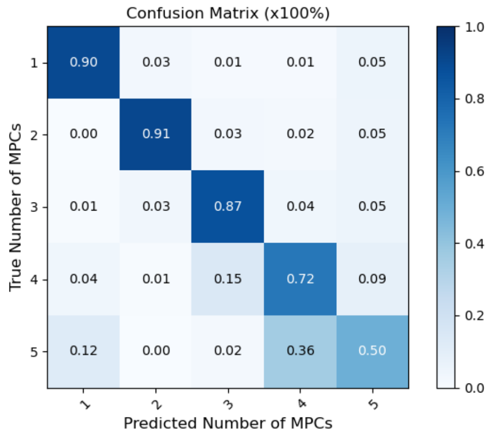

Fig. 4. LaRGE [5] classification accuracy per class for $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{8\times51\times100\times8\times51\times100}$, the combined channel dataset at $f_{c_{\text{down}}} = 2.6$ GHz, and $f_{c_{\text{up}}} = 2.8$ GHz. The decision threshold is $\rho = 0.57$.

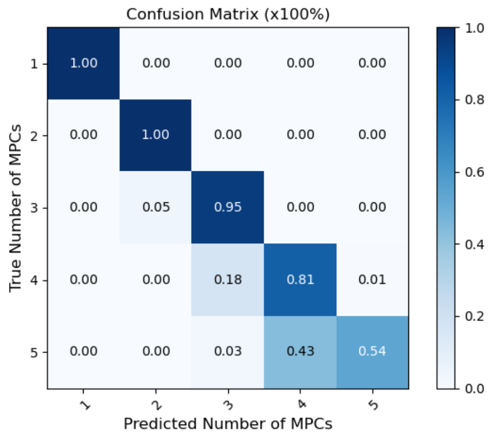

Fig. 5. Classification accuracy per class of the Deep-LaRGE algorithm. The input is $\mathbf{G} \in \mathbb{R}^{50\times3}$, where $N = 50$, and the 3-dimensional channel dataset at $f_c = 0$. The threshold is $\xi = 0.8$.

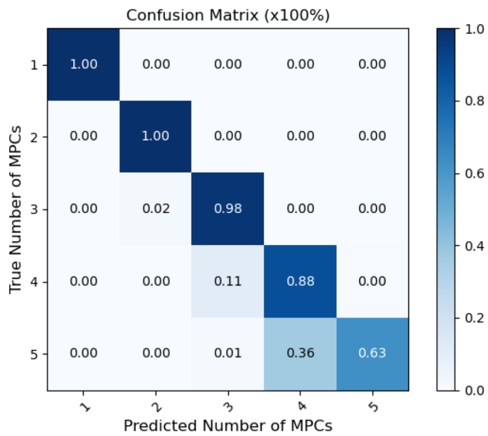

Fig. 6. Classification accuracy per class of the Deep-LaRGE algorithm. The input is $\mathbf{G} \in \mathbb{R}^{50\times6}$, where $N = 50$, and concatenated 3-dimensional channel datasets at $f_{c_{\text{down}}} = 2.6$ GHz, and $f_{c_{\text{up}}} = 2.8$ GHz. The threshold is $\xi = 0.8$.

$\mathbf{g} \in \mathbb{R}^{50\times1}$, with $N = 50$. Figure 1 presents the classification accuracy for the ECNet in our 1-dimensional dataset. It can be observed that estimating the number of sources is especially difficult for 1-dimensional channels due to the potentially close spacing between the MPCs. Figure 2 shows the improvement on the classification performance of the ECNet when we change its input to the higher order singular values of the tensor channels on the uplink and the downlink $\mathbf{G} \in \mathbb{R}^{8\times6}$, without extension of the 2-mode singular values as ECNet did not propose it. The classification accuracy of 3 and 4 MPCs improved, but the ECNet still has difficulties in correctly identifying 4 and 5 MPCs.

As the non-AI/ML baseline, we select LaRGE [5] which takes as input the higher order singular values as in Equation (10), and the decision threshold is set to $\rho = 0.57$. Figure 3 presents the results of LaRGE for the 3-dimensional channels at $f_c = 0$, $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{8\times51\times100}$. In addition, Figure 4 shows the classification accuracy for LaRGE when coupling the higher-order singular values for the channels at downlink and uplink $[\boldsymbol{\mathcal{S}}_{\text{up}}, \boldsymbol{\mathcal{S}}_{\text{down}}] \in \mathbb{R}^{8\times51\times100\times8\times51\times100}$. If compared to the ECNet, LaRGE with the coupled tensor channels as input has a better performance for classifying $2, 3$ and 4 MPCs.

Regarding our proposed AI/ML method for model order selection, we implement and train the architecture in Table I using TensorFlow 2.0, Keras and Python. We set $K = 50$, $N'_{\text{sub}} = 51$, and assert the number of observable higher-order singular values at the NN input to $N = K = 50$. The weights of the layers are initialized from a truncated normal distribution with zero mean and standard deviation $\sigma = \sqrt{\frac{1}{m}}$, where m is the input size of each weight layer [14]. The initial learning rate is set to 2×10^{-3} and the Adam optimizer [15] is used. The supervised training runs for 400 epochs with batch size of 128. For both datasets, at baseband and downlink/uplink, the 15000 dataset samples are divided as 70% for training, and 30% for testing. By our NN design, the NN output is also of size $N = 50$, and the threshold for the signal singular value is set to $\xi = 0.8$, which already has a higher reliability margin than LaRGE where $\rho = 0.56$.

Figure 5 presents the Deep-LaRGE classification accuracy per class for the 3-dimensional dataset at baseband, $\mathbf{G}^{50\times3}$. It can be observed that our proposed AI/ML method is already more accurate than the baselines presented, as channels with 1 and 2 MPCs are classified with 100 % accuracy and classifying channels with more MPCs than they actually have just happens between 4 and 5 MPCs, mainly due to the clustering of MPCs. Moreover, in Figure 6 we plot the Deep-LaRGE classification accuracy for the 3-dimensional dataset on the uplink and the downlink, $\mathbf{G}^{50\times6}$. The results show that increasing the number of higher-order modes which serve as input to Deep-LaRGE also helps to improve its classification accuracy of all classes. Moreover, at the threshold of $\xi = 0.8$, none of the data samples are classified with more MPCs than they actually have, which is challenging for the other classifiers, especially for the ECNet in Figure 2.

In addition to the higher-order singular values as input,

ISBN 978-3-8007-6050-3

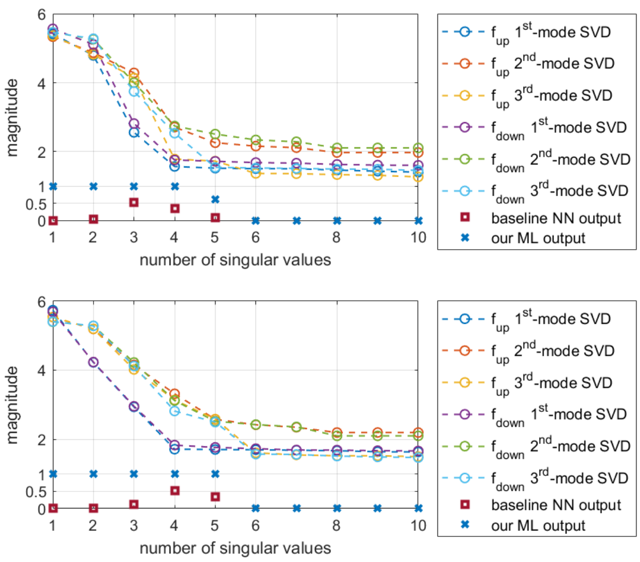

Fig. 7. Comparison of two sample classification outcomes for the 5 MPCs dataset at downlink and uplink. We plot the higher-order singular values which serve as input to the ECNet [8] as well as to the Deep-LaRGE algorithm, and the classification output of both AI/ML approaches.

we attribute the success of Deep-LaRGE to the design decision of formulating model order selection as a multi-label classification problem. Since each output neuron is classified independently from its neighbors, our AI/ML method could achieve a high reliability. On the contrary, in [8] the classification is modeled as a multi-class classification problem which is easily confused when the dataset has closely spaced MPCs. Figure 7 supports this claim, where we plot two sample results for classifying 3-dimensional channels with 5 MPCs with downlink and uplink information. The input to both NNs is plotted, as well as their respective classification output. In the channel sample on the top, Deep-LaRGE classifies it as 4 MPCs with a high reliability, and the classification would be correct if $\xi = 0.5$. Nevertheless, for the same channel, the ECNet has higher probability for 3 MPCs, but the probability difference between 3 and 4 MPCs classes is small. In the channel sample on the bottom of Figure 7, Deep-LaRGE is confident on classifying the input as 5 MPCs. However, the ECNet is unsure between 4 and 5 MPCs and, finally, miss-classifies it as 4 MPCs. Hence, the ECNet classification performance is not reliable.

V. Conclusion

In this paper we propose the Deep-LaRGE algorithm for model order selection in MIMO OFDM systems with almost coherent MPCs. Our results for model order selection shows that the use of higher-order singular values as input to the Deep-LaRGE algorithm is effective on improving the classification performance. Moreover, our AI/ML design is successful in enhancing the classification accuracy without rapidly increasing the NN computational complexity for multidimensional inputs. For future work, we may consider the effect of varying SNRs and to have an AI/ML architecture for model order selection working directly on the channel matrix.

Acknowledgement

This research was partly funded by German Ministry of Education and Research (BMBF) under grant 16KIS1184 (FunKI).

References

[1] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[2] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[3] A. Quinlan, J.-P. Barbot, P. Larzabal, and M. Haardt, "Model order selection for short data: An exponential fitting test (EFT)," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–11, 2006.

[4] J. P. C. L. da Costa, M. Haardt, F. Romer, and G. Del Galdo, "Enhanced model order estimation using higher-order arrays," in *Proc. Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*, 2007, pp. 412–416.

[5] A. A. Korobkov, M. K. Diugurova, J. Haueisen, and M. Haardt, "Multi-dimensional model order estimation using LineAr Regression of Global Eigenvalues (LaRGE) with applications to EEG and MEG recordings," in *Proc. 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1005–1009.

[6] M. Haardt, F. Roemer, and G. Del Galdo, "Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3198–3213, 2008.

[7] A. Barthelme, R. Wiesmayr, and W. Utschick, "Model order selection in DoA scenarios via cross-entropy based machine learning techniques," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4622–4626.

[8] Y. Yang, F. Gao, C. Qian, and G. Liao, "Model-aided deep neural network for source number detection," *IEEE Signal Processing Letters*, vol. 27, pp. 91–95, 2020.

[9] B. Vilas Boas, W. Zirwas, and M. Haardt, "Combining AI/ML and PHY layer rule based inference - some first results," in *Proc. IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022, pp. 1–5.

[10] T. Obara, S. Suyama, J. Shen, and Y. Okumura, "Joint fixed beamforming and eigenmode precoding for super high bit rate massive MIMO systems using higher frequency bands," in *Proc. IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, 2014, pp. 607–611.

[11] R. S. Ganesan, W. Zirwas, B. Panzner, K. I. Pedersen, and K. Valkealahti, "Integrating 3D channel model and grid of beams for 5G mMIMO system level simulations," in *Proc. IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–6.

[12] *Study on New Radio Access Technology Physical Layer Aspects (Release 14)*, 3^{rd} Generation Partnership Project, 2017, 3GPP TR 38.802 V14.2.0 (2017-09).

[13] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 806–811, 1985.

[14] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

