The Indonesian Journal of Computer Science
www.ijcs.net
Volume 13, Issue 3, June 2024
https://doi.org/10.33022/ijcs.v13i3.3941

Deep Learning based Channel Estimation and Hybrid Beamforming for 5G Massive MIMO Wireless Communications

[1]Thwe Zin Tun, [2]Zin Mar Lwin, [3]Tin Tin Hla
[1]thwezintun57@gmail.com, [2]dr.zinmar80@gmail.com, [3]tintinhla99@gmail.com
[1]Government Technical Institute, Myanmar
[2]Mandalay Technological University, Myanmar
[3]Mandalay Technological University, Myanmar

Article Information

Submitted : 8 May 2024
Reviewed : 30 May 2024
Accepted : 15 Jun 2024

Keywords

channel estimation, hybrid beamforming, massive MIMO, 5G mmWave, LSTM network

Abstract

Hybrid beamforming (BF), which divides beamforming operation into radio frequency (RF) and baseband (BB) domains, plays a critical role in MIMO communication at millimeter-wave(mmWave) frequencies. This paper also introduces offline training and prediction schemes for channel estimation and hybrid beamforming. The aim of this paper is that to increase spectral efficiency over more data streams by leveraging the deep learning based LSTM network. The LSTM network is used to train the numeric values from sequence data and predict on new sequence data. The performance is evaluated under different parameters including number of data streams (1, 2, 3 and 4) with different signal-to-noise ratio (SNR) for different carrier frequencies (28GHz, 38GHz, 60GHz and 73GHz) through computer simulation using MATLAB. The simulation results verified that the proposed method can achieve higher spectral efficiency when the number of data streams increases and the value of SNR-Test increases too.

Attribution-ShareAlike 4.0 International License

A. Introduction

MmWave communication and massive MIMO are the technologies that will enable 5G design objectives. There is a mutual integration between these two technologies. mmWave-massive MIMO communications, which offers notable gains in energy and spectrum efficiency. It also increases mobile network capacity and attains high multiplexing gains. Massive MIMO linear precoding techniques and the most common non-linear precoding schemes are presented in [1]. Using fewer RF chains, hybrid beamforming offers a cost-effective solution by drastically lowering hardware costs and power consumption. The authors of [2] offered a comprehensive analysis of hybrid beamforming for mmWave systems and beyond, based on a newly developed taxonomy for various hardware configurations.

An summary of 5G research, standardization testing, and deployment problems was given by the authors in [3]. In 5G wireless networks, hybrid transceivers that integrate lower-dimensional digital signal processing units with high-dimensional analog phase shifters and power amplifiers enable mmWave large multiple-input multiple-output (MIMO) communications. In reference to system models of the hybrid transceiver's structures, digital and analog beamforming matrices with potential antenna configuration scenarios, and hybrid beamforming in heterogeneous wireless networks, the authors in [4] tracked the development of hybrid beamforming for massive MIMO communications. A thorough analysis of hybrid beamforming, a crucial component of huge MIMO mmWave systems, is given in [4].

A deep learning (DL) framework to handle both hybrid beamforming and channel estimation was proposed in [5] with the goals of minimizing complexity and offering robustness. Future wireless communication networks will be able to support millimeter wave frequency band communications thanks to the development of highly directed beamforming technology. In [6], a signal subspace-based high-precision multipath channel estimation technique was covered. Maximizing the total rate that can get close to the completely digital beamforming system performance is the goal of hybrid beamforming. For 5G wireless networks, a combination of mmWave, hybrid beamforming techniques, and massive MIMO systems yields better data speeds and more cell coverage. The performance of a multiuser-massive MIMO hybrid beamforming system at 28, 39 GHz of FR2 frequency bands as well as at 66 GHz has been examined by the authors in [7] by examining the impact of varying variables such as the quantity of users, transmitting and receiving antennas, and the type of modulation.

The advancement in wireless telecommunications from 1G to 6G was briefly discussed in [8], and then, the concept of beamforming was introduced. Instead than focusing on feedback accuracy, the primary goal of the CsiFBnet is to optimize beamforming performance gain [9]. The authors in [9] applied this idea to two representative scenarios: single-cell systems and multi-cell systems. An unsupervised learning strategy trained the entire neural networks in [9]. This paper [10] proposed a data-driven deep learning (DL)-based unified hybrid beamforming framework for both the time division duplex (TDD) and frequency division duplex (FDD) systems with implicit channel state information (CSI) by modeling the transmission modules such as an end-to-end (E2E) neural network. For multiuser large multiple-input multiple-output (MIMO)-orthogonal frequency

division multiplexing (OFDM) systems, hybrid beamforming designs were taken into consideration in [11]. One alternating maximization approach, where the analog precoding was optimized using Riemannian manifold optimization, was used to maximize the weighted spectral efficiency [11].

Using back propagation, a unique joint hybrid processing framework (JHPF) that enables end-to-end optimization was designed through the application of deep learning (DL) [12]. The three components of the [12] framework are the hybrid processing designer, signal flow simulator, and signal demodulator. These three parts, in turn, produce the hybrid processing matrices for the transceiver, simulate an airborne signal transmission, and use neural networks (NNs) to map detected symbols to their original bits. Through its singular value decomposition (SVD), the unconstrained (optimal) beamformers of a transceiver were developed that approach the highest attainable data rates [13]. The effectiveness of the designs was confirmed by simulation findings [14] [15], which also show that the hybrid transceivers' performance is comparable to that of their fully digital version. The performance of the suggested algorithm was assessed by the authors in [16] using both the Rayleigh and mmWave channels.

In order to optimize the system's spectral efficiency, the authors suggested two approaches [17]. The system's performance can be enhanced with the use of these two techniques. The feasibility of combining low-resolution phase shifters with hybrid precoders and combiners in mmWave systems was examined in study [18]. A machine learning-based design process that works with a variety of beamforming/combining architectures was used by the authors in [19]. The performance evaluation of a deep learning approach when used in 5G mmWave multicellular networks is the goal of [20]. Massive MIMO channel estimation is still a challenging problem due to the hybrid structure of the precoding and the flaws in the RF chain, despite the fact that numerous solutions for hybrid beamforming have been studied in the last few years [21]–[24]. The authors in [25] created a codebook-based hybrid precoding for downlink multiuser mmWave large MIMO systems, building on earlier research.

In order to provide hybrid beamforming and channel estimation for 5G mmWave massive MIMO communication systems, this study suggests a deep learning-based beamforming strategy using LSTM networks that have been specially constructed. In a huge MIMO system with many data streams, a single transmitter and receiver communication at mmWave frequencies are examined. Channel state information (CSI) feedback refers to the information about the wireless channel conditions provided by the users (or mobile station) back to the base station. This information helps the base station adapt its transmission parameters to optimize the communication link, including beamforming and power control. The high spectral efficiency can be obtained without CSI feedback by leveraging deep learning based LSTM network, instead it only knows the spatial statistics of the channel. Five LSTM networks are used for training, testing and prediction in this system. Thousands of instances of the channel matrix H are generated to provide simulation data for testing and training. The channel covariance matrix (CCM) , **R**, is generated from the channel matrix **H**. Then, corresponding to each instance of the channel covariance matrix **R**, the fully digital optimal beamforming matrix $\mathbf{F}_{\text{opt}}, \mathbf{W}_{\text{opt}}$ was calculated from the orthogonal matching

pursuit (OMP) algorithm. After the LSTM network has been trained and evaluated, it can be deployed to make predictions on new, unseen sequences of data such as the value of $\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}$. Depending on the value of $\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}$, the spectral efficiency can be calculated.

The LSTM-1 and LSTM-2 deep networks are employed at the BS. They take in a channel covariance matrix (CCM) as input and provide hybrid precoders as an output. The LSTM-1 network trains the real and imaginary part of dataset **R** and predict the $\mathbf{F}_{RF}$ value on the testing data. The LSTM-2 is used to train the angular part of the dataset **R** and predict $\mathbf{F}_{RF}$ value on testing data. At the mobile station (MS), there are three deep networks. The received signal **R** is trained by using LSTM-3 to estimate the channel and predict the channel matrix **H1** on the testing data. The estimated channel matrix **H1** is then fed to LSTM-4 and LSTM-5 to design the hybrid combiner weights at the output. The real and imaginary part of the channel matrix **H1** is trained and predict to get the $\mathbf{W}_{BB}$ on the testing data. The angular part of the channel matrix **H1** is trained and predict to get the $\mathbf{W}_{RF}$ on the testing data.

The rest of the paper is organized as follows. The wideband mmWave channel system model and problem formulation are explained in Section **B** of the ensuing section. Channel estimation are described in Section **C**. Section **D** presents overview of LSTM network. Section **E** introduces learning-based hybrid beamformer design. Section **F** gives the numerical simulation results and discussion, and Section **G** concludes the paper.

Notation: Throughout the paper, boldface lower case and upper case symbols represent vector and matrix quantities, respectively. In the case of a vector **a**, $[\mathbf{a}]_i$ represents its **i**-th element. For a matrix **A**, $[\mathbf{A}]_{:,i}$ and $[\mathbf{A}]_{i,j}$ denote the **i**-th column and the (**i**, **j**)-th entry, respectively. $\mathbf{A}^*$, $\mathbf{A}^T$ and $\mathbf{A}^H$ represent the conjugate, transpose and Hermitian of **A**. The Kronecker product is denoted by Ä while the Hadamard product is given by e . || . $||_F$ is the Frobenious norm, I_N is the identity matrix of size N×N. E **{.}** denotes the statistical expectation.

B. System Model and Problem Formulation

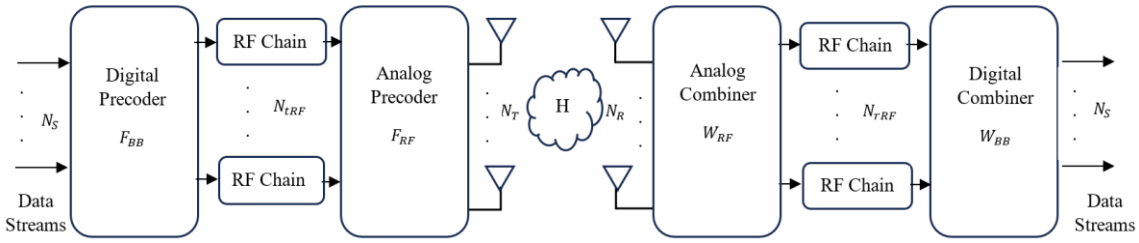


Figure 1. System architecture of a hybrid beamforming transceiver based on mmWave massive MIMO

In this work, a single user is considered in mmWave MIMO communication system with multiple antennas. Let N_s be the number of data streams to be

transmitted from the base station with N_T transmit antennas to the user with N_R antennas. The BS is equipped with N_{tRF} analog phase shifters with analog beamformer $\mathbf{F}_{RF} \in \mathbb{C}^{N_T \times N_{tRF}}$ and baseband beamformer $\mathbf{F}_{BB} \in \mathbb{C}^{N_{tRF} \times N_S}$. Then the transmitted signal becomes $x = \mathbf{F}_{RF}\mathbf{F}_{BB}s$ where $s \in \mathbb{C}^{N_s}$ is the symbol vector desired to be transmitted. $\mathbf{F}_{RF}$ consists of analog phase shifters, each of which has unit-modulus elements, i.e., $\left|[\mathbf{F}_{RF}]_{i,j}\right|^2 = 1$. Also, the power constraint $\left\|\mathbf{F}_{RF}\mathbf{F}_{BB}\right\|^2 = N_s$ that is enforced by the normalization of the baseband precoder $\mathbf{F}_{BB}$.

Assuming a block-fading channel model, the received signal at the MS is given by

$$\bar{y} = \sqrt{\rho}\mathbf{H}\mathbf{F}_{RF}\mathbf{F}_{BB}s + n \quad (1)$$

where ρ represents the average received power, $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ is the mmWave channel matrix and $n: CN(0,\sigma^2\mathbf{I}_{N_R})$ denotes the additive white Gaussian noise (AWGN) vector. At the receiver, the received signal is first processed by analog combiners $\mathbf{W}_{RF}$, then the receiver employs low-dimensional $N_{RF} \times N_s$ digital combiners $\mathbf{W}_{BB}$ to process the RF signal to obtain the received symbol vector as $\hat{y} = \mathbf{W}_{BB}^H\mathbf{W}_{RF}^H\bar{y}$, i.e.,

$$\hat{y} = \sqrt{\rho}\mathbf{W}_{BB}^H\mathbf{W}_{RF}^H\mathbf{H}\mathbf{F}_{RF}\mathbf{F}_{BB}s + \mathbf{W}_{BB}^H\mathbf{W}_{RF}^H n \quad (2)$$

where the analog combiners $\mathbf{W}_{RF} \in \mathbb{C}^{N_R \times N_{rRF}}$ have element-wise constraint $\left\{[\mathbf{W}_{RF}]_{:,i}[\mathbf{W}_{RF}]_{:,i}^H\right\}_{ii} = 1$ similar to the RF precoders.

The presence of substantial route loss resulting in limited spatial selectivity scattering is one of the primary characteristics of mmWave channels. High antenna correlation may also arise from the usage of large densely packed antenna arrays in mmWave transceivers. Beam-space or virtual channel representation can be used to simulate the extremely directed character of propagation at mmWave frequencies and the high dimensionality of MIMO channels with massive antenna arrays. Hence, the channel matrix **H** include the contributions of L clusters, each of which has $\mathbf{N_{sc}}$ scattering path/rays within the cluster. Thus, we can represent the downlink channel matrix as

$$\mathbf{H} = \sqrt{\frac{N_T N_R}{N_{sc} L}} \sum_{l=1}^{L} \sum_{r=1}^{N_{sc}} \alpha_{l,r} a_R(\Phi_{l,r},\theta_{l,r}) a_T^H(\Phi_{l,r},\theta_{l,r}) \quad (3)$$

where $\alpha_{l,r} \in \mathbb{C}$ denotes the complex gain corresponding to the r-th path in the l-th cluster, which are assumed to be independent zero-mean Gaussian random variables. $a_R(\Phi_{l,r},\theta_{l,r})$ and $a_T(\Phi_{l,r},\theta_{l,r})$ represent the array response vectors of the receive and transmit antenna arrays respectively. In the millimeter wave frequency band, signal scattering is very significant, thus multiple path effects have to be included. Here, both the transmitter and receiver sides, due to the use of array antennas, both the direction angle $\Phi_{l,r}$ and the elevation angle $\theta_{l,r}$ need to be considered. In particular, the array response vectors of receive and transmit arrays for a uniform rectangle array (URA) is defined as

$a_T(\Phi,\theta) = \left[1, e^{j\frac{2\pi}{\lambda}\bar{d}p}, e^{j\frac{2\pi}{\lambda}\bar{d}2p} .. e^{j\frac{2\pi}{\lambda}AMp}, e^{j\frac{2\pi}{\lambda}\bar{d}q} e^{j\frac{2\pi}{\lambda}\bar{d}(p+q)} ... e^{j\frac{2\pi}{\lambda}d(Mp+Nq)}\right]^T$ and

$a_R(\Phi,\theta) = \left[1, e^{j\frac{2\pi}{\lambda}\bar{d}p}, e^{j\frac{2\pi}{\lambda}\bar{d}2p} .. e^{j\frac{2\pi}{\lambda}AMp}, e^{j\frac{2\pi}{\lambda}\bar{d}q} e^{j\frac{2\pi}{\lambda}\bar{d}(p+q)} ... e^{j\frac{2\pi}{\lambda}d(Mp+Nq)}\right]^T$ respectively. Where $p = \cos(\Phi_{l,r})\cos(\theta_{l,r}), q = \sin(\Phi_{l,r})\cos(\theta_{l,r})$; M and N denote the number of horizontal and vertical antennas of the array antenna and meet $MN = N_T$, respectively; $\bar{d}$ is the uniform distance between the antennas and $\lambda = \frac{c_0}{f_c}$ is the wavelength for the carrier frequency f_c with the speed of light c_0.

Statistical beamforming techniques reduce feedback overhead by providing an intermittent update of the channel information based on the channel statistics rather than instantaneous feedback. Various algorithms can be used to estimate the channel covariance matrix (CCM) in practice. These include compressed covariance sensing approaches, power angular spectrum estimates, and temporal averaging techniques that gather a single snapshot of incoming signals. The CCM is accessible at the BS in this work because it is a specific field study. The covariance channel matrix at the transmitter can be written as (4).

$$\mathbf{R} = \frac{1}{N_T} E\{\mathbf{H}^H\mathbf{H}\} \quad (4)$$

In real-world applications, estimating the channel matrix can be a difficult undertaking, particularly when numerous antennas are used for massive MIMO communications. Literature indicates several mmWave channel estimation techniques [22], [23], [24]. Within the DL framework, a deep network estimates the channel by taking in the received signals as input and producing the channel matrix estimate at the output layer.

Specifically, designing of hybrid precoders $\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}$ by maximizing the overall spectral efficiency of the system. Given that the Gaussian symbols are conveyed via the millimeter-wave (mmWave) channel, the problem of designing a hybrid beamformer can be expressed as follows.

$$\max_{\mathbf{F}_{RF},\mathbf{F}_{BB},\mathbf{W}_{RF},\mathbf{W}_{BB}} \log_2 \left| \mathbf{I}_{N_s} + \frac{\rho}{N_s} \grave{U}_n^{1} \mathbf{W}_{BB}^H \mathbf{W}_{RF}^H \mathbf{H} \mathbf{F}_{RF} \mathbf{F}_{BB} \mathbf{F}_{BB}^H \mathbf{F}_{RF}^H \mathbf{H}^H \mathbf{W}_{RF} \mathbf{W}_{BB} \right|,$$

Subject to: $\mathbf{F}_{RF} \hat{I} F_{RF}, \mathbf{W}_{RF} \hat{I} W_{RF}$,

$$\left\| \mathbf{F}_{RF}\mathbf{F}_{BB} \right\|_F^2 = N_s, \quad (5)$$

where $\grave{U}_n = \sigma_n^2 \mathbf{W}_{BB}^H \mathbf{W}_{RF}^H \mathbf{W}_{RF} \mathbf{W}_{BB} \hat{I} £^{N_S \times N_S}$ corresponds to the combiner-processed noise term in the received signal (2). F_{RF} and W_{RF} are the feasible sets for the RF precoder and combiners which obey the unit-modulus constraint. The aim of this paper is to recover $\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}$ for the given received pilot signal. In fact, the hybrid beamformers are designed via orthogonal matching pursuit (OMP) algorithm which does not require a predefined codebook.

C. Channel Estimation

In real-world applications, estimating the channel matrix can be a difficult undertaking, particularly when massive MIMO systems are involved and there are many antennas. Within the DL framework, a deep network uses the pilot signals it received during the preamble step to estimate the channel. In this case, the downlink scenario where the BS activates RF chains $\bar{f}_u \in \mathbb{C}^{N_T}$ to transmit pilot signals $\bar{s}_u$ on the beams for $u = 1,...,M_T$. Then, the receiver activates M_R RF chains to apply $\bar{w}_v$ for $v = 1,...,M_R$ to process the received pilots. Since the number of RF chains in the receiver is limited by $N_{RF}(< M_R)$, only N_{RF} combining vectors can be used at a single channel use. Hence, the total channel use in the channel acquisition process is $\left\lceil \frac{M_R}{N_{RF}} \right\rceil$. Then, the transmit and receive beamforming matrices become $\bar{\mathbf{F}} = [\bar{f}_1, \bar{f}_2, ..., \bar{f}_{M_T}] \in \mathbb{C}^{N_T \times M_T}$ and $\bar{\mathbf{W}} = [\bar{w}_1, \bar{w}_2, ..., \bar{w}_{M_R}] \in \mathbb{C}^{N_R \times M_R}$ respectively. Specifically, $\bar{\mathbf{F}}$ and $\bar{\mathbf{W}}$ can be constructed as the first M_T (or M_R) column vectors of an $N_T \times N_T$ or $N_R \times N_R$ Discrete Fourier Transform (DFT) matrix.

After processing through combiners, the received pilot signal becomes

$$\bar{\mathbf{Y}} = \bar{\mathbf{W}}^H \mathbf{H} \bar{\mathbf{F}} \bar{S} + \tilde{N} \quad (6)$$

where $\bar{S} = \text{diag}\{\bar{s}_1, ..., \bar{s}_{M_T}\}$ denotes the pilot signals and $\tilde{N} = \bar{\mathbf{W}}^H \bar{N}$ is the effective noise matrix where $\bar{N}$ denotes the AWGN matrix which corrupts the pilot training data by $SNR_{\tilde{N}}$. Without loss of generality, $\bar{S} = \mathbf{I}_{M_T}$, then the received signal becomes

$$\bar{\mathbf{Y}} = \bar{\mathbf{W}}^H \mathbf{H} \bar{\mathbf{F}} + \tilde{N} \quad (7)$$

By processing $\bar{\mathbf{Y}}$, the initial channel estimate (ICE) as

$$\mathbf{Y} = T_T \bar{\mathbf{Y}} T_R \quad (8)$$

where $T_T = \begin{cases} \bar{\mathbf{W}}, M_R < N_R \\ \left(\bar{\mathbf{W}}\bar{\mathbf{W}}^H\right)^{-1}\bar{\mathbf{W}}, M_R \geq N_R \end{cases}$ and $T_R = \begin{cases} \bar{\mathbf{F}}^H, M_T < N_T \\ \bar{\mathbf{F}}^H\left(\bar{\mathbf{F}}\bar{\mathbf{F}}^H\right)^{-1}, M_T \geq N_T \end{cases}$. Because $\mathbf{Y}$ will be employed in the proposed DL framework in the future to acquire better channel estimates, it is the initial channel estimate (ICE). Similarly, the pretrained network LSTM receives $\mathbf{Y}$ after it is received at the receiver, which enhances the channel estimation performance. To create the hybrid combiners, the enhanced channel estimate is then inserted into the LSTM.

D. Overview of LSTM Network

This work implements DNN-based hybrid beamforming and channel estimation using LSTM model. A Recurrent Neural Network (RNN) architecture called LSTM successfully resolves the vanishing gradient problem that arises with a poorly constructed RNN. Recursive neural networks (RNNs) use the outputs of individual nodes to influence the input of that same node in a recursive manner. RNN can maintain memory since its current output is predicated on earlier calculations. Nevertheless, it is well known that RNNs have a "vanishing gradient" issue, in which the loss function's derivative with regard to the weight parameter gets extremely tiny. In order to alleviate this problem, LSTMs have been modified to incorporate new gates that enhance gradient control and safeguard long-range dependencies.

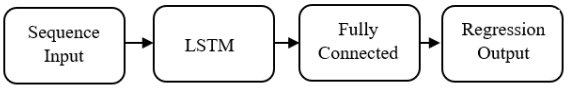


Figure 2. The LSTM network architecture for regression

Figure 2 shows the construction of a basic LSTM network for regression output. A sequence input layer, an LSTM layer, a fully connected layer, and a regression output layer make up the network's final layers. Figure 3 illustrates the fundamental composition of the current LSTM model as well as the cell algorithms. Each time step in an LSTM comprises a unique status known as the cell state that houses the long-term memory's worth of data. In Figure 3, the time series input is shown at the bottom and the result is shown at the top.

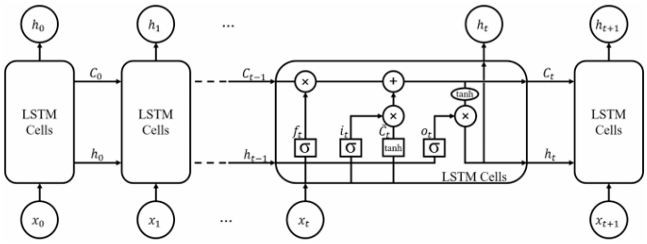


Figure 3. Basic LSTM layer structure for the time step 0 to t+1, with a detailed calculation illustration shown in the LSTM cell at time step t [26]

Every time step, each LSTM cell modifies six parameters. Equations (9) through (14) display the comprehensive algorithms. The forget gate parameter f_t, which decides how much of the previous cell state needs to be forgetten by a sigmoid function with a linear calculation on the current input xt and the previous result h_{t-1}. The linear equations in different steps have different input weights (W), recurrent weight (R) and biases (b) in each LSTM cell. The closer f_t is to 0, the more the sigmoid function forgets the previous cell stae h_{t-1}. The input gate i_t, which decides what new information is going to be remembered by adding it to the cell state. The input gate parameter i_t is calculated by the sigmoid function with a linear relation on x_t and h_{t-1} as well. $\%_t$is the candidate of new cell state values, and it was calculated by a tanh function with a linear relation on x_t and h_{t-1}. Then, the cell state c_t is updated. In the end, the output parameter o_t is calculated by the sigmoid function with a linear relation on x_t and h_{t-1}. The current step of the final output result, h_t, is the production of o_t and the tanh function value of cell state c_t.

$$f_t = \sigma_{\%}(W_f x_t + R_f h_{t-1} + b_f) \quad (9)$$

$$i_t = \sigma_{\%}(W_i x_t + R_i h_{t-1} + b_i) \quad (10)$$

$$\%_t = \sigma_c(W_{\%} x_t + R_{\%} h_{t-1} + b_{\%}) \quad (11)$$

$$o_t = \sigma_{\%}(W_o x_t + R_o h_{t-1} + b_o) \quad (12)$$

$$c_t = f_t \, e \, c_{t-1} + i_t \, e \, g_t \quad (13)$$

$$h_t = o_t \, e \, \sigma_c(c_t) \quad (14)$$

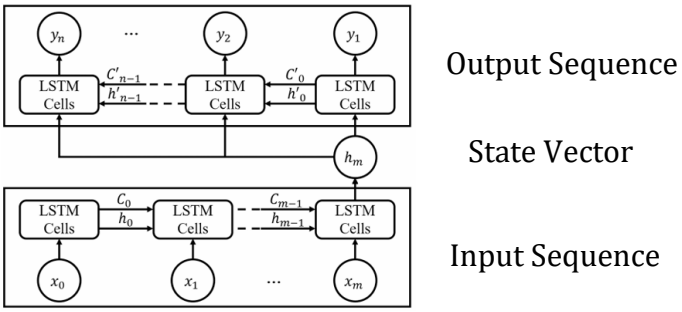


Figure 4. One-layer LSTM based sequence-to-sequence model structure with m time step input and n time step output [26]

Long-term dependency issues can be resolved using LSTM, however as Figure 3 illustrates, one drawback is that it requires the same time steps for input and output. Figure 4 displays the sequence-to-sequence regression model based on LSTM. An input sequence for the LSTM with n time steps can be created by storing the final output (input sequence) from the LSTM with m time steps in a cell called state vector. The time step problem has been resolved by this sequence-to-sequence structure as the steps of the input and output can differ. The input data, the coded vector containing all the required information from, and the predicted data are all represented in Figure 4.

E. Learning-Based Channel Estimation and Hybrid Beamformer Design

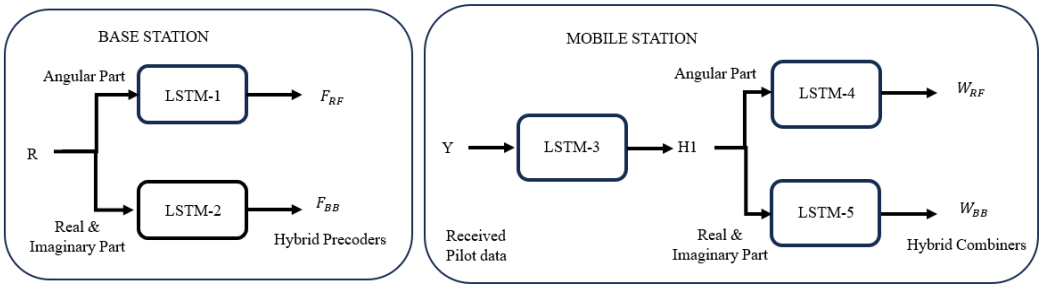


Figure 5. System architecture of mmWave MIMO based transceiver with channel estimation and hybrid beamforming

Five LSTM deep network architectures which are shown in Figure 5. LSTM-1 and LSTM-2 are used at the BS only and it learns the channel statistics from **R** and obtain the hybrid precoders $\mathbf{F}_{RF}$ and $\mathbf{F}_{BB}$. LSTM-3, LSTM-4 and LSTM-5 are placed at the MS only, to estimate the channel **H1** and construct the hybrid combiners $\mathbf{W}_{RF}$ and $\mathbf{W}_{BB}$ respectively. In the following, the details of each deep network architecture was discussed.

Input Data

The input data is partitioned into three components to enrich the input features. In particular, the real, imaginary parts and the absolute value of each entry is used for the DL networks depending on the application. This approach provides good features

for fitting the data in the training state as well as extracting new features inherit in the input. In particular, the input for LSTM-1 is denoted as $X_R \in \mathbb{R}^{N_T \times N_T \times 1}$ whose (i, j)-th entry is given by $[X_R]_{:,:,1}(i,j) = \Re\{[\mathbf{R}]_{i,j}\}$, and LSTM-2 as $X_R \in \mathbb{R}^{N_T \times N_T \times 2}$ whose (i, j)-th entry is given by $[X_R]_{:,:,1}(i,j) = \mathrm{Re}\{[\mathbf{R}]_{i,j}\}$, $[X_R]_{:,:,2}(i,j) = \mathrm{Im}\{[\mathbf{R}]_{i,j}\}$, respectively. For LSTM-3, the input is denoted as $X_Y \in \mathbb{R}^{N_R \times N_T \times 2}$ and, similarly $[X_Y]_{:,:,1}(i,j) = \mathrm{Re}\{[\mathbf{Y}]_{i,j}\}$, $[X_Y]_{:,:,2}(i,j) = \mathrm{Im}\{[\mathbf{Y}]_{i,j}\}$. Finally, the input for LSTM-4 is given by $X_{H1} \in \mathbb{R}^{N_R \times N_T \times 1}$ where $[X_{H1}]_{:,:,1}(i,j) = \Re\{[\mathbf{H1}]_{i,j}\}$, and LSTM-5 is given by $X_{H1} \in \mathbb{R}^{N_R \times N_T \times 2}$ where $[X_{H1}]_{:,:,1}(i,j) = \mathrm{Re}\{[\mathbf{H1}]_{i,j}\}$, $[X_{H1}]_{:,:,2}(i,j) = \mathrm{Im}\{[\mathbf{H1}]_{i,j}\}$ respectively.

Labeling the deep network

The labels of LSTM-1 and LSTM-2 which are the hybrid precoders $\mathbf{F}_{RF}$ and $\mathbf{F}_{BB}$. Therefore, the output label of LSTM-1 is represented by $\mathbf{z_{R1}}$ as

$$\mathbf{z_{R1}} = \left[\mathrm{vec}\{\Re\mathbf{F}_{RF}\}^T\right]^T \tag{15}$$

which is an $N_{RF}N_T \times 1$ real-valued vector. For LSTM-2, the labels by $\mathbf{z_{R2}}$ as

$$\mathbf{z_{R2}} = \left[\mathrm{vec}\{\mathrm{Re}\{\mathbf{F}_{BB}\}\}^T, \mathrm{vec}\{\mathrm{Im}\{\mathbf{F}_{BB}\}\}^T\right]^T \tag{16}$$

which is an $2N_sN_{RF} \times 1$ real-valued vector. For LSTM-3, the labels by $\mathbf{z_Y}$ as

$$\mathbf{z_Y} = \left[\mathrm{vec}\{\mathrm{Re}\{\mathbf{H1}\}\}^T, \mathrm{vec}\{\mathrm{Im}\{\mathbf{H1}\}\}^T\right]^T \tag{17}$$

which is an $2N_RN_T \times 1$ real-valued vector. Finally, the output label of LSTM-4 by $\mathbf{z_{H1}}$ as

$$\mathbf{z_{H1}} = \left[\mathrm{vec}\{\Re\{\mathbf{W}_{RF}\}\}^T\right]^T \tag{18}$$

which is an $N_RN_{RF} \times 1$ real-valued vector. And then, the output label of LSTM-5 by $\mathbf{z_{H2}}$ as

$$\mathbf{z_{H2}} = \left[\mathrm{vec}\{\mathrm{Re}\{\mathbf{W}_{BB}\}\}^T, \mathrm{vec}\{\mathrm{Im}\{\mathbf{W}_{BB}\}\}^T\right]^T \tag{19}$$

which is an $2N_{RF}N_s \times 1$ real-valued vector.

Network Architectures and Training

The deep networks in Figure 5, LSTM-1, LSTM-2, LSTM-3, LSTM-4 and LSTM-5 have the input-output pairs as $\{X_R, \mathbf{z_{R1}}\}$, $\{X_R, \mathbf{z_{R2}}\}$, $\{X_Y, \mathbf{z_Y}\}$, $\{X_{H1}, \mathbf{z_{H1}}\}$and $\{X_{H1}, \mathbf{z_{H2}}\}$ respectively. Regression layers with the appropriate size based on the application are the output layer of all networks. The network parameters that have been described are one way to get good performance for the problem under consideration, even if other network topologies with other values are equally feasible. The network parameters come from a process of hyperparameter tuning that yields the optimum results for the scenario under consideration.

Using MATLAB on a PC, the suggested deep networks are implemented and trained. Table 1 outlines the phases of the algorithm for generating training data. Every one of the five LSTM networks uses the same hyperparameter. An LSTM layer with 200 hidden units, a fully connected layer the size of the number of responses, and a dropout layer with a dropout probability of 0.5 make up each LSTM network.

Utilizing the solver "adam," the network parameters were updated at a learning rate of 0.001 and a mini-batch size of 128 samples. Ten epochs are used to train the data. Set the gradient threshold to 1 to stop the gradients from blowing up. Keep the sequences arranged according to length by setting "Shuffle" to "never." Similar to Algorithm 1, N = 100 distinct scenarios are realized in order to train the suggested LSTM structures. Synthetic additive noise is added to the training data on the CCM, channel matrix, and received pilot signal, which are defined by and SNR, respectively. For each scenario, a channel matrix (along with the related covariance matrix) is constructed. SNR value is utilized during training to strengthen the networks' resistance to distorted input characteristics. Hence, SNR = 20 dB and SNR = 0dB are considered. Hence, $SNR_R = 20\log_{10}(\frac{|[\mathbf{R}]_{i,j}|^2}{\sigma_R^2})$, $SNR_H = 20\log_{10}(\frac{|[\mathbf{H}]_{i,j}|^2}{\sigma_H^2})$ and $SNR_Y = 20\log_{10}(\frac{|[\mathbf{Y}]_{i,j}|^2}{\sigma_Y^2})$ where σ_R^2, σ_H^2, s_Y^2 are the variance of AWGN corresponding to the input data. For the prediction process, a test data which is separately generated by adding noise on received pilot signal with SNR-Test.

Table 1. Algorithm for Training data generation for DL network

Algorithm 1 Training data generation for LSTM-1, LSTM-2, LSTM-3, LSTM-4 and LSTM-5

Input: N, SNR.

Output: Training datasets for the networks in Figure 2: D_{LSTM-1}, D_{LSTM-2}, D_{LSTM-3}, D_{LSTM-4}, D_{LSTM-5}.

1: Generate channel realizations $\{\mathbf{H}^n\}_{n=1}^N$.

2: Generate channel covariance matrix realizations $\{\mathbf{R}^n\}_{n=1}^N$.

3: Initialize with t = 1 while the dataset length is T = N.

4: **for** $1 \le n \le N$ **do**

5: $\mathbf{H}^n : CN(\mathbf{H}^n, \sigma_H^2)$

6: $\mathbf{R}^n : CN(\mathbf{R}^n, \sigma_R^2)$

7: Generate received pilot signal from (6) as $\bar{\mathbf{Y}}^{(n)} = \bar{\mathbf{W}}^H \mathbf{H}^{(n)} \bar{\mathbf{F}} + \bar{N}^{(n)}$

8: Construct $\mathbf{Y}^{(n)}$ from (7) by using $\bar{\mathbf{Y}}^{(n)}$

9: Using $\mathbf{H}^{(n)}$, find the hybrid precoders $\mathbf{F}_{RF}^{(n)}$ and $\mathbf{F}_{BB}^{(n)}$ from orthogonal matching pursuit algorithm (OMP)

10: Find the hybrid combiners $\mathbf{W}_{RF}^{(n)}$ and $\mathbf{W}_{BB}^{(n)}$ from OMP

11: Input for LSTM-1, $X_R]_{i,j,1} = Đ\{[\mathbf{R}]_{i,j}\}$

12: Input for LSTM-2, $X_R]_{i,j,1} = \mathrm{Re}\{[\mathbf{R}]_{i,j}\}$, $X_R]_{i,j,2} = \mathrm{Im}\{[\mathbf{R}]_{i,j}\}$

13: Output for LSTM-1, $\mathbf{z}_{R1}$.

14: Output for LSTM-2, $D_{LSTM-1} = ((X_R^{(1)}, \mathbf{z}_{R1}^{(1)}), \ldots, (X_R^{(T)}, \mathbf{z}_{R1}^{(T)}))$.

15: Input for LSTM-3, $X_Y]_{i,j,1} = \mathrm{Re}\{[\mathbf{Y}]_{i,j}\}$, $X_Y]_{i,j,2} = \mathrm{Im}\{[\mathbf{Y}]_{i,j}\}$

16: Output for LSTM-3, $D_{LSTM-5} = ((X_H^{(1)}, \mathbf{z}_{H2}^{(1)}), ..., (X_H^{(T)}, \mathbf{z}_{H2}^{(T)}))$.

17: Input for LSTM-4, $\begin{bmatrix} X_H \end{bmatrix}_{:,1} = \text{Đ}\{[\mathbf{H}]_{:,j}\}$

18: Input for LSTM-5, $\begin{bmatrix} X_H \end{bmatrix}_{:,1} = \text{Re}\{[\mathbf{H}]_{:,j}\}$, $\begin{bmatrix} X_H \end{bmatrix}_{:,2} = \text{Im}\{[\mathbf{H}]_{:,j}\}$

19: Output for LSTM-4, $\mathbf{z}_{H1}$.

20: Output for LSTM-5, $\mathbf{z}_{H2}$.

21: $t = t + 1$.

22: **end** for n,

23: $D_{LSTM-1} = ((X_R^{(1)}, \mathbf{z}_{R1}^{(1)}), ..., (X_R^{(T)}, \mathbf{z}_{R1}^{(T)}))$.

24: $D_{LSTM-2} = ((X_R^{(1)}, \mathbf{z}_{R2}^{(1)}), ..., (X_R^{(T)}, \mathbf{z}_{R2}^{(T)}))$.

25: $D_{LSTM-3} = ((X_Y^{(1)}, \mathbf{z}_Y^{(1)}), ..., (X_Y^{(T)}, \mathbf{z}_Y^{(T)}))$.

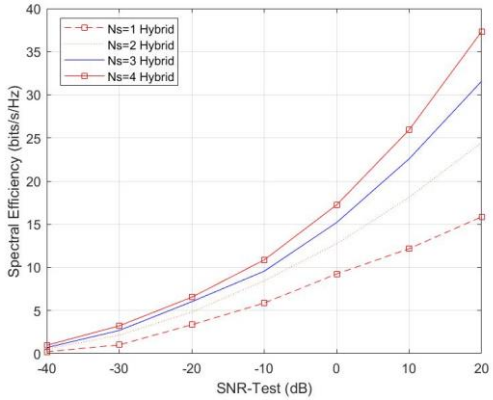
26: $D_{LSTM-4} = ((X_H^{(1)}, \mathbf{z}_{H1}^{(1)}), ..., (X_H^{(T)}, \mathbf{z}_{H1}^{(T)}))$.

27: $D_{LSTM-5} = ((X_H^{(1)}, \mathbf{z}_{H2}^{(1)}), ..., (X_H^{(T)}, \mathbf{z}_{H2}^{(T)}))$.

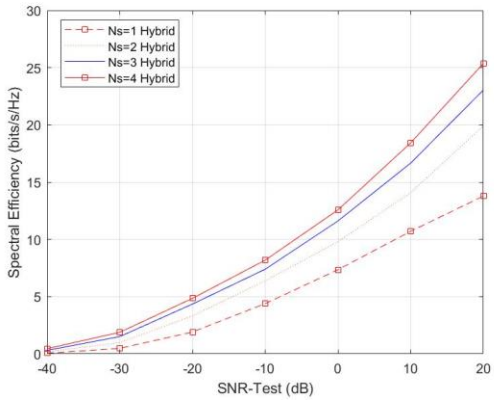
F. Numerical Simulations

This section provides a numerical evaluation of the DL framework's performance. The different numbers of data streams per user is compared using the proposed DL approach. Throughout the simulations, massive MIMO system with $N_{tRF} = N_{rRF} = 4$ RF chains for $N_T = 64$ and $N_R = 16$ antennas are considered. The antennas are deployed with half wavelength spacing at carrier frequencies $f_c = 28\,\text{GHz}$, $f_c = 38\,\text{GHz}$, $f_c = 60\,\text{GHz}$ and $f_c = 73\,\text{GHz}$. The examination of these carrier frequencies aids in exposing the key characteristics of the new frequency bands, as well as their maximum coverage distance, usage visibility, and mmWave system-wide behavior. Unless stated otherwise, there are L = 5 clusters of all transmit and receive paths which are uniform randomly selected from the interval $\{F, q\} \hat{I} [-p, p]$ with angular spread of 5°. In preamble stage, the transmitter emits beams by using RF chains while, at the receiver, all of the RF chains are active. The transmit and received beams are formed by selecting $\bar{\mathbf{F}}$ and $\bar{\mathbf{W}}$ as $N_T \times N_T$ and $N_R \times N_R$ DFT matrices respectively. The preamble data are distinct from the training stage in the prediction step.

In this section, the simulations will be presented to show the effects of the data streams (1, 2, 3 and 4). Figure 6 to 9 show the spectral efficiency of various data streams for varying SNR-Test, given SNR = 20dB and 0dB. The OMP technique serves as an upper bound on the performances of the deep learning approaches since it is utilized to acquire the labels of the deep networks for hybrid beamforming. But even the benchmark OMP algorithm requires flawless channel information. The corruptions in the DL input, which lead to variations from the label data (obtained by OMP) at the output regression layer, account for the discrepancy between the OMP algorithm and the DL frameworks.

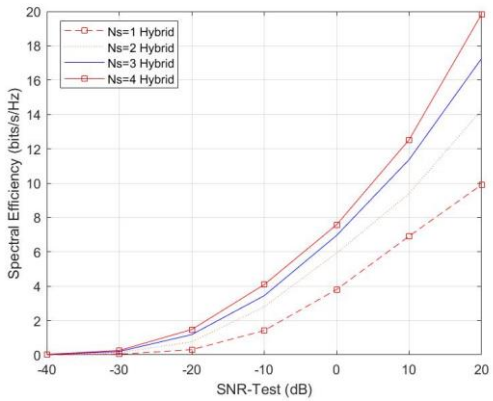


(a) SNR = 20dB

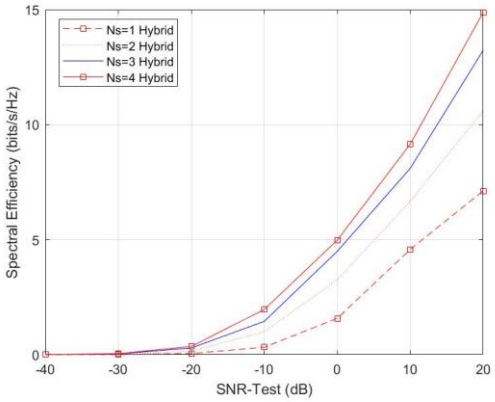


(b) SNR = 0dB

Figure 6. Spectral efficiency of different number of data streams for varying SNR-Test with $N_T = 64$, $N_R = 16$, $N_{tRF} = 4$ at $f_c = 28\,GHz$



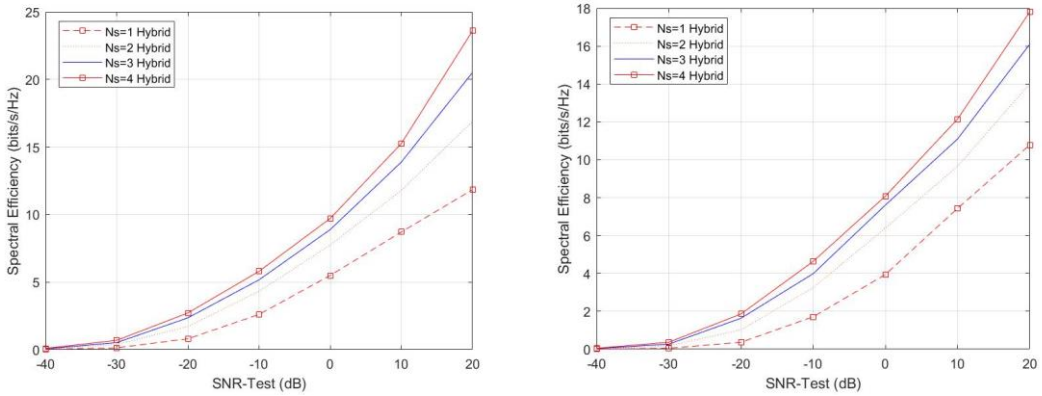
(a) SNR = 20dB



(b) SNR = 0dB

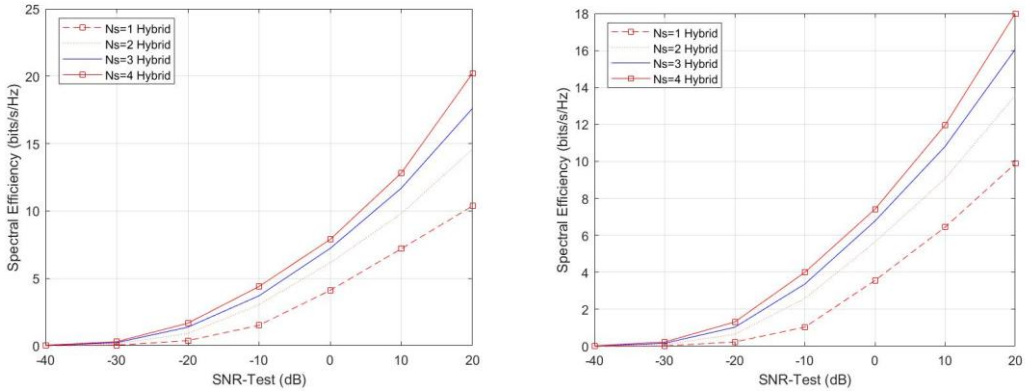
Figure 7. Spectral efficiency of different number of data streams for varying SNR-Test with $N_T = 64$, $N_R = 16$, $N_{tRF} = 4$ at $f_c = 38\,GHz$

Figure 6 shows the spectral efficiency versus SNR-Test when SNR = 20dB and SNR = 0dB at the carrier frequency 28GHz. Figure 7 shows the spectral efficiency versus SNR-Test when SNR = 20dB and SNR = 0dB at the carrier frequency 38GHz. Figure 8 and 9 show the simulation results at the carrier frequency 60GHz and 73GHz respectively. The number of data stream $N_s = 4$ achieve higher spectral efficiency than the number of data stream $N_s = 1$ at all carrier frequency. The signal to noise ratio SNR = 20 dB at all carrier frequency achieve higher spectral efficiency than SNR = 0dB. These Figures 6 to 9 show that the spectral efficiency improves significantly when the number of data streams increase as the SNR-Test values increases too. The spectral efficiency of approximately 38 bit/s/Hz at carrier frequency $f_c = 28\,GHz$ for the number of data stream $N_s = 4$ is the highest achievable value among other carrier frequency. In addition, the hybrid beamforming can perform close to what optimal weights can offer using less hardware.



(a) SNR = 20dB (b) SNR = 0dB

Figure 8. Spectral efficiency of different number of data streams for varying SNR-Test with $N_T = 64$, $N_R = 16$, $N_{tRF} = 4$ at $f_c = 60$ GHz



(a) SNR = 20dB (b) SNR = 0dB

Figure 9. Spectral efficiency of different number of data streams for varying SNR-Test with $N_T = 64$, $N_R = 16$, $N_{tRF} = 4$ at $f_c = 73$ GHz

The robustness of the approach is examined in relation to the estimated channel data based on all of the simulation results. Therefore, when the received pilot data is tainted by noise identified by SNR-Test, the channel matrix predicted by LSTM is used in the beamforming method. Since SNR-Test only uses estimated channel data during the combiner design stage, its noise only impacts the performance of the deep learning approach's combiner design (and not the precoder design). From Figures 6 to 9 that all of the number of data streams reach their maximum performance after SNR-Test ≥ 20dB. In particular, the number of data streams $N_s = 4$ has more robust performance than the other number of data stream in all figures from Figures 6 to 9. This observation states that the algorithm requires at least approximately SNR-Test = 20dB noise level for sufficient channel estimate in setting.

G. Conclusion

For massive MIMO systems, hybrid beamforming is a crucial technique that enables fewer RF chains and, consequently, higher system spectral efficiency. However, the estimation of the channel imposes significant signaling overhead,

particularly in communications, and the design of digital and analog precoders is difficult. By leveraging the proposed deep learning based LSTM networks can improve the spectral efficiency of the communication system. When the system can be trained using simply spatial statistical channels and not have to pay for the expensive process of obtaining optimal solutions, this deep learning approach reduces both training time and cost. In wireless communications, especially in massive MIMO systems where transmission occurs over time-varying channels, LSTM networks can effectively model the temporal dynamics of the spatial statistical channel. LSTM networks are well-suited for processing such sequenced data, as they can maintain and update a memory state over time, allowing them to learn pattern and correlations in the sequential data. Furthermore, it was shown that the sequence to sequence method is a useful technique for time series prediction. The results show that the BS can be trained to achieve the near-optimal hybrid precoder and robustly construct the hybrid beamforming, even without access to the complete CSI. Several numerical simulations showed that deep learning algorithms perform better and offer improved spectral efficiency. Training the deep networks for multiple distinct channel conditions, each of which is tainted by artificial noise, yields a strong performance. Furthermore, the suggested approach is applicable to real-time systems. It is anticipated that the first 5G network rollout will make use of the frequencies under investigation.

H. Acknowledgment

The authors would like to acknowledge Department of Electronic Engineering at Mandalay Technological University (MTU) for providing essential resources.

I. References

[1] T. Kebede, Y. Wondie, J. Steinbrunn, H. B. Kassa, and K. T. Kornegay, "Precoding and Beamforming Techniques in mmWave-Massive MIMO: Performance Assessment," IEEE Access, vol. 10, pp. 16365–16387, 2022, doi: 10.1109/ACCESS.2022.3149301.

[2] J. Zhang, X. Yu, and K. B. Letaief, "Hybrid beamforming for 5G and beyond millimeter-wave systems: A holistic view," IEEE Open Journal of the Communications Society, vol. 1, pp. 77–91, 2020, doi: 10.1109/OJCOMS.2019.2959595.

[3] M. Shafi et al., "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," IEEE Journal on Selected Areas in Communications, vol. 35, no. 6, pp. 1201–1221, Jun. 2017, doi: 10.1109/JSAC.2017.2692307.

[4] I. Ahmed et al., "A survey on hybrid beamforming techniques in 5G: Architecture and system model perspectives," IEEE Communications Surveys and Tutorials, vol. 20, no. 4, pp. 3060–3097, Oct. 2018, doi: 10.1109/COMST.2018.2843719.

[5] A. M. Elbir, "A Deep Learning Framework for Hybrid Beamforming Without Instantaneous CSI Feedback," Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.10971

[6] R. Hu, L. Jiang, and P. Li, "Hybrid Beamforming with Deep Learning for Large-Scale Antenna Arrays," IEEE Access, vol. 9, pp. 54690–54699, 2021, doi: 10.1109/ACCESS.2021.3069037.
[7] R. Dilli, "Hybrid Beamforming in 5G NR Networks Using Multi User Massive MIMO at FR2 Frequency Bands," Wireless Personal Communications, vol. 127, no. 4, pp. 3677–3709, Dec. 2022, doi: 10.1007/s11277-022-09952-z.
[8] S. Hamid et al., "Hybrid Beamforming in Massive MIMO for Next-Generation Communication Technology," Sensors, vol. 23, no. 16, Aug. 2023, doi: 10.3390/s23167294.
[9] J. Guo, C.-K. Wen, and S. Jin, "Deep Learning-Based CSI Feedback for Beamforming in Single- and Multi-cell Massive MIMO Systems," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.06099
[10] Z. Gao et al., "Data-Driven Deep Learning Based Hybrid Beamforming for Aerial Massive MIMO-OFDM Systems with Implicit CSI," Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.06778
[11] J. Du, W. Xu, C. Zhao, and L. Vandendorpe, "Weighted Spectral Efficiency Optimization for Hybrid Beamforming in Multiuser Massive MIMO-OFDM Systems," Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.12255
[12] P. Dong, H. Zhang, and G. Y. Li, "Framework on Deep Learning-Based Joint Hybrid Processing for mmWave Massive MIMO Systems," IEEE Access, vol. 8, pp. 106023–106035, 2020, doi: 10.1109/ACCESS.2020.3000601.
[13] T. Peken, S. Adiga, R. Tandon, and T. Bose, "Deep Learning for SVD and Hybrid Beamforming," IEEE Transactions on Wireless Communications, vol. 19, no. 10, pp. 6621–6642, Oct. 2020, doi: 10.1109/TWC.2020.3004386.
[14] M. Majumder, H. Saxena, S. Srivastava, and A. K. Jagannatham, "Optimal Bit Allocation-Based Hybrid Precoder-Combiner Design Techniques for mmWave MIMO-OFDM Systems," IEEE Access, vol. 9, pp. 54109–54125, 2021, doi: 10.1109/ACCESS.2021.3070921.
[15] B. Y. Chen, Y. F. Chen, and S. M. Tseng, "Hybrid Beamforming and Data Stream Allocation Algorithms for Power Minimization in Multi-User Massive MIMO-OFDM Systems," IEEE Access, vol. 10, pp. 101898–101912, 2022, doi: 10.1109/ACCESS.2022.3208704.
[16] Y. Liu, Q. Zhang, X. He, X. Lei, Y. Zhang, and T. Qiu, "Spectral-efficient hybrid precoding for multi-antenna multi-user mmWave massive MIMO systems with low complexity," Eurasip Journal on Wireless Communications and Networking, vol. 2022, no. 1, Dec. 2022, doi: 10.1186/s13638-022-02150-2.
[17] D. Zhang, Y. Wang, X. Li, and W. Xiang, "Hybrid beamforming for downlink multiuser millimetre wave MIMO-OFDM systems," IET Communications, vol. 13, no. 11, pp. 1557–1564, Jul. 2019, doi: 10.1049/iet-com.2018.6061.
[18] Z. Wang, M. Li, Q. Liu, and A. L. Swindlehurst, "Hybrid Precoder and Combiner Design with Low-Resolution Phase Shifters in mmWave MIMO Systems," IEEE Journal on Selected Topics in Signal Processing, vol. 12, no. 2, pp. 256–269, May 2018, doi: 10.1109/JSTSP.2018.2819129.
[19] J. Chen et al., "Hybrid Beamforming/Combining for Millimeter Wave MIMO: A Machine Learning Approach," IEEE Transactions on Vehicular Technology, vol. 69, no. 10, pp. 11353–11368, Oct. 2020, doi: 10.1109/TVT.2020.3009746.

[20] S. Lavdas, P. K. Gkonis, E. Tsaknaki, L. Sarakis, P. Trakadas, and K. Papadopoulos, "A Deep Learning Framework for Adaptive Beamforming in Massive MIMO Millimeter Wave 5G Multicellular Networks," Electronics (Switzerland), vol. 12, no. 17, Sep. 2023, doi: 10.3390/electronics12173555.
[21] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep Learning for Channel Estimation: Interpretation, Performance, and Comparison," IEEE Transactions on Wireless Communications, vol. 20, no. 4, pp. 2398–2412, Apr. 2021, doi: 10.1109/TWC.2020.3042074.
[22] A. K. Gizzini and M. Chafii, "A Survey on Deep Learning based Channel Estimation in Doubly Dispersive Environments," Jun. 2022, [Online]. Available: http://arxiv.org/abs/2206.02165
[23] M. Wang, A. Wang, Z. Liu, and J. Chai, "Deep learning based channel estimation method for mine OFDM system," Scientific Reports, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-43971-5.
[24] H. Hirose, T. Ohtsuki, and G. Gui, "Deep Learning-Based Channel Estimation for Massive MIMO Systems with Pilot Contamination," IEEE Open Journal of Vehicular Technology, vol. 2, pp. 67–77, 2021, doi: 10.1109/OJVT.2020.3045470.
[25] Y. Huang, C. Liu, Y. Song, and X. Yu, "DFT codebook-based hybrid precoding for multiuser mmWave massive MIMO systems," Eurasip Journal on Advances in Signal Processing, vol. 2020, no. 1, Dec. 2020, doi: 10.1186/s13634-020-00669-4.
[26] Z. Xiang, J. Yan, and I. Demir, "A Rainfall-Runoff Model With LSTM-Based Sequence-to-Sequence Learning," Water Resources Research, vol. 56, no. 1, Jan. 2020, doi: 10.1029/2019WR025326.