Low Complexity Deep Learning Based Coordinated Beamforming for mmWave Massive MIMO Vehicular Networks

Vasileios P. Rekkas*, Sotirios P. Sotiroudis*,Panagiotis Sarigiannids†, George K. Karagiannidis ‡, Christos G. Christodoulou§, Sotirios K. Goudos*¶

*ELEDIA@AUTH, School of Physics, Aristotle University of Thessaloniki, Thessaloniki, Greece,
e-mail: (vrekkas, ssoti, sgoudo)@physics.auth.gr

†Department of Informatics and Telecommunications Engineering, University of Western Macedonia, Kozani, Greece
Email: psarigiannidis@uowm.gr

‡School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece
Email: geokarag@auth.gr

§Department of Engineering and Computing, The University of New Mexico, USA
e-mail: christos@unm.edu

¶Department of Electronics and Communication Engineering, Bharath University, Chennai, Tamilnadu, India

2024 Panhellenic Conference on Electronics & Telecommunications (PACET) | 979-8-3503-1884-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/PACET60398.2024.10497006

***Abstract*—Enabling high mobility in millimeter-wave (mmWave) systems is crucial for next-generation wireless communication systems. Challenges arise in applications such as vehicular communications and wireless virtual/augmented reality. MmWave systems grapple with concerns related to narrow beams, signal vulnerability to obstructions that affect coverage, and the necessity for frequent handovers. Additionally, identifying optimal beamforming vectors in large antenna array mmWave systems involves substantial training overhead, impacting efficiency. This paper introduces a novel solution to address challenges in enabling high mobility in mmWave systems. The approach combines deep learning (DL) and coordinated beamforming to support applications such as vehicular communications and wireless virtual/augmented reality. The solution involves multiple coordinating base stations (BSs) serving a mobile user, utilizing a single pilot training sequence. A DL model predicts beamforming vectors based on received signals, providing reliable coverage, low latency, and minimal training overhead for highly mobile mmWave applications. Simulations demonstrate its effectiveness in high-mobility scenarios, approaching the performance of an optimal solution, a genie-aided solution that knows the optimal beamforming vectors without any training overhead.**

***Index Terms*—Millimeter wave, coordinated beamforming, deep learning, data rates.**

I. Introduction

Millimeter wave (mmWave) spectrum has gained considerable interest in recent years as an enabler of high-data-rate communications due to the wide bandwidths available for wireless communications services [1]. The mmWave frequency band spans from 30 GHz to 300 GHz, but the inherent physical limitations result in shorter transmission distances for higher frequencies, leading to elevated path loss. Therefore, problems such as path loss and blockage must be addressed by using smaller cellular cells for higher frequencies [2].

Massive multiple-input multiple-output (mMIMO) systems can propagate signals in the same time-frequency resource and serve numerous users concurrently, employing hundreds of antennas simultaneously. The highly directional transmissions, facilitated by the short wavelength of mmWave, allow for the installation of numerous antennas at the transceiver in a cellular network, substantially enhancing network capacity. In scenarios with poor channel estimation, path loss, and terminal-specific antenna correlation, large-scale antenna systems significantly increase upload and download rates [3]. Additionally, in environments with rapid propagation changes, large-scale antenna systems reliably provide high throughput for both forward and reverse link connections. Using mMIMO systems can improve the capacity and reliability of wireless systems, effectively overcoming the challenges of increased path loss in the mmWave spectrum [3].

This paper uses data-driven approaches to identify optimal beamforming vectors, utilizing various deep learning (DL) models. The goal is to achieve reliable coverage and low latency for highly mobile mmWave applications while minimizing training overhead. The remainder of the paper is organized as follows. Section II briefly describes the system model and the simulation setup of the study. Section III presents the DL models that are used for coordinated beamforming. In Section IV simulation parameters and results are presented, while conclusions are drawn in Section V.

II. System model

This study adopts the system model described in [4], where there are N base stations (BS), each equipped with M antennas. Facilitated by an appropriate infrastructure, these BSs collaborate to transmit information to a mobile user, with a single antenna. To mitigate the effects of the frequency-selective wireless channel between any BS and the user, orthogonal frequency division multiplexing (OFDM) with m subcarriers is used. For the m-th subcarrier, the connection

y_m between input and output to send the symbol s_m can be described as:

$$y_m = \sum_{n=1}^{N} r_{m,n}^{u} a_n f_{m,n} s_m + g_m \quad (1)$$

where $r_{m,n} \in \mathbb{C}$ represents the impulse response of the channel on the m-th subcarrier connecting the n-th BS with the user u, and a_n corresponds to the analog beamformer utilized by the n-th BS. Furthermore, $f_{m,n} \in \mathbb{C}$ is a precoding factor provided by the n-th Base Station (BS). It is essential to emphasize that these precoding factors collectively constitute the digital precoding vector $f_m \in \mathbb{C}^{N\times 1}$, constructed collaboratively by the participating BSs. Finally, g_m implements an additive white Gaussian noise to the system.

A. Simulation setup

This paper introduces an innovative solution for highly mobile mmWave applications that combine communication and DL. The proposed coordinated beamforming system serves a mobile user using deep learning to predict BSs' beamforming vectors. This prediction is based on signals received at distributed BSs using omni- or quasi-omni-beam patterns, capturing multipath signatures of user location and surroundings. The DL model requires minimal training overhead and adapts to any environment without pre-deployment training. Integrated with coordinated beamforming, it inherits coverage and reliability advantages. This paper aims to contribute an improved integrated DL and coordinated beamforming solution, reducing coordination overhead for wide-coverage and low-latency gains to enhance and maximize the system's effective achievable rates. The neural networks (NNs) studied in this work are compared with the DL approach and a baseline model as described in [4]. The baseline approach, relies on uplink training for the creation of baseband beamforming vectors, where BSs initially choose their beamforming vectors from a predetermined codebook. Subsequently, a central processor formulates the baseband beamforming to achieve coherent combining at the user end. This approach though, necessitates significant training overhead. Furthermore, a comparison is made between the proposed NNs and a genie-aided solution, that knows the optimal beamforming vectors without any training overhead, as presented in [4] is conducted. Finally, a complexity comparison is made between the proposed NNs and the DL model in [4], to emphasize not only the higher achievable rates of the NNs but also the lower complexity (in terms of parameters) of the proposed methods.

The effectiveness of the NNs will be assessed by comparing the effective achievable rate with the increasing sizes of the training dataset. The DeepMIMO dataset [5] is utilized, where channel configurations are generated using ray-tracing data from the wireless inSite simulator [6]. Specifically, we focus on the "O1-60" ray-tracing scenario, involving multiple BSs concurrently serving a mobile user in the 60 GHz band.

TABLE I
PARAMETER VALUES

Parameter	Value
Active BSs	3, 4, 5, 6
Active Users	54300 (Row 1000 to row 1300, 181 users per row)
BSs antennas	256 ($T_x = 1, T_y = 32, T_z = 8$)
Bandwidth	0.5 GHz
OFDM sub-carriers	1024
OFDM sub-carriers limit	$K_{DL} = 64$

III. Deep learning models

Tabular data are typically organized with each column representing a distinct feature and each row representing a unique instance. Traditional models such as support vector machine (SVM) and tree-based algorithms are commonly used to analyze tabular data, demonstrating satisfactory performance with limited data [7]. In contrast, DL models, while generally outperformed by traditional methods on smaller datasets, excel on larger datasets because of their capacity to discern intricate patterns within the data. To address the limitations of traditional algorithms, there is a focus on developing neural networks specifically tailored for tabular data [8].

A. WideDeep Learning

WideDeep Learning constitutes a machine learning model structure that harnesses the strengths of both DL and conventional linear models, creating a hybrid approach. An exposition of the architecture is presented [9]:

- Wide Component: The wide component adopts a generalized linear model

$$y = \mathbf{v}^T \mathbf{u} + c \quad (2)$$

  where y represents the prediction, v denotes the model parameters, u signifies a vector of features, and c stands for the bias term. Crucial cross-product transformations that capture interactions among binary features are defined as:

$$\psi_k(\mathbf{u}) = \prod_{i=1}^{d} u_{mki} \quad (3)$$

  where $\psi_k(\mathbf{u})$ represents a boolean variable with a value of one when the ith feature is included in the k-th transformation ψ_k; otherwise, it is zero. This accounts for interactions among binary features, which introduces nonlinearity to the generalized linear model.
- Deep Component: The deep component, a feed-forward neural network, converts categorical features into low-dimensional embedding vectors. The hidden layers perform computations as:

$$z^{(l+1)} = g(A^{(l)} z^{(l)} + \mathbf{b}^{(l)}) \quad (4)$$

- Joint Training: Integration of the wide and deep components involves a joint training approach, where a weighted sum of their output log odds forms the prediction. This combined prediction is then subjected to a common

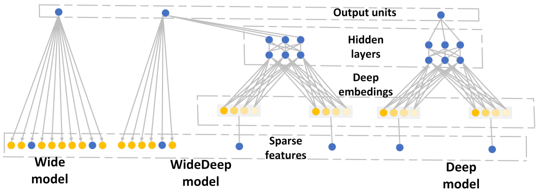

Fig. 1. WideDeep architecture.

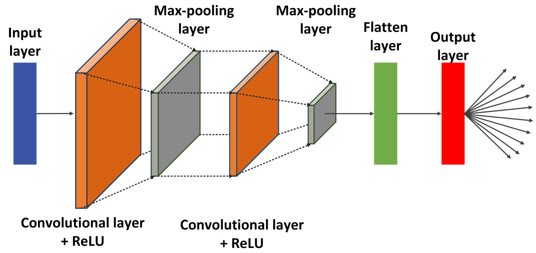

Fig. 2. CNN architecture.

logistic loss function during joint training. The optimization process encompasses all parameters simultaneously, incorporating both the wide and deep segments, as well as the weights governing their summation. In particular, the wide segment serves to address the deficiencies of the deep part with minimal cross-product feature transformations. The joint training of the WideDeep model involves backpropagating gradients from the output, simultaneously impacting both components through mini-batch stochastic optimization. This process integrates the predictions of both elements, assigning specific weights to ensure a proportional influence on the prediction. Calculating the weighted sum involves multiplying the output of each component by its assigned weight and applying the sigmoid function, resulting in a probability-like value (zero and one). The prediction of the model for a regression task can be described as follows:

$$P(Y=1|\mathbf{u})=\sigma(\mathbf{w}_{\text{wide}}^T[\mathbf{u},\psi(\mathbf{u})]+\mathbf{w}_{\text{deep}}^T z^{(lf)}+b) \quad (5)$$

where Y represents the binary class label, σ denotes the sigmoid function, $\psi(\mathbf{u})$ stands for the cross-product transformations of the original features u, and b is the bias term. The vector $\mathbf{w}_{\text{wide}}$ comprises all the weights associated with the wide model, and $\mathbf{w}_{\text{deep}}$ refers to the weights applied to the final activations $z^{(lf)}$.

B. Convolutional neural network (CNN)

CNNs represent prominent models in DL, proficient at discerning patterns in data without the need for manual feature extraction. Their adaptability to retraining in new recognition tasks makes them particularly suitable for DL challenges involving large data sets. The schematic overview of a comprehensive CNN architecture is illustrated in Fig. 2, encompassing the following layer types [10]:

- Input Layer: This initial layer introduces the input data in a format conducive to subsequent processing, facilitating the extraction of high-level features through a series of hidden layers.
- Convolution Layers: This segment computes the convolution of the data parameters using multiple filters of uniform shape compared to the input layer but with smaller dimensions. The process results in a feature map of the data after convolution across the entire input.
- Pooling Layers: These layers reduce the dimensions of subsequent layers by executing downsampling through max or average pooling. This reduction improves computational efficiency and reduces complexity.
- Output Layer: In this final layer, CNN generates outputs.

Also, a non-linear activation function is applied to each element in the feature space. The rectified linear unit (ReLU) activation function, often chosen in the CNN literature, is mathematically defined as $f(x)=\max(0,x)$.

IV. Simulation Results

In this study, two neural networks, namely WideDeep learning and CNN are evaluated in comparison to a DL approach, a baseline, and an optimal genie-aided approach where the implementations outlined in [4] are used.

The WideDeep model integrates both wide (linear) and deep (non-linear) components. The wide component is a generalized linear model that comprises 64 units and processes spatial features, capturing linear relationships in the data. In our case, the nonlinearity in the wide network is not introduced from a cross-product transformation, but utilizing a single dense layer with 64 neurons and the ReLU activation function. The deep component, an MLP, processes spatial characteristics through two dense layers, each with 256 neurons, ReLU activation, and a dropout rate of 0.1 . The output of both components is concatenated, forming a unified representation. The output layer, comprising 512 neurons, predicts optimal beamforming using ReLU activation. Model compilation employs Mean Squared Error loss and the Adam optimizer, while training involves iterations with a batch size of 100 over 10 epochs.

The proposed CNN model uses spatial characteristics and consists of four convolutional layers with eight filters each, a kernel size of three, ReLU activation, and max pooling. The flattened output is connected to four dense layers, each with 256 neurons, ReLU activation, and a dropout rate of 0.02 . The output layer comprises 512 neurons representing the number of beams activated with the ReLU function. The model is trained using the Adam optimizer with a batch size of 128 for 10 epochs, and Mean Squared Error loss function.

Both models were tested on datasets with increasing DL size ratios, demonstrating flexibility and efficacy in various beam prediction scenarios for mmWave communication systems. The training set is generated using a maximum of 80% of the total available data, while the remaining data are reserved for testing. The performance of neural networks will be evaluated

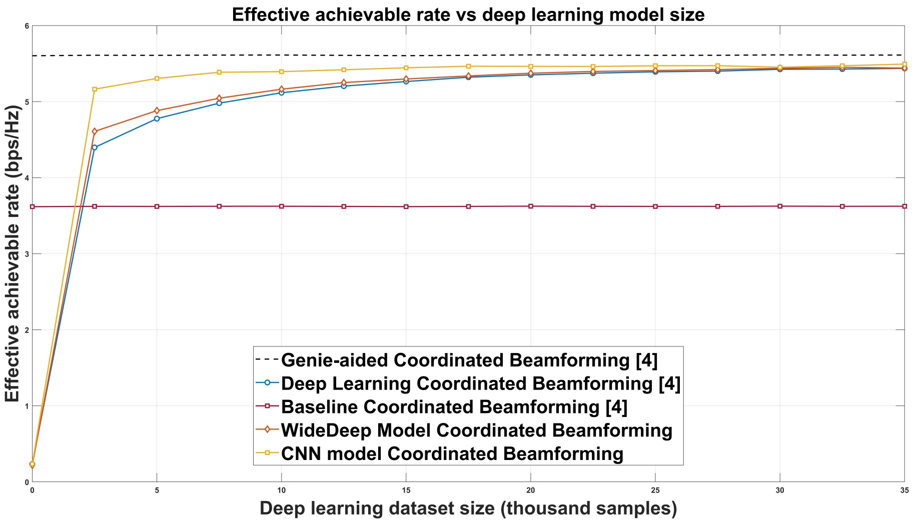

Fig. 3. Simulation results

based on the effective achievable rate relative to the size of the training data set, as shown in Fig. 3.

Surpassing the performance of the previously presented models [4], the proposed DL approaches achieve optimal performance with a reduced amount of data. Importantly, this performance improvement is achieved with a complexity reduction, as detailed in Table II. WideDeep approach achieves better effective achievable rates for the DL model, up to 4.81% for 2500 samples. Additionally, the proposed CNN architecture outperforms all models and offers rates close to the optimal genie-aided solution, with an improvement of up to 17.43% for the same data size and better rates for all sizes.

TABLE II
COMPARISON OF NETWORKS PARAMETERS

Model	Number of Parameters	Parameters Size (MB)
DL model [4]	460.544	1.78 MB
WideDeep model	312.384	1.19 MB
CNN model	335.728	1.28 MB

V. CONCLUSION

This paper investigates neural network architectures to address the coordinated beamforming problem while minimizing the necessary dataset size for training. Using the DeepMIMO dataset, improvements have been showcased in terms of achievable rates and complexity reduction. Future research endeavors aim to explore the problem within a multi-user context while utilizing multiple antennas. Furthermore, the study will include the examination of hybrid architectures for transceivers and the exploration of higher frequency bands.

ACKNOWLEDGMENT

This work has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096456 (NANCY). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or SNS JU. Neither the European Union nor the SNS JU can be held responsible for them.

The research work was also supported by the Hellenic Foundation for Research and Innovation (HFRI) under the 3rd Call for HFRI PhD Fellowships(Fellowship Number: 6646).

REFERENCES

[1] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.

[2] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, "Millimeter-wave massive mimo communication for future wireless systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 836–869, 2017.

[3] P. Tarafder and W. Choi, "Deep reinforcement learning-based coordinated beamforming for mmwave massive mimo vehicular networks," *Sensors*, vol. 23, no. 5, 2023.

[4] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[5] A. Alkhateeb, "Deepmimo: A generic deep learning dataset for millimeter wave and massive mimo applications," *arXiv preprint arXiv:1902.06435*, 2019.

[6] Remcom, "Wireless Insite," https://www.remcom.com/wireless-insite.

[7] Q. Zheng, Z. Peng, Z. Dang, L. Zhu, Z. Liu, Z. Zhang, and J. Zhou, "Deep tabular data modeling with dual-route structure-adaptive graph networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 9715–9727, 2023.

[8] M. I. Iqbal, M. S. H. Mukta, A. R. Hasan, and S. Islam, "A dynamic weighted tabular method for convolutional neural networks," *IEEE Access*, vol. 10, pp. 134 183–134 198, 2022.

[9] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[10] K. Ahmad and S. Hussain, "Machine learning approaches for radio propagation modeling in urban vehicular channels," *IEEE Access*, vol. 10, pp. 113 690–113 698, 2022.