

Article

Implementation of Deep-Learning-Based CSI Feedback Reporting on 5G NR-Compliant Link-Level Simulator †

Daniel Gaetano Riviello ^{1,*} , Riccardo Tuninato ² , Elisa Zimaglia ³, Roberto Fantini ³ and Roberto Garello ² 

¹ Department of Electrical, Electronic, and Information Engineering, University of Bologna, 40136 Bologna, Italy

² Department of Electronics and Telecommunications (DET), Politecnico di Torino, 10129 Torino, Italy; riccardo.tuninato@polito.it (R.T.); roberto.garello@polito.it (R.G.)

³ TIM S.p.A., 10148 Torino, Italy; elisa.zimaglia@telecomitalia.it (E.Z.); roberto.fantini@telecomitalia.it (R.F.)

* Correspondence: daniel.riviello@unibo.it

† This manuscript is an extended version of the conference paper: Zimaglia, E.; Riviello, D.G.; Garello, R.; Fantini, R. A Novel Deep Learning Approach to CSI Feedback Reporting for NR 5G Cellular Systems. In Proceedings of the 2020 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW), Riga, Latvia, 1–2 October 2020.

Abstract: Advances in machine learning have widened the range of its applications in many fields. In particular, deep learning has attracted much interest for its ability to provide solutions where the derivation of a rigorous mathematical model of the problem is troublesome. Our interest was drawn to the application of deep learning for channel state information feedback reporting, a crucial problem in frequency division duplexing (FDD) 5G networks, where knowledge of the channel characteristics is fundamental to exploiting the full potential of multiple-input multiple-output (MIMO) systems. We designed a framework adopting a 5G New Radio convolutional neural network, called NR-CsiNet, with the aim of compressing the channel matrix experienced by the user at the receiver side and then reconstructing it at the transmitter side. In contrast to similar solutions, our framework is based on a 5G New Radio fully compliant simulator, thus implementing a channel generator based on the latest 3GPP 3-D channel model. Moreover, realistic 5G scenarios are considered by including multi-receiving antenna schemes and noisy downlink channel estimation. Simulations were carried out to analyze and compare the performance with current feedback reporting schemes, showing promising results for this approach from the point of view of the block error rate and throughput of the 5G data channel.

Keywords: 5G; New Radio; deep learning; convolutional neural network; CSI reporting



Citation: Riviello, D.G.; Tuninato, R.; Zimaglia, E.; Fantini, R.; Garello, R. Implementation of Deep-Learning-Based CSI Feedback Reporting on 5G NR-Compliant Link-Level Simulator. *Sensors* **2023**, *23*, 910. <https://doi.org/10.3390/s23020910>

Academic Editors: Giuseppe Caso, Anna Brunstrom, Özgü Alay, Harilaos Koumaras, Almudena Díaz Zayas and Valerio Frascolla

Received: 20 November 2022

Revised: 2 January 2023

Accepted: 7 January 2023

Published: 12 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

New Radio (NR) wireless networks are anticipated to support very high data rates and new applications that will create opportunities for a radically new radio technology paradigm [1]. In this context, machine learning (ML) has regained attention for its achievements especially in the upper layers, such as in cognitive radio [2,3], self-organizing networks (SONs) [4] and resource management [5,6]. However, the authors of [7] claim that the potential use of deep learning (DL) to the physical layer will be increasingly recognized. This appears to be reasonable considering the new features of next-generation communications, such as complex scenarios, with a consequent lack of information about the channel model, high data rates and accurate processing requirements [7].

Studies on ML applications to the physical layer have been carried out, for example, for modulation recognition [8,9], signal classification [10], multiple input multiple output (MIMO) detection [11], channel state information (CSI) feedback reduction [12,13], non-orthogonal multiple access (NOMA) [14], hybrid precoding [11,15], encoding and

decoding [16], and channel estimation and equalization [17–20]. Researchers have presented a range of arguments to support the adoption of DL techniques in contemporary communication environments [7,11,16]:

- deep networks are very good and adaptable function approximators—they learn the weights of the model optimizing the overall performance through a training process, instead of requiring a rigorous mathematical description that could be hard to extract in complex scenarios;
- DL-based systems could replace manual feature extractors and learning features automatically, flexibly adapting the structure and the parameters of the model, with considerable improvement to the end-to-end performance;
- DL-based algorithms work very well with large amounts of data due to their intrinsic parallel and distributed computing, leading to speeding up of computation;
- DL models can overcome the block structure to provide a complete view of the system and optimize it end-to-end.

This paper proposes a DL-based solution for CSI feedback improvement in 5G-based communication systems. The availability of information about the colorful characteristics of the communication channels has a fundamental impact on many transmission features in most modern radio-access technologies [21]. This framework of information can vary from a rough estimate of the radio link path loss, which is useful for adjusting the transmitted power, to very precise knowledge of the channel amplitude and phase [21]. Measurements and estimates can be performed by either the transmitter or the receiver side. Focusing on the downlink case, the knowledge of the channel characteristics can be acquired by means of device measurements or by the network itself, depending on the particular scenario:

- when working in frequency division duplexing (FDD) mode, downlink and uplink could potentially be significantly different. In consequence, to obtain reliable information about the downlink channel, measurements must necessarily be acquired by the user device and then reported to the network; the network will exploit this feedback to properly set the transmission parameters for future downlink transmissions;
- in time-division duplexing (TDD) communications, when downlink and uplink transmissions experience the same channel characteristics, feedback from the device is not necessary, since the network itself can obtain useful information about the downlink features of interest by measuring them in the uplink transmissions.

In general, the evaluation of the radio environment is referred to as channel sounding and requires the adoption of specific reference signals (RS) on which the receiver can perform measurements.

The adoption of appropriate channel modelling is critical when performing simulations involving signal processing, such as channel sensing. Several channel models have been reported in the literature, with two main approaches described: stochastic and statistical. The former relies on the definition of the probability density function (PDF) of the channel parameters depending on the geometry of the environment, while the latter exploits real measures on which to base the definition of the channel. Each of these solutions has possible drawbacks, but a hybrid approach can be chosen, firstly, stochastically, characterizing the channel parameters through their PDF, and subsequently leveraging real data to confirm and eventually adapt them. In this investigation, we rely on the latest 3D stochastic channel model for 5G mmWave massive MIMO communications released by 3GPP in TR 38.901 [22,23]. It covers the entire range of 0.5–100 GHz, with particular relevance to the new challenges represented by mmWave communications, where some phenomena, such as path attenuation, noise, interference, and atmospheric attenuation, can seriously impact the signal transmission. One of its main characteristics is the modelling of the channel through a certain number of clusters distributed over the area, which groups the scatterers and so produces different replicas of the propagated signal (i.e., different rays). It is categorised as a clustered delay line (CDL) model. Moreover, while, in 4G, the propagation waves were considered only in the horizontal plane, thus produc-

ing a 2D channel model, mmWave communications (and thus 5G) require a 3D channel model, characterising the signal propagation depending on the elevation coordinate. The channel model generation is performed inside a link-level simulator designed by TIM, which also provides the resource blocks generation, transmission and reception for a 5G 3GPP-compliant communication, with evaluation of the primary downlink shared channel (PDSCH) throughput and the block error rate (BER).

1.1. Literature Review

Several studies, such as [12,13,24,25], suggest that, with the introduction of massive MIMO technologies in 5G systems, the potential gain from an accurate estimation of channel characteristics will considerably increase, with channel sounding becoming a critical issue. Moreover, as the authors of [12,13] point out, the most-used approaches to reduce the feedback overhead, based on codebooks or vector quantization, process feedback quantities whose dimensions scale with the number of antennas, making these techniques unsuitable in massive MIMO scenarios. These criticisms have stimulated new studies aimed at overcoming the difficulties associated with traditional approaches with DL-based solutions.

The authors of [12] designed an autoencoder based on neural networks to compress and then reconstruct the channel state information, called CsiNet. This paper stimulated significant interest among researchers who sought to expand the applicability and improve the performance. In [26], the authors present two different structures, ConvCsiNet and ShuffleCsiNet, both able to outperform CsiNet from an NMSE point of view. Even though the latter has reduced complexity with respect to the first structure, it is still more complex than CsiNet. One of the most recent advancements in DL is the concept of attention, which is exploited in [27,28]. In the former, long short-term memory (LSTM) is adopted to learn the temporal correlation of channels, while an attention mechanism is developed to perceive local information and automatically weight feature information. This kind of solution can further improve the reconstruction capabilities, but comes at a higher cost that could not be supported in terms of user equipment. A solution to effectively decrease the complexity of CsiNet was proposed in [29], which identified a lighter, fully connected (FC) layer as the key to encoder compression. The network parameters are transformed from a 32-bit floating point number to a 1-bit binary number, optimizing a positive scale factor that minimises the distance between standard weights and binary weights. In [30], a DL-based CSI feedback framework is described for beam-forming (BF) design, CsiBFnet. Its goal is to maximize the BF performance gain rather than the feedback accuracy, including with respect to single-cell or multi-cell scenarios. Another interesting solution is the CQNet described in [31], which focuses on the impact of quantisation on the channel estimation coefficients and exploits radial coordinates to improve quantisation scheme efficiency.

1.2. Paper Contribution and Organization

The solutions presented in the papers referred to above increase the possibilities of DL-based solutions for CSI; however, there is a lack of studies focusing on the characteristics of a system based on 5G NR and, therefore, the channel model and performance in terms of the BLER of the data channel. In this article, we extend our findings in [32] and propose a DL-based technique for channel state information feedback reporting in FDD transmission mode, based on the implementation of a convolutional neural network, called NR-CsiNet. The improvements in our approach with respect to [12] can be summarized in terms of the following new features:

1. provision of MIMO capabilities through the use of antenna arrays at both the transmitter and receiver sides;
2. addressing the estimation of the downlink channel affected by noise;
3. testing the system with a 5G NR fully compliant simulator with a clustered delay line channel model.

The remainder of this paper is organized as follows: Section 2 introduces CSI-RS, which are reference signals used for downlink channel sounding. Section 3 provides a description of the New Radio link simulator software developed for all our experiments in order to outline the simulation environment. Section 4 presents the NR-CsiNet DL model, describing its internal structure and the data-processing steps required. The simulation results are presented in Section 5. Section 6 concludes the article with some observations and suggestions for potential future research directions.

2. Channel State Information Reference Signals

One of the key design principles of NR is to avoid, as far as possible, “always-on” signals [21,33]. In release 8 of the LTE standard, channel sounding for downlink transmissions is performed by means of device measurements on cell-specific reference signals (CRS) [34], which are transmitted over the whole transmission bandwidth in each subframe. From release 10 on [35], these reference signals are complemented by another type of reference signal called CSI-RS, which, in contrast to the first, is not expected to be continuously transmitted. The main reason for this upgrade lays in the necessity to support spatial multiplexing with more than four layers; however, as [21] points out, CSI-RS introduction has opened the way to further technological extensions, such as coordinated multi-point (CoMP) [36] operations and interference estimation.

In 3GPP specifications [37], the channel quality indicator (CQI), rank indicator (RI), precoding matrix indicator (PMI), and several other quantities, are jointly referred to as channel state information (CSI). The precoding matrix indicator is reported by the user equipment (UE) to the base station (BS); it represents an indication of what the user considers to be a suitable precoding matrix to be applied on downlink transmissions. In particular, the PMI is no more than a set of indexes or a single index pointing to a specific entry in the precoding codebook, which contains a list of all the possible precoding matrices \mathbf{W} that the device can select and report to the network. There exists at least one codebook for each permitted combination of N_T (number of antenna ports) and N_L (number of layers) associated with the configured CSI-RS.

It is important to stress that precoding codebooks play a role only in the context of PMI reporting and do not impose any restriction in the choice of the precoding matrix to be applied by the network [21]. In some cases, it is convenient for the network to select the precoding matrix suggested by the device, but, when, for example, the network is in possession of additional information, the precoding computation may lead to a different choice.

As [21] reports, the typical use case of multi-antenna precoding is multi-user MIMO (MU-MIMO) [38–40]. In this case, the main purposes are to direct the energy towards the device and simultaneously limit the interference towards the other users scheduled on the same time-frequency resource. In this kind of scenario, it is clear that the network needs more detailed information about the channel experienced by different UEs so that it can take into account the PMI reported by all simultaneously scheduled devices when selecting the precoding matrix [21].

The NR standard defines two types of CSI reporting, each characterized by a different size and structure of the codebook [37]:

- Type I CSI codebooks: designed for single user-MIMO (SU-MIMO) scenarios, where a single user is scheduled within a certain time-frequency resource, transmitting on a potentially large number of layers in parallel;
- Type II CSI codebooks: designed for MU-MIMO scenarios, where many users are scheduled within the same time-frequency resource, each transmitting on a limited number of layers in parallel (two at most).

For fast time-varying wireless channels, the adoption of delayed CSI at the transmitter can help reduce the feedback frequency of the channel estimates [41].

3. New Radio Link-Level Simulator

New Radio link simulator software was used for all the experiments conducted and described in this paper. The simulator was developed in Telecom Italia laboratories with an engine implemented in MATLAB®. The most critical blocks in terms of execution speed were, instead, implemented in C language and linked to the MATLAB engine via MEX. The purpose was to model a radio interface which was fully compliant with 3GPP specification and to evaluate the link-level performances of 5G-based point-to-point communications. The simulator is a link-level simulator and targets scenarios with a single next-generation base station or gNodeB and a single UE for a SU-MIMO scenario where the UE can deploy an antenna array.

The various blocks of the simulator provide a bit-level model of the 5G NR physical layer, as described in the 38 series of the 3GPP specifications [37,42–44]. The physical downlink shared channel (PDSCH) transmission chain can be represented as the block scheme of Figure 1, which also reflects the 5G NR simulator structure. The red blocks are those which exploit the C language to speed-up the computations.

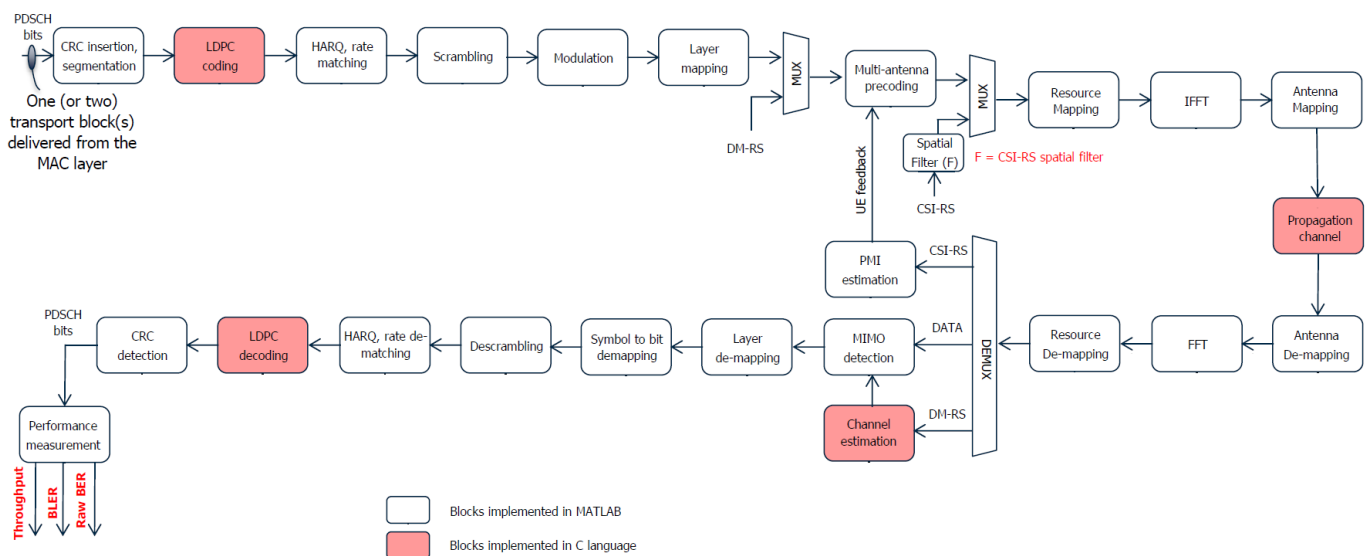


Figure 1. New Radio link-level simulator scheme.

3.1. Transmission Chain

A detailed description of the main blocks is provided as follows.

Transport Block generator Data are generated as a sequence of bits. Depending on the transport size (A) and the modulation and coding scheme (MCS), in particular, on the code rate (R), the LPDC base graph selection is performed:

- if $A \leq 292$, or if $A \leq 3824$ and $R \leq 0.67$, or if $R \leq 0.25$, LDPC base graph 2 is used;
- otherwise, LDPC base graph 1 is used.

Input bits are then segmented into different blocks; the CRC parity bits are calculated on and attached to each block. Depending on the LPDC base graph, the maximum dimension of blocks can change. The lowest order information bit is mapped to the most significant bit of the transport block (i.e., parity bits are found at the end of the block).

LDPC encoder LDPC coding is applied to the bit sequence for a given code block. N encoded bits are obtained as output, where $N = 66 Z_c$ for LDPC base graph 1 and $N = 50 Z_c$ for LDPC base graph 2 (Z_c value can be found in subclause 5.2.2 of TR 38.212 [42]).

Rate-matching block A rate-matching block is required to adapt the size of the transport block after LDPC coding to the size of the allocated resources that can be used for PDSCH transmission. PDSCH coded bits must be rate-matched around the resource elements occupied by the CSI-RS and other reference signals, otherwise the coding rate may exceed the unit value (i.e., the REs in the slot available for PDSCH are not sufficient to transmit the information bits). In practice, it is expected that the gNB scales down the MCS in the slots where the CSI-RS transmission causes an LDPC coding rate higher than about 0.95. This approach is also modelled in the link simulator. The throughput calculation must consider this effect, re-scaling the formula accordingly.

Scrambling operation Before modulation, the codewords (up to two) are scrambled, as in subclause 7.3.1 of [42], resulting in a block of scrambled bits.

QAM modulation The block of scrambled bits for each codeword are QAM-modulated using one of the modulation schemes (selected by the MCS) from table 7.3.1.2-1 [42], resulting in a block of complex-valued modulation symbols.

Layer and subcarrier mapping The modulated symbols are mapped to one or more layers (if multi-layer transmission is enabled), by splitting the codewords (one or two) among them. Then, the symbols are inserted into the actual resource elements of the physical resource block, through a predefined look-up table, based on the 3GPP resource allocation procedure.

MIMO processing The precoding matrix is applied to the PDSCH channel, i.e., sub-carriers containing data. If PMI reporting is enabled, the precoding matrix F_j is selected by the transmitter at the gNB as the j -th element of the codebook, where j is the PMI index reported by the UE. The digital beam-forming scheme implemented in the simulator is wideband; that is, the same beam-forming matrix F_j is applied to all the PRBs. If, instead, the channel estimate is given through feedback by the UE, (e.g., SRS transmission in uplink), downlink beam-forming schemes based on the channel matrix can be applied exploiting the downlink/uplink channel reciprocity.

CSI-RS mapping The transmission of CSI-RS signals in the link-level simulator is modelled in a realistic way. The CSI-RS signals are generated and mapped in the time frequency-grid and configuration parameters set the number of ports and the transmission period, denoted as *CSI-RS period*, which can take the set of values defined in the 3GPP TS 38.331 [45] specifications. The CSI-RS signals are transmitted once every [4, 5, 8, 10, 16, 20, 32, 40, 64, 80, 160, 320, 640] slots.

OFDM modulator The OFDM modulation is performed through the IFFT Operation. Before the IFFT, an IFFT shift must be applied to properly translate the spectrum into the positive frequency domain. The IFFT must be applied on an OFDM symbol-basis. The channel bandwidth and sub-carrier spacing (SCS) determine the size of the IFFT and the number of samples for each OFDM symbol. At this point a cyclic prefix is added at the beginning of each OFDM symbol, through a look-up table, with a duration selected between “normal” or “extended”. The “normal” option is selected for the simulations according to the used sub-carrier spacing of 30 kHz. This value is compatible with the maximum delay of the channel that falls into the selected CP length.

Oversampling and filtering An oversampling step increases the number of samples for each symbol by a selected factor and the sampling frequency is increased accordingly. A raised cosine filter is used as the transmitting filter.

Receiver side At the receiver side, a finite impulse response filter is applied to remove the undesired frequencies and a down-sampler stage re-samples the signal to the normal frequency, by the same factor as the up-sampler. To account for the delay due to the signal transmission and FIR filter, at each iteration, the signal at the receiver side is buffered and processed on a block basis, where the block length in time is

equal to the slot duration. The receiver must then reverse all the signal-processing applied by the transmitter with suitable blocks, such as an OFDM signal demodulator, a sub-carrier demapper, and so on. The MIMO processing consists of the application of a combining matrix over the received symbols to exploit the antenna array at the receiver if more than one antenna is configured. The combining matrix can be used for various purposes, i.e., improving the SINR or for separating the different layers if more than one data stream is transmitted using spatial multiplexing. The LDPC decoder applies the offset min-sum (OMS) decoding algorithm. The performance measurements are applied on the CRC detection for the BLER, while the raw BER is computed on the number of actual erroneous bits before the LDPC decoding. The approximate peak throughput of the simulated NR configuration can be calculated by means of the formula provided in the TS 38.306 3GPP [46] specification:

$$\text{data rate [Mbps]} = 10^{-6} \cdot \sum_{j=1}^J \left(v_L^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{max} \cdot \frac{N_{PRB}^{BW(j),\mu} \cdot 12}{T_s^\mu} \cdot (1 - OH^{(j)}) \right) \quad (1)$$

where:

- J is the number of aggregated component carriers in a band or band combination.
- $R_{max} = 948/1024$ is the maximum LDPC code rate.
- $v_L^{(j)}$ is the maximum number of layers.
- $Q_m^{(j)}$ is the maximum modulation order
- $f^{(j)}$ is the scaling factor (it can take the values 1, 0.8, 0.75 and 0.4) and is signalled per band or per band combination).
- μ is the numerology.
- T_s^μ is the average OFDM symbol duration in a subframe for numerology μ and normal CP, i.e., $T_s^\mu = \frac{10^{-3}}{14 \cdot 2^\mu}$.
- $N_{PRB}^{BW(j),\mu}$ is the maximum RB allocation in maximum UE supported bandwidth $BW^{(j)}$ with numerology μ .
- $OH^{(j)}$ is the overhead (0.14 for DL FR1-0.18 for DL FR2-0.08 for UL FR1-0.10 for UL FR2).

3.2. Channel Model Generation

The key element of this simulation framework is clearly the fading channel block: for an NR-compliant link simulator, the 3GPP has two defined models for NR link and system evaluation purposes [22,47]:

- Tapped delay line (TDL): for this channel model, the correlation between different antennas is defined statically by a correlation matrix. The TDL model is based on the description of the impulse response of the channel;
- Clustered delay line (CDL): with this channel model the direction of the signal in the space is modelled. The model is based on the description of the main departure and arrival directions of the signal in the space and the number of clusters corresponds to the number of channel reflections.

The CDL channel model is adopted for all the simulations presented in this article due to its importance for practical applications. In particular, we select the CDL-B profile. This CDL profile is specific for non-line-of-sight (NLOS) transmissions and can be used for sub 6 GHz frequency, where line-of-sight (LOS) is not mandatory. A characterising property is that the first path in time is also the strongest one.

The channel between the user and the BS depends on the environment geometry. In our scenario, the user is placed at a distance of 100 m from the BS, with a height of 1.5 m, while the BS has a height of 10 m. The UE direction could be defined as DOMINANT_PATH (the same direction of the main azimuth of arrival (AoA) direction), RANDOMLY (uniformly generated around $[-180^\circ, +180^\circ]$) and ARBITRARY (statically selecting a UE direction). The delay spread can be selected among the set ('VERY SHORT', 'SHORT', 'NOMINAL',

‘LONG’, ‘VERY_LONG’). NOMINAL is fixed for our set of simulations, which corresponds to 100 ns. The coherence time of the channel depends on the Doppler spread and on the user speed, which is fixed at 3 km/h (i.e., a walking pedestrian). The CDL profile channel requires the generation of N_c clusters. Each cluster n will be assigned a different delay (depending on the delay spread times), a set of angular orientations, including AoA, azimuth of departure (AoD), zenith of departure (ZoD) and zenith of arrival (ZoA), and a power gain P_n . The azimuth angle spread of departure and arrival (ASD and ASA, respectively) and the zenith angle spread of departure and arrival (ZSD and ZSA, respectively), characterize the entire set of clusters. These values are used for the generation of the M rays that form each cluster. The direction of a certain ray is given by the angular orientation of its own cluster, the angular spreads and a different coefficient for each ray. The angle for azimuth is ϕ , while the angle for zenith is θ . At both the gNB and UE side, three levels of antenna correlation are possible: LOW, MEDIUM and HIGH. The level of correlation is inversely related to the angle spread (i.e., larger angle spreads correspond to lower correlation and vice versa). The CDL model for cluster n , transmitter antenna element u and receiver antenna element s is, thus, a combination of M NLOS rays and a LOS ray when the UE is in a LOS condition:

$$\mathbf{H}_{u,s,n}^{\text{NLOS}} = \sqrt{\frac{P_n}{M}} \sum_{m=1}^M \begin{bmatrix} F_{rx,u,\theta}(\theta_{n,m,ZoA}, \phi_{n,m,AoA}) \\ F_{rx,u,\phi}(\theta_{n,m,ZoA}, \phi_{n,m,AoA}) \end{bmatrix}^T \begin{bmatrix} e^{(j\Phi_{n,m}^{\theta\theta})} & \sqrt{\kappa_{n,m}^{-1}} e^{(j\Phi_{n,m}^{\theta\phi})} \\ \sqrt{\kappa_{n,m}^{-1}} e^{(j\Phi_{n,m}^{\phi\theta})} & e^{(j\Phi_{n,m}^{\phi\phi})} \end{bmatrix} \begin{bmatrix} F_{tx,s,\theta}(\theta_{n,m,ZoD}, \phi_{n,m,AoD}) \\ F_{tx,s,\phi}(\theta_{n,m,ZoD}, \phi_{n,m,AoD}) \end{bmatrix} \exp\left(\frac{j2\pi\hat{r}_{tx,n,m}^T \cdot \vec{d}_{rx,u}}{\lambda_0}\right) \exp\left(\frac{j2\pi\hat{r}_{tx,n,m}^T \cdot \vec{d}_{tx,s}}{\lambda_0}\right) \exp\left(\frac{j2\pi\hat{r}_{tx,n,m}^T \cdot \vec{v}}{\lambda_0}\right) \quad (2)$$

$$\mathbf{H}_{u,s}^{\text{LOS}} = \begin{bmatrix} F_{rx,u,\theta}(\theta_{LOS,ZoA}, \phi_{LOS,AoA}) \\ F_{rx,u,\phi}(\theta_{LOS,ZoA}, \phi_{LOS,AoA}) \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} F_{tx,s,\theta}(\theta_{LOS,ZoD}, \phi_{LOS,AoD}) \\ F_{tx,s,\phi}(\theta_{LOS,ZoD}, \phi_{LOS,AoD}) \end{bmatrix} \exp\left(\frac{-j2\pi\vec{d}_{3D}}{\lambda_0}\right) \exp\left(\frac{j2\pi\hat{r}_{tx,LOS}^T \cdot \vec{d}_{rx,u}}{\lambda_0}\right) \exp\left(\frac{j2\pi\hat{r}_{tx,LOS}^T \cdot \vec{d}_{tx,s}}{\lambda_0}\right) \exp\left(\frac{j2\pi\hat{r}_{tx,LOS}^T \cdot \vec{v}}{\lambda_0}\right) \quad (3)$$

where F is the field pattern in the vertical or horizontal plane; $\hat{r}_{tx,n,m}$ and $\hat{r}_{rx,n,m}$ are the ray m direction vectors in spherical units, heading from the BS and reaching the UE, respectively; \vec{d} is the location vector of the antenna elements; \vec{d}_{3D} is the distance between the BS and the UE; \vec{v} is the user speed; and λ_0 is the wavelength corresponding to the carrier frequency. Moreover, $\{\Phi_{n,m}^{\theta\theta}, \Phi_{n,m}^{\theta\phi}, \Phi_{n,m}^{\phi\theta}, \Phi_{n,m}^{\phi\phi}\}$ are the random initial phases for each ray m of each cluster n and for four different polarization combinations ($\theta\theta, \theta\phi, \phi\theta, \phi\phi$), with distribution $\sim \mathcal{U}(-180^\circ, +180^\circ)$, and $\kappa_{n,m}$ is the cross-polarization power ratio (XPR) for each ray m of each cluster n (7.5-21 in [22]).

The channel generation implementation in the link-level simulator is obtained by several blocks. All the blocks launch the same mex-file with a different configuration:

- *cdlModelInit*: during the configuration of the link simulator parameters, the channel is initialized and the memory is allocated on the basis of parameter selection (e.g., the allocated memory grows with the number of antenna elements). The *mexLock()* function is used to reserve memory to the mex application.
- *cdlModelRegen*: during both the initialization and running phase, the main AoA/AoD of the clusters can be modified to analyze multiple MIMO configurations in a single simulation.
- *cdlModelRun*: the CDL is generated each slot and is applied to the input data. The time-to-frequency conversion of the channel is applied to report the generated channel matrix as a function of the frequency.
- *cdlModelDelete*: at the end of the simulation, the memory reserved to the mex application is deallocated using the *mexUnlock()* function.

The simulator can then operate in two different modes: the frequency domain or the time domain. In the time domain mode, the channel is applied on the oversampled OFDM signal after the IFFT. In the frequency domain mode, the channel is applied in the frequency domain on the OFDM signal before the IFFT. In this case, the OFDM modulator and demodulator are by-passed, as are all the blocks between them, allowing a significant

reduction in the computational load. However, the frequency domain mode is applicable only in scenarios characterized by low mobility (i.e., when the channel is nearly constant within the OFDM symbol duration) and when the maximum delay of the channel is lower than the cyclic prefix length. In the simulations presented in the following, the time domain mode is selected and an impulse response is generated after a certain time interval, defined as an integer multiple of the slot length, and then interpolated in the time domain. For the frequency domain, an impulse response would be generated for each OFDM symbol.

4. A Deep Learning-Based CSI Feedback for New Radio

To obtain the optimal precoder for MIMO communications, the channel knowledge is required by the transmitter (i.e., the BS). In the frequency division duplexing (FDD) mode, the uplink and downlink transmissions occur in different frequency bands, so there is no channel reciprocity. Thus, the UE must estimate the downlink channel and send this estimation back to the BS. This procedure is the channel state information feedback. The drawback of this system is the unbearable overhead resulting from the higher number of antenna elements, both in transmission and reception, as highlighted in [12]. The only way to adopt the FDD mode for MIMO transmission is to significantly reduce the feedback overhead—machine learning represents one viable solution to address the problem. In [12], the author designed a DL-based CSI sensing and recovery mechanism, named CsiNet. The main difference from previous DL-based investigations was consideration of the recovery stage. The CsiNet model is based on two main modules:

- Encoder: the encoder generates compressed codewords by learning a proper mapping from the estimated channel matrices (dimensionality reduction). It is implemented by the UE.
- Decoder: the decoder tries to recover the original CSI from the codewords generated by the encoder, learning the inverse mapping. It is implemented at the BS sides, whereas the codewords are received from the UE through a feedback channel.

In this paper, we extend the CsiNet model for CSI information reduction and recovery in a 5G context, producing a deep convolutional neural network (CNN), named NR-CsiNet. The additional features are intended to widen the applicability of the model to a more practical range of use cases:

- We sought to make the model compatible with the typical 5G NR MIMO communications, in contrast to the single receiving antenna limitation of [12]. Practically, this introduces a third dimension to the channel matrices due to the different channel coefficients for each receiving antenna.
- We also considered the downlink channel estimation, whereas this additional step was beyond the scope of [12]. This means that, instead of feeding our NR-CsiNet with perfect channel matrices, we provide as input, noisy matrices, estimated through the CSI-RS.

The CNN is built and trained via Tensorflow's Keras API [48]. To make this DL-based system work with the 5G NR simulator, the traditional feedback reporting blocks are rearranged. A new block imports the pre-trained encoder and uses it to elaborate an encoded version of the channel matrix, previously estimated from the CSI-RS. Then, at the transmitter side, we insert a specular decoding block which imports the pre-trained decoder model and uses the reconstructed channel matrix at its output to determine the optimal precoding vector.

4.1. System Model

While the authors of [12] adopted a multiple-input single-output (MISO) system, i.e., a transmitter deploying an antenna array of N antennas with only one antenna at the receiver side, here, we consider a 5G NR-compliant MIMO system, with M receiving antennas. For the c -th sub-carrier at the m -th receiving antenna, the received signal can be expressed as

$$y_{m,c} = \mathbf{h}_{m,c}^H \mathbf{v}_c x_c + z_{m,c} \quad (4)$$

where:

- $\mathbf{h}_{m,c}$ is the complex channel vector relative to the c -th sub-carrier and to the m -th receiving antennas of size $N \times 1$;
- \mathbf{v}_c is the complex precoding vector of size $N \times 1$;
- x_c is the complex data symbol transmitted on the c -th sub-carrier;
- $z_{m,c}$ denotes the additive noise on the c -th sub-carrier relative to the m -th receiving antenna.

The precoding vector $\mathbf{v} = \{\mathbf{v}_c : c = 1, \dots, \tilde{C}\}$, with \tilde{C} the number of sub-carriers of the OFDM system, is computed by the BS once it receives the CSI feedback information $\mathbf{H}_c = [\mathbf{h}_1, \dots, \mathbf{h}_M] \in \mathbb{C}^{N \times M}$. The amount of coefficients the UE must feedback is then $N \times M \times \tilde{C}$, which can easily become too demanding from a complexity point of view, since resource elements must be dedicated to the channel reporting, instead of data packets (i.e., a capacity drop). The M $y_{m,c}$ symbols received per sub-band are then combined at the receiver side and the final estimated symbol \hat{y}_c on the c -th sub-carrier is obtained as follows:

$$\hat{y}_c = \mathbf{u}_c^H \mathbf{y}_c \quad (5)$$

where \mathbf{u}_c is an $M \times 1$ combiner and $\mathbf{y}_c = [y_{1,c}, \dots, y_{M,c}]^T$ is an $M \times 1$ vector containing all the received symbols on the c -th sub-carrier.

4.2. The Channel Estimation Task

When considering massive MIMO systems, the channel estimation task is fundamental and many studies have addressed the issue [19,20] since beam-forming techniques strongly depend on the outcome of this phase. In [12], the main focus is the compression task, i.e., the model in [12] takes as input a perfect channel matrix and learns efficient encoding and decoding algorithms to compress and decompress it with minimum error. Our model, instead, aims to deal also with the challenging downlink channel estimation task. This means that the compression and decompression process is not performed on the ideal channel matrix generated by the CDL channel module, but on a noisy estimation of it, coming from the CSI-RS compensation. The model is expected to learn efficient encoding and decoding algorithms that, besides compressing and decompressing the input matrices, must be able to remove the noisy contributions and reconstruct the clean channel matrices with minimum error. It should be noted that, to simplify the experiments, we decided to consider only the noise affecting the downlink transmission to avoid simulating the uplink channel.

To include this channel estimation task, we need to define two different instances of the channel matrix \mathbf{H} :

- an ideal channel matrix \mathbf{H}^{id} , generated by the CDL module of the software simulator. It represents the desired output of the NR-CsiNet model;
- a noisy version of the perfect channel matrix \mathbf{H}^{CSI-RS} , estimated from the CSI-RS pilots by means of a simple interpolation. It represents the input of the NR-CsiNet model.

To obtain a first raw estimation of the complete downlink complex channel matrix for each transmitting/receiving antenna pair, we performed a cubic radial basis function (RBF) interpolation on the real and the imaginary parts separately, using a MATLAB predefined library.

An example of the result obtained is shown in Figure 2, where n and m are the transmitting and receiving antenna indexes, respectively. The estimated \mathbf{H}^{CSI-RS} matrices are collected by simulating different conditions of the channel; practically, we select a couple of different SNR values, 10 dB and 20 dB, and train a different NR-CsiNet model for each of these values. Then, we test the behavior of these models at different levels of channel quality; to be precise, simulations are performed with SNR values varying between -5 dB and 35 dB, with a 5 dB granularity. To simplify the notation, in the following subsections, we refer to the two channel instances \mathbf{H}^{id} and \mathbf{H}^{CSI-RS} as a single matrix \mathbf{H} , meaning that all the processing steps described have to be performed on both.

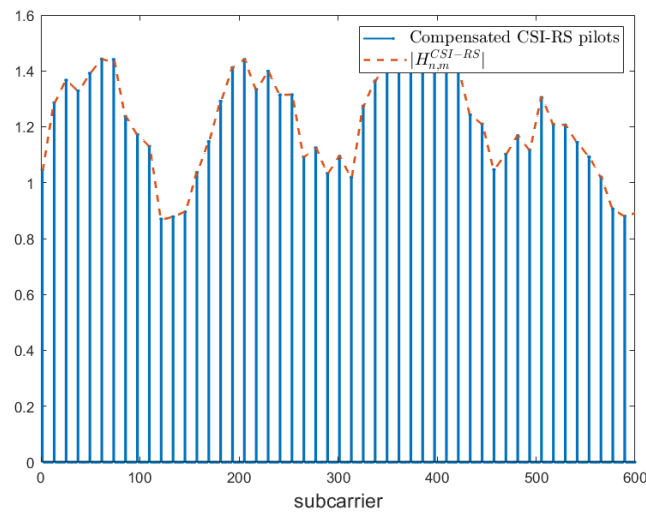


Figure 2. Module of interpolated channel matrix $|\mathbf{H}_{n,m}^{\text{CSI-RS}}|$.

4.3. DFT-Based Preprocessing of CSI Information

The first processing that is applied to the estimated channel matrix in the feedback task is the 2D discrete Fourier transform (DFT), with the goal of sparsifying the coefficients of \mathbf{H} in the angular-delay domain, as proposed in [12]. To deal with more antennas at the receiver, this operation is iterated M times, thus producing a different transformed spatial-frequency channel matrix $\tilde{\mathbf{H}}_m$ for each receiving antenna:

$$\tilde{\mathbf{H}}_m = \mathbf{F}_d \mathbf{H}_m \mathbf{F}_a^H \quad (6)$$

where \mathbf{F}_d and \mathbf{F}_a are $\tilde{C} \times \tilde{C}$ and $N \times N$ DFT matrices, respectively. \mathbf{H}_m has size $\tilde{C} \times N$ and m is the index of the receiving antenna. Figure 3 shows an instance of \mathbf{H}_m with $N = 8$ and $\tilde{C} = 600$.

As the inter-arrival time between rays only spans a certain range, only a few elements of the matrix $\tilde{\mathbf{H}}_m$ contain significant information, while the vast majority of the matrix elements are made up of coefficients close to zero; thus, DFT processing is necessary. It is obvious that the remaining columns of $\tilde{\mathbf{H}}_m$ can be eliminated and that only the core $C < \tilde{C}$ columns are significant (Figure 4).

A truncated version of $\tilde{\mathbf{H}}_m$ is then obtained as a first step to decrease the feedback overhead, denoted as $\tilde{\mathbf{H}}_m^t$ and having size $C \times N$.

Successively, we obtain the spatial-frequency domain version of the truncated matrix in the angular-delay domain $\tilde{\mathbf{H}}_m^t$, through an inverse DFT (IDFT). This matrix, denoted \mathbf{H}_m^t , eases the compression and reconstruction task of the CNN, since, in the spatial-frequency domain, the “image” representation of the matrix is characterized by a smoother profile (Figure 3), whereas, in the angular-delay domain, it is recognizable as a composite of crisp and sharp lines (Figure 4). This last step is not considered in the DFT processing addressed in [12]; we anticipate that it will improve the overall system behaviour.

Lastly, the matrix \mathbf{H}_m^t is normalised to assume values only inside the interval $[0, 1]$. This can help the training of the CNN by reducing the effect of outliers.

In a specular way, at the output of the decoder implemented at the BS, a rescaling operation, followed by a DFT and an inverse DFT operation, must be performed.

4.4. Multiple Receiving Antennas

While the authors of paper [12] limit their DL-based solution to single receiving antenna scenarios ($M = 1$), our purpose is to make the model applicable also to MIMO contexts. With this in mind, the downlink channel information relative to each distinct receiving antenna must, in some way, be organized at the input of the CNN. We decided to maintain only two input channels, one for the real and one for the imaginary parts of

the matrices $\mathbf{H}_m^{\text{CSI-RS},t}$; this means that the matrices relative to distinct receiving antennas are conveyed through the CNN on the same input channels.

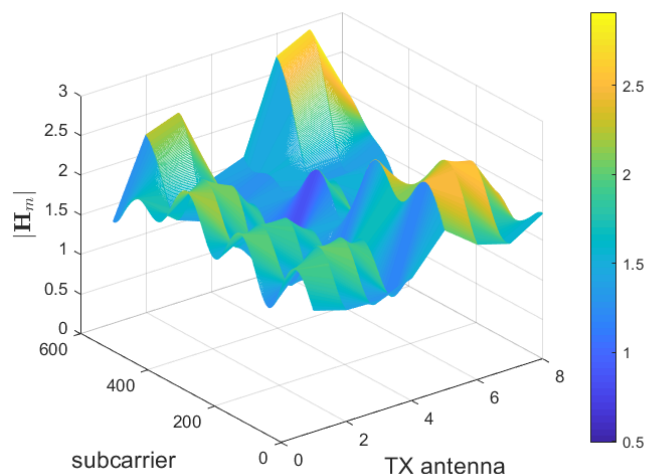


Figure 3. Module of channel matrix $|\mathbf{H}_m|$ [32].

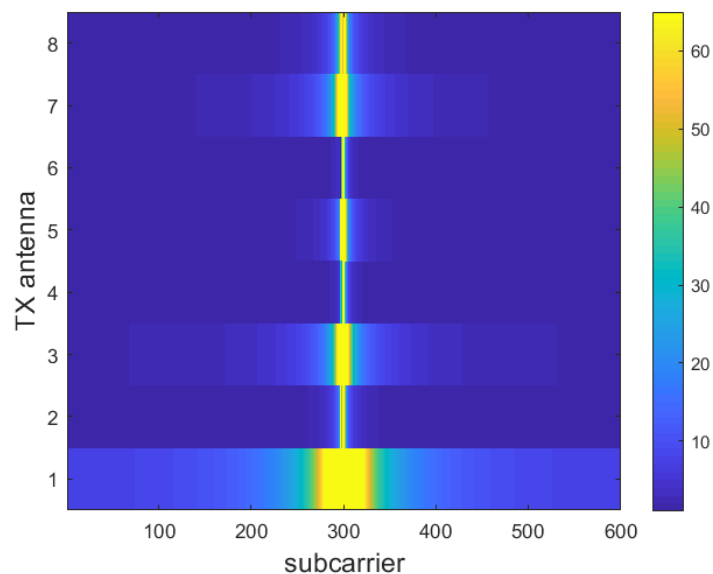


Figure 4. Module of DFT-transformed channel matrix $|\tilde{\mathbf{H}}_m|$ [32].

The reasons behind this approach are several:

- we do not need to train a different CNN model for each distinct receiving antenna;
- the number of parameters learned by the CNN is not affected by the number of receiving antennas M . This means that in massive MIMO scenarios, where $M \gg 1$, the complexity of the model does not increase with M and, thus, the risk of running into overfitting phenomena remains limited;
- if the different receiving antennas were spatially correlated, this correlation could be exploited during the learning process and, probably, smaller training datasets would be sufficient to obtain effective and generalized models.

4.5. NR-CsiNet Encoder

The structure of the encoder of the NR-CsiNet (Figure 5a) is based on the encoder designed for the CsiNet proposed in [12]. It consists of a deep convolutional neural network, where the input has size $C \times N \times 2$, corresponding to the real and imaginary parts of the truncated matrices $\mathbf{H}_m^{\text{CSI-RS},t}$. The first layer of the CNN is a convolutional layer adopting

a filter of size 3×3 that generates 2 feature maps. A reshape process vectorizes the two feature maps into a vector of size $T \times 1$, where $T = 2 \cdot C \cdot N$. The last block is a fully connected layer which outputs the actual codeword \mathbf{s} , one for each receiving antenna, as a vector of size R . The overall compressing ratio is $K = \frac{R}{C \cdot N}$.

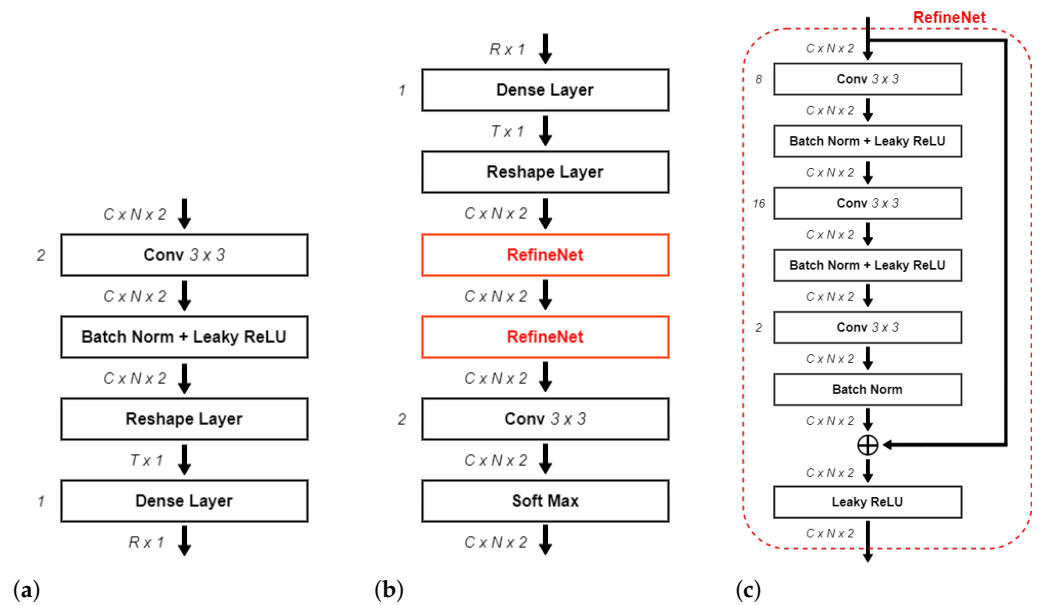


Figure 5. NR-CsiNet representation. (a) Block scheme of the NR-CsiNet encoder. (b) Block scheme of the NR-CsiNet decoder. (c) Block scheme of the RefineNet unit.

4.6. NR-CsiNet Decoder

As for the encoder, the NR-CsiNet decoder (Figure 5b) utilizes the implementation scheme of the CsiNet decoder presented in [12]. The first layer is a fully connected layer which receives as input the codeword \mathbf{s} produced by the encoder at the UE side, sent to the BS. The result of this layer is a coarse reconstruction of the real and imaginary part of the channel matrix $\mathbf{H}_m^{id,t}$ in the form of a vector of size $T \times 1$. A reshape layer transforms the vector back to the two matrices corresponding to the real and imaginary part of the channel, i.e., two matrices of size $C \times N$ each. In order to improve these initial estimates, a particular set of blocks, called the RefiNet Unit (Figure 5c), is inserted in the decoder. Each RefineNet unit is composed of three convolutional layers using a kernel of size 3×3 . The first and second layers generate 8 and 16 feature maps, respectively, while the third outputs the final reconstruction of the $\mathbf{H}_m^{id,t}$ matrices (in the form of two feature maps). The feature maps produced by the three layers of a RefineNet unit are forced by a zero-padding technique to have the same size as the input channel matrix ($C \times N$). Moreover, each layer is followed by a rectified linear unit (ReLU) [49,50] and a batch normalisation layer [50,51]. Before the last leaky ReLU computation, the input of the RefiNet is added to the output of the last CNN (after proper batch normalisation). This step is necessary since, as reported in the literature [52], the sequence of many CNN layers may lead to a vanishing gradient problem. To identify a suitable trade-off between the complexity and reconstruction capabilities, the authors of [12] investigated the impact of the number of RefiNet units, determining that two RefineNet units were sufficient. In our work, we investigated the effect of adopting different numbers of RefiNet units and observed that exploiting more than two RefiNet units did not significantly impact the reconstruction quality, though at the expense of greater complexity. An additional convolutional layer follows the RefiNet Units and, finally, a softmax layer [50,53] scales the reconstructed coefficients into the $[0, 1]$ range.

4.7. Training Process

Using the same notation adopted in [12], the set of trainable parameters is denoted as $\Theta = \{\Theta_{en}, \Theta_{de}\}$. The truncated input and output of the NR-CsiNet model for the i -th patch are denoted as $\mathbf{H}_i^{\text{CSI-RS},t}$ and $\hat{\mathbf{H}}_i^t = f(\mathbf{H}_i^{\text{CSI-RS},t}; \Theta) = f_{de}(f_{en}(\mathbf{H}_i^{\text{CSI-RS},t}; \Theta_{en}); \Theta_{de})$, respectively.

The loss function is defined as

$$L(\Theta) = \frac{1}{S} \sum_{i=1}^S \|\mathbf{H}_i^{\text{CSI-RS},t}; \Theta - \mathbf{H}_i^{\text{id},t}\|_2^2 \quad (7)$$

where S denotes the number of training samples and $\|\cdot\|_2$ is the Euclidean norm operator.

5. Simulation Results

In this section, we report the main results collected from the NR-CsiNet model and described in the previous sections. The assessment of the performance is carried out through simulation trials exploiting the NR link-level simulator presented in Section 3 by substituting the blocks for CSI feedback definition and reporting with our DL-based models. The transmitter precoding matrix is obtained by a single value decomposition (SVD) operation on the recovered channel matrix, which is generated from the compressed feedback sent back by the UE. This means selecting the eigenvector that corresponds to the higher eigenvalue as the precoding vector for the subsequent downlink transmission. Taking advantage of the spatial diversity given by the M receiving antennas, a maximal ratio combiner (MRC) is used at the receiving side.

Table 1 reports the main parameters of interest concerning the simulator settings:

Table 1. Simulation parameters for NR-CsiNet training.

NR Simulator Parameters	
System bandwidth	5 MHz
Subcarrier spacing	15 kHz
Time slot	1 ms
Carrier frequency	3.64 GHz
iMCS	21
(Modulation, coding rate)	(64-QAM, 0.694)
TBS	19,968 bit
NFFT	512
\tilde{C}	300
Transmission layers	1
Codewords	1
CSI-RS Tx period	4 slots
CSI reporting type	Type I
Channel model	CDL-B
Delay spread model	Nominal

More details on some of the main parameters are provided below:

Time slot Time duration of the transport block of data transmission, corresponding to 14 OFDM symbols.

Modulation and coding scheme (MCS) Configuration parameter of the system to determine the coding rate and the order and type of modulation to be adopted for the transmission.

Transport block size (TBS) Value corresponding to the number of bits transmitted in one slot.

NFFT Smaller power of two greater than the number of sub-carriers used by the system to be used by FFT and IFFT operations.

Number of sub-carriers for data transmission \tilde{C} Computed as the difference between the NFFT value and the number of sub-carriers composing the band guards. In our system $\tilde{C} = 512 - 106 - 106$.

Delay spread model Parameter that determines the delay profile, affecting the scaling factor for the time of arrival of each cluster signal, leading to a shorter or longer delay spread.

It should be noted that the adopted bandwidth is smaller than the one commonly expected for a 5G NR network. Behind this choice are considerations of computational complexity since a larger bandwidth would lead to a significant increase in simulation and training time.

Table 2 shows the hyperparameters adopted for the training and testing process of our model.

Table 2. NR-CsiNet model parameters [32].

NR-CsiNet Parameters	
Training set	1000 sim of 100 slots
Validation set	100 sim of 100 slots
Testing set	100 sim of 100 slots
Learning rate	0.001
Loss function	MSE
Optimizer	Adam
Epochs	300
RefineNet units	2
K_{DFT}	3

The DFT-compression factor K_{DFT} , which should not be confused with the final compression factor K introduced in Section 4.5, is defined as:

$$K_{\text{DFT}} = \frac{\tilde{C}}{C} \quad (8)$$

where \tilde{C} and C are the same parameters defined in Section 4.

The transmitting antenna system consists of a 2×2 planar array formed of dual-polarized antenna components, while the receiver is equipped with a single dual-polarized antenna; thus, we have $N = 8$ transmitting antennas and $M = 2$ receiving antennas. Since there are a relatively small number of antennas, it is obvious that this is not a massive MIMO scenario. This decision was forced by considerations solely related to computational complexity because adding more antennas will increase simulation times proportionately. Larger values of N and/or M should be envisioned for future research.

In our system, the DFT-compression factor is fixed at $K_{\text{DFT}} = 3$. The training process of the neural network is performed for two different SNR levels: 10 dB and 20 dB. Figures 6 and 7 highlight how the normalised mean square error (NMSE) of the channel reconstruction for the NR-CsiNet model does not depend on the training data SNR level, except for the lowest compression value K ($K = 1/120$).

The PDSCH throughput results for SNR training values of 10 dB and 20 dB are reported in Figures 8 and 9, respectively. The relevant SNR range is $[-5, 23]$ dB. The results for the second evaluation metric, BLER1, i.e., BLER at the first transmission attempt, are reported in Figures 10 and 11 for the $[0, 23]$ dB SNR range. The performance of the NR-CsiNet is compared with the PMI reporting technique, referred to here as follow PMI. This procedure requires channel measurements by the UE from CSI-RS, in addition to which a PMI is determined [37]. The PMI is an index that the UE sends back to the BS to suggest the preferable precoding vector corresponding to that index from a predefined precoding matrix.

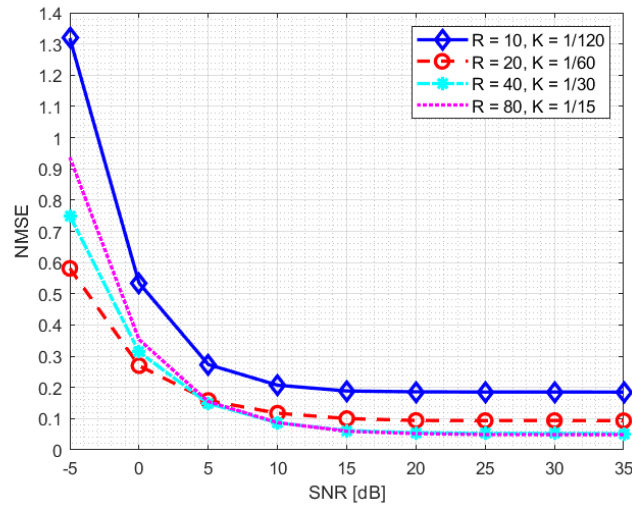


Figure 6. NMSE statistics for NR-CsiNet model trained at SNR = 10 dB.

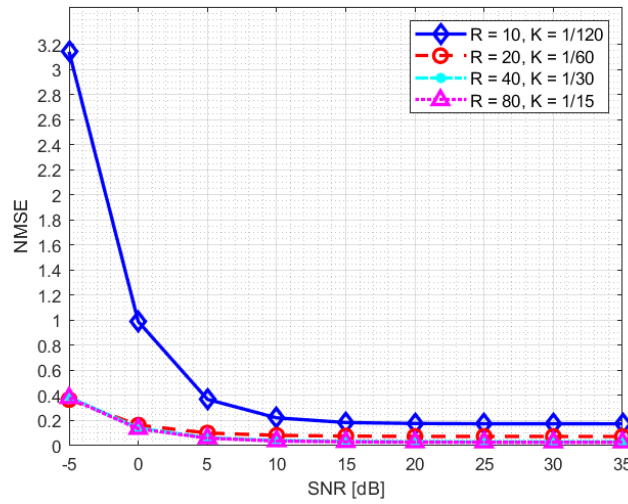


Figure 7. NMSE statistics for NR-CsiNet model trained at SNR = 20 dB.

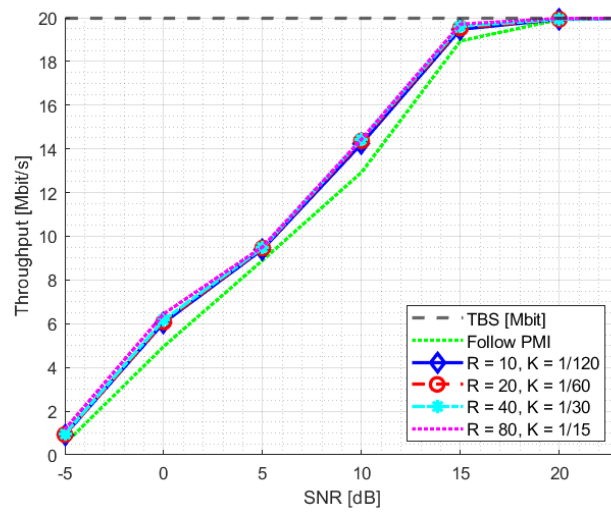


Figure 8. Throughput statistics for NR-CsiNet model trained at SNR = 10 dB.

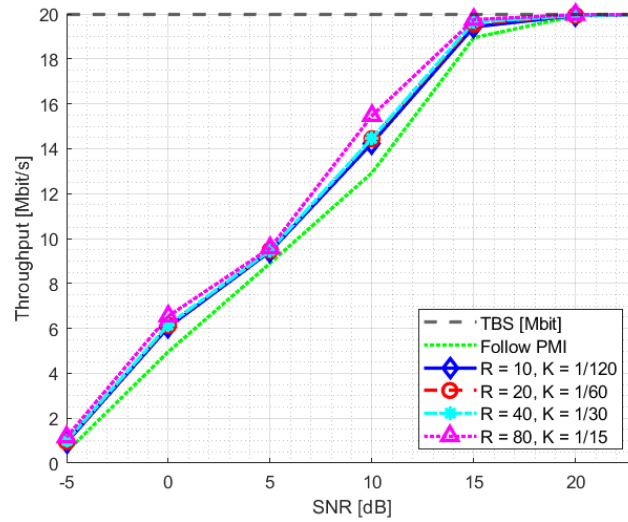


Figure 9. Throughput statistics for NR-CsiNet model trained at SNR = 20 dB [32].

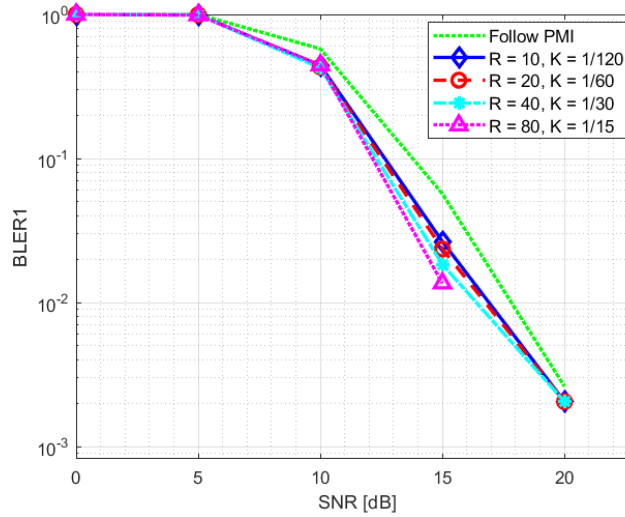


Figure 10. BLER1 statistics for NR-CsiNet model trained at SNR = 10 dB.

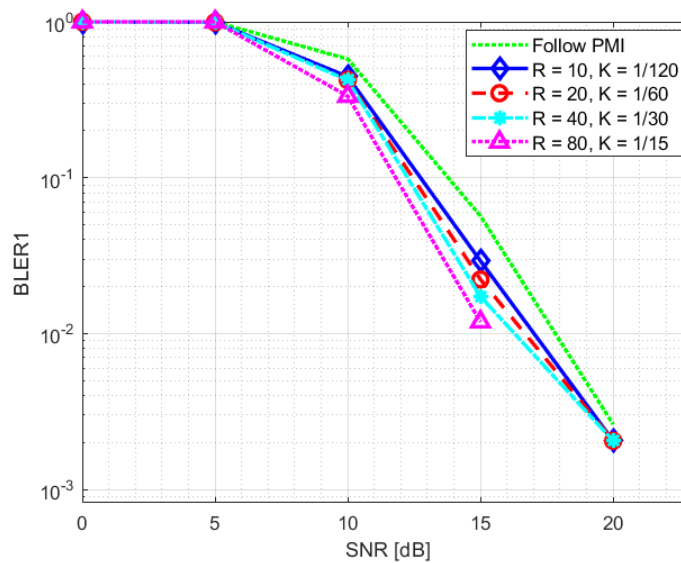


Figure 11. BLER1 statistics for NR-CsiNet model trained at SNR = 20 dB [32].

It is possible to infer from the graphs that, even when dealing with noisy channel estimates rather than ideal channel matrices, our DL-based technique performs better than the follow PMI algorithm in 8×2 cases. The plots specifically illustrate that over 20 dB of SNR, both the NR-CsiNet and follow PMI throughput curves achieve the transport block size (TBS) maximum limit; however at lower SNR, all the variations of our DL solution perform better. It is important to note that, in all of our experiments, we used the same channel model configuration. The simulation results demonstrated that, despite being intended for multipath contexts, this type of channel model typically presents a dominant path, which implicitly defines a preferential direction for signal transmission. It is obvious that the follow PMI is expected to function well when taking into account such a channel model. In fact, among a range of potential beams, the PMI index merely determines which one is most precisely pointed to the receiver. Our hypothesis is that our DL-based technique has the best chance of outperforming the follow PMI algorithm in “richer” channel realizations when the multipath is not dominated by a single LOS cluster.

The complexity of DL-based solutions is one of the main shortcomings in their implementation at the user equipment side, characterized by limited power and hardware resources. In Table 3, we report the number of parameters and FLOPs required for the NR-CsiNet encoder, which are greatly dependent on the choice of the factor K . For the CNN layer, the number of learnable parameters are computed as $par_{CNN} = \# filters \cdot (filter\ width \cdot filter\ height \cdot input\ feature\ maps + 1)$ while the FLOPs are $FLOP_{CNN} = 2 \cdot \# filters \cdot filter\ width \cdot filter\ height \cdot Output\ size$; for the FC layer, the parameters are $par_{FC} = (Input\ size + 1) \cdot Output\ size$, while the FLOPs are $FLOP_{FCL} = 2 \cdot Input\ size \cdot Output\ size$.

Table 3. Complexity of the encoder in terms of learnable parameters and FLOPs.

K	Parameters	FLOPs
1/120	5.381	179.200
1/60	10.725	243.200
1/30	21.411	371.200
1/15	42.785	627.200

The FLOP amount is of the same order of magnitude for both the CNN and the FC, while the FC requires a significantly higher number of parameters since it needs a weight for each input-output node link (plus the bias). The advanced hardware designed specially for neural network operations can significantly help the deployment of such technology. For the follow PMI, the parameters required are almost negligible, since the DFT codebook can be generated whenever required. Nonetheless, the complexity from a FLOPs perspective is almost zero at the base station; however, at the user side it can be quite significant, since it requires a full search over all the DFT beam-forming codewords in each frequency position to maximize the capacity. Thus, the number of FLOPs can be estimated as the multiplication between two complex matrices of size $M \times N$ and $N \times C_s$: $FLOP_{followPMI} = 8MC_sNL$; where C_s is the codebook size and L is the number of frequency bins where the PMI is computed. In our tests, a different PMI is required for each PRB; thus, $L = 25$ and we use as the codebook a DFT beamspace with an oversampling factor $O_F = 4$; therefore, $C_s = O_F N = 32$. The final number of FLOPs with follow PMI is equal to $FLOP_{followPMI} = 102.400$. The use of a system such as NR-CsiNet instead of the simpler follow PMI demands higher complexity; however, unlike follow PMI, it can allow the BS to apply advanced precoding techniques that exploit complete CSI, which is not achievable through simplified codebooks (e.g., in the case of users in NLOS and/or who could interfere with each other). We aim to investigate NR-CsiNet capabilities for more sophisticated MIMO scenarios in future studies.

6. Conclusions

The CSI feedback for FDD transmission is an open problem and DL-based techniques have made a new tool available to attempt to solve it. The channel state information availability is of crucial importance for MIMO communications and for 5G New Radio. In this paper, we investigated the potential of a neural network technique to serve as an alternative to the traditional feedback reporting block. From the work of the authors reported in [12], we obtained a NR-CsiNet model, able to deal with SU-MIMO systems and providing channel estimation capabilities. Simulations were performed by exploiting a 5G NR-compliant physical layer simulator with a CDL channel model, which enables modelling of the complex environment envisioned by the 5G network. The results showed that our DL-based feedback system can achieve performances comparable to those obtained with a traditional PMI-based reporting technique and clearly outperformed it in some cases. The framework and the data collected in this study provide a solid basis for more in-depth tests of the capabilities of DL for CSI feedback in 5G NR, enabling investigation of a larger number of scenarios and different configurations, such as larger MIMO systems (i.e., massive MIMO), and exploitation of the channel matrix to support spatial division multiplexing solutions, that are currently difficult to obtain in FDD mode.

Moreover, our objective was to highlight useful aspects of research on this topic, which is of significant interest considering the fundamental role that MIMO systems play in network performance improvement.

Author Contributions: Conceptualization, R.G. and R.F.; formal analysis, E.Z., D.G.R. and R.G.; investigation, E.Z., D.G.R. and R.T., software, E.Z. and R.F.; supervision, D.G.R. and R.G.; writing—original draft preparation, R.T. and E.Z.; writing—review and editing, D.G.R., R.T. and R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from TIM S.p.A. Restrictions apply to the availability of these data, which were used under licence for this study. Data are available from the corresponding author, D.G.R., with the permission of TIM S.p.A., upon reasonable request.

Acknowledgments: Riccardo Tuninato acknowledges support from TIM S.p.A. through the Ph.D. scholarship.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jiang, C.; Zhang, H.; Ren, Y.; Han, Z.; Chen, K.; Hanzo, L. Machine Learning Paradigms for Next-Generation Wireless Networks. *IEEE Wirel. Commun.* **2017**, *24*, 98–105. [[CrossRef](#)]
2. Bkassiny, M.; Li, Y.; Jayaweera, S.K. A Survey on Machine-Learning Techniques in Cognitive Radios. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1136–1159. [[CrossRef](#)]
3. Sharma, V.; Bohara, V. Exploiting machine learning algorithms for cognitive radio. In Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Delhi, India, 24–27 September 2014; pp. 1554–1558.
4. Murudkar, C.V.; Gitlin, R.D. Optimal-Capacity, Shortest Path Routing in Self-Organizing 5G Networks using Machine Learning. In Proceedings of the 2019 IEEE 20th Wireless and Microwave Technology Conference (WAMICON), Cocoa Beach, FL, USA, 8–9 April 2019; pp. 1–5.
5. Khan Tayyaba, S.; Khattak, H.A.; Almogren, A.; Shah, M.A.; Ud Din, I.; Alkhalifa, I.; Guizani, M. 5G Vehicular Network Resource Management for Improving Radio Access Through Machine Learning. *IEEE Access* **2020**, *8*, 6792–6800. [[CrossRef](#)]
6. Hussain, F.; Hassan, S.A.; Hussain, R.; Hossain, E. Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1251–1275. [[CrossRef](#)]
7. Wang, T.; Wen, C.; Wang, H.; Gao, F.; Jiang, T.; Jin, S. Deep learning for wireless physical layer: Opportunities and challenges. *China Commun.* **2017**, *14*, 92–111. [[CrossRef](#)]
8. Zhou, R.; Liu, F.; Gravelle, C.W. Deep Learning for Modulation Recognition: A Survey With a Demonstration. *IEEE Access* **2020**, *8*, 67366–67376. [[CrossRef](#)]

9. Zhang, M.; Zeng, Y.; Han, Z.; Gong, Y. Automatic Modulation Recognition Using Deep Learning Architectures. In Proceedings of the 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, Greece, 25–28 June 2018; pp. 1–5.
10. Rajendran, S.; Meert, W.; Giustiniano, D.; Lenders, V.; Pollin, S. Deep Learning Models for Wireless Signal Classification With Distributed Low-Cost Spectrum Sensors. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 433–445. [[CrossRef](#)]
11. Huang, H.; Guo, S.; Gui, G.; Yang, Z.; Zhang, J.; Sari, H.; Adachi, F. Deep Learning for Physical-Layer 5G Wireless Techniques: Opportunities, Challenges and Solutions. *IEEE Wirel. Commun.* **2020**, *27*, 214–222. [[CrossRef](#)]
12. Wen, C.; Shih, W.; Jin, S. Deep Learning for Massive MIMO CSI Feedback. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 748–751. [[CrossRef](#)]
13. Wang, T.; Wen, C.; Jin, S.; Li, G.Y. Deep Learning-Based CSI Feedback Approach for Time-Varying Massive MIMO Channels. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 416–419. [[CrossRef](#)]
14. Gui, G.; Huang, H.; Song, Y.; Sari, H. Deep Learning for an Effective Nonorthogonal Multiple Access Scheme. *IEEE Trans. Veh. Technol.* **2018**, *67*, 8440–8450. [[CrossRef](#)]
15. Huang, H.; Song, Y.; Yang, J.; Gui, G.; Adachi, F. Deep-Learning-Based Millimeter-Wave Massive MIMO for Hybrid Precoding. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3027–3032. [[CrossRef](#)]
16. O’Shea, T.; Hoydis, J. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [[CrossRef](#)]
17. Ye, H.; Li, G.Y.; Juang, B. Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 114–117. [[CrossRef](#)]
18. Soltani, M.; Pourahmadi, V.; Mirzaei, A.; Sheikhzadeh, H. Deep Learning-Based Channel Estimation. *IEEE Commun. Lett.* **2019**, *23*, 652–655. [[CrossRef](#)]
19. Fascista, A.; De Monte, A.; Coluccia, A.; Wymeersch, H.; Seco-Granados, G. Low-Complexity Downlink Channel Estimation in mmWave Multiple-Input Single-Output Systems. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 518–522. . LWC.2021.3134826. [[CrossRef](#)]
20. Kebede, T.; Wondie, Y.; Steinbrunn, J. Channel Estimation and Beamforming Techniques for mm Wave-Massive MIMO: Recent Trends, Challenges and Open Issues. In Proceedings of the 2021 International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates, 31 October–2 November 2021; pp. 1–8. [[CrossRef](#)]
21. Dahlman, E.; Parkvall, S.; Skold, J. *5G NR: The Next Generation Wireless Access Technology*; Elsevier Science: 2018.
22. 3GPP. *Study on Channel Model for Frequencies from 0.5 to 100 GHz*; Version 14.3.0; Technical Report (TR) 38.901; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2018.
23. Riviello, D.G.; Di Stasio, F.; Tuninato, R. Performance Analysis of Multi-User MIMO Schemes under Realistic 3GPP 3-D Channel Model for 5G mmWave Cellular Networks. *Electronics* **2022**, *11*, 330. [[CrossRef](#)]
24. Xie, H.; Gao, F.; Jin, S.; Fang, J.; Liang, Y. Channel Estimation for TDD/FDD Massive MIMO Systems With Channel Covariance Computing. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 4206–4218. [[CrossRef](#)]
25. Han, Y.; Hsu, T.; Wen, C.; Wong, K.; Jin, S. Efficient Downlink Channel Reconstruction for FDD Multi-Antenna Systems. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 3161–3176. [[CrossRef](#)]
26. Cao, Z.; Shih, W.T.; Guo, J.; Wen, C.K.; Jin, S. Lightweight Convolutional Neural Networks for CSI Feedback in Massive MIMO. *IEEE Commun. Lett.* **2021**, *25*, 2624–2628. [[CrossRef](#)]
27. Li, Q.; Zhang, A.; Liu, P.; Li, J.; Li, C. A Novel CSI Feedback Approach for Massive MIMO Using LSTM-Attention CNN. *IEEE Access* **2020**, *8*, 7295–7302. [[CrossRef](#)]
28. Ji, D.J.; Cho, D.H. ChannelAttention: Utilizing Attention Layers for Accurate Massive MIMO Channel Feedback. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 1079–1082. [[CrossRef](#)]
29. Lu, Z.; Wang, J.; Song, J. Binary Neural Network Aided CSI Feedback in Massive MIMO System. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 1305–1308. [[CrossRef](#)]
30. Guo, J.; Wen, C.K.; Jin, S. Deep Learning-Based CSI Feedback for Beamforming in Single- and Multi-Cell Massive MIMO Systems. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 1872–1884. [[CrossRef](#)]
31. Liu, Z.; Zhang, L.; Ding, Z. An Efficient Deep Learning Framework for Low Rate Massive MIMO CSI Reporting. *IEEE Trans. Commun.* **2020**, *68*, 4761–4772. [[CrossRef](#)]
32. Zimaglia, E.; Riviello, D.G.; Garello, R.; Fantini, R. A Novel Deep Learning Approach to CSI Feedback Reporting for NR 5G Cellular Systems. In Proceedings of the 2020 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW), Riga, Latvia, 1–2 October 2020; Volume 1, pp. 47–52. [[CrossRef](#)]
33. Ahmadi, S. *5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio Standards*; Elsevier Science: 2019.
34. 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 8)*; Version 8.9.0; Technical Specification (TS) 36.211; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2009.
35. 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)*; Version 10.7.0; Technical Specification (TS) 36.211; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2013.
36. 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects (Release 9)*; Version 9.0.0; Technical Specification (TS) 36.814; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2010.
37. 3GPP. *NR; Physical Layer Procedures for Data (Release 15)*; Version 15.9.0; Technical Specification (TS) 38.214; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2020.

38. Roy, R.H. Spatial division multiple access technology and its application to wireless communication systems. In Proceedings of the 1997 IEEE 47th Vehicular Technology Conference. Technology in Motion, Phoenix, AZ, USA, 4–7 May 1997; Volume 2, pp. 730–734.
39. Liu, L.; Chen, R.; Geirhofer, S.; Sayana, K.; Shi, Z.; Zhou, Y. Downlink MIMO in LTE-advanced: SU-MIMO vs. MU-MIMO. *IEEE Commun. Mag.* **2012**, *50*, 140–147. [[CrossRef](#)]
40. Spencer, Q.H.; Peel, C.B.; Swindlehurst, A.L.; Haardt, M. An introduction to the multi-user MIMO downlink. *IEEE Commun. Mag.* **2004**, *42*, 60–67. [[CrossRef](#)]
41. Zhang, T.; Wang, R. Achievable DoF Regions of Three-User MIMO Broadcast Channel With Delayed CSIT. *IEEE Trans. Commun.* **2021**, *69*, 2240–2253. [[CrossRef](#)]
42. 3GPP. NR; *Physical Channels and Modulation*; Version 15.6.0; Technical Specification (TS) 38.211; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2019.
43. 3GPP. NR; *Multiplexing and Channel Coding*; Version 15.6.0; Technical Specification (TS) 38.212; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2019.
44. 3GPP. NR; *Physical Layer Procedures for Control*; Version 15.6.0; Technical Specification (TS) 38.213; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2019.
45. 3GPP. *Radio Resource Control (RRC); Protocol Specification*; Version 16.1.0; Technical Report (TR) 38.331; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2020.
46. 3GPP. *User Equipment (UE) Radio Access Capabilities*; Version 15.3.0; Technical Report (TR) 38.306; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2018.
47. 3GPP. NR; *Study on Test Methods*; Version 2.4.0; Technical Report (TR) 38.810; 3rd Generation Partnership Project (3GPP): Valbonne, France, 2018.
48. Tensorflow Keras Module. Available online: https://www.tensorflow.org/api_docs/python/tf/keras (accessed on 5 January 2023).
49. ReLU Keras API. Available online: https://keras.io/api/layers/activation_layers/relu/ (accessed on 5 January 2023).
50. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning (Adaptive Computation and Machine Learning)*; MIT Press: Cambridge, MA, USA, 2016.
51. Batch Normalization Keras API. Available online: https://keras.io/api/layers/normalization_layers/batch_normalization/ (accessed on 5 January 2023).
52. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [[CrossRef](#)]
53. Softmax Keras API. Available online: https://keras.io/api/layers/activation_layers/softmax/ (accessed on 5 January 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.