Deep Learning Techniques with MATLAB's Dynamic Toolkit for Channel Estimation in 5G Networks and Beyond

Miguel Á. Fuentes Ramírez[1*], Andrés F. Chacón Alonso[1*] and Gustavo A. Puerto Leguizamón[1*]

[1*] Universidad Distrital Francisco José de Caldas, Bogotá, 111711, Colombia.

*Corresponding author(s). E-mail(s): mafuentes@udistrital.edu.co; afchacona@udistrital.edu.co; gapuerto@udistrital.edu.co;

Abstract

This paper presents a toolkit aimed at streamlining channel estimation in mobile communications, a process that is crucial for optimizing transmitter-receiver communication. Despite the challenges of attenuation, multipath losses, noise and delays affecting signal transmission, traditional methods for channel estimation (CE) such as least squares (LS) and minimum mean square error (MMSE) fall short in several scenarios. To address this problem, the toolkik employs Deep Learning (DL) techniques for pilot-based channel estimation in 5G systems, especially under mobility-induced Doppler shift conditions, whereby the toolset allows emulating an environment as well as monitoring it in real time. Implemented in MATLAB, the tools facilitate simulations-emulations in line-of-sight and non-line-of-sight environments using a tapped delay line model (TDL). Performance evaluation metrics include bit-error rate (BER), error vector magnitude (EVM), estimation time, and mean square error (MSE). The results highlight the superiority of convolutional autoencoder (CAE), convolutional neural networks (CNN) and Denoising convolutional neural networks (DnCNN) over linear interpolation and practical estimation techniques in specific scenarios. In particular, the interface can dynamically adjust the signal-to-noise ratio (SNR), speed and modulation according to user settings, ensuring adaptability. In addition, it supports multiple artificial neural network (ANN) models, increasing its versatility for comprehensive analysis in dynamic mobile communication environments.

Keywords: Convolutional Neural Networks, Channel Estimation, Mobile communications, MATLAB toolkit, 5G and beyond

1 Introduction

The growing demand for fast, robust and reliable wireless communications requires strategic optimization of spectrum resources and data transmission recovery processes. In this quest, the advent of fifth-generation (5G) technology is fundamental, spearheading the digital transformation with its distinctive attributes of low latency, expansive bandwidth, maximum throughput and increased capacity [1]. To increase the performance of 5G communication systems and meet user needs, three fundamental service types: enhanced Mobile Broadband (eMBB), Mass Machine Type Communication (mMTC), and Ultra Reliable and Low Latency Communications (uRLLC) were proposed for 5G communications [2]. A cornerstone of 5G technology is orthogonal frequency division multiplexing (OFDM), operable in both uplink and downlink, which enhances system flexibility and scalability across diverse application domains, deployment modes and frequency spectrum [3]. Consequently, multiple subcarrier spacing numerologies (15 kHz, 30 kHz, 60 kHz, 120 kHz, 240 kHz and 480 kHz) and a wide range of subcarrier modulation schemes, such as quadrature phase shift keying (QPSK), quadrature amplitude modulation (16QAM), 64QAM and 256QAM, proliferate to suit various use cases.

CE becomes of paramount importance in communication systems by predicting the impact of signal transmission through propagation media. CE improves transmission quality by counteracting the effects of attenuation and distortion. Pilot-assisted methods, such as least squares (LS) and minimum mean square error (MMSE) techniques, although prevalent, face performance limitations and complexities in various scenarios. Consequently, DL techniques are gaining ground in CE, promising adaptability to changing channel conditions.

DL, a branch of machine learning and artificial intelligence, has become the backbone of contemporary computing, data analysis and scientific research, and heralds the fourth industrial revolution. Leveraging its prowess in processing vast data sets, DL has multiple applications spanning image and speech recognition, natural language processing, time series predictions, and data classification [4–7]. Anchored on artificial neural networks (ANNs), DL algorithms unravel complex patterns and features in intricate prediction or classification tasks, outperforming traditional ML methodologies due to their effectiveness with large datasets [8].

Previous work developed on DL models for CE dealt with Long Short-Term Memory (LSTM) models that constitute a recurrent neural network (RNN) architecture, and studies [9–11] demonstrate their superiority over the LS estimator across varying modulation schemes, channel models, pilot sequences, and antenna numbers, albeit with higher complexity. Addressing CE challenges, CNN models [12] treat the communication channel as a resource grid, showcasing CNN's superiority over LS and practical estimation across all Tapped Delay Line (TDL) models. Hybrid models like Convolutional LSTM [13], Autoencoders [14, 15], and Faster Super-Resolution CNN (FSRCNN) [16] underscore the significance of precise CE to mitigate channel effects, akin to image denoising or super-resolution techniques in OFDM systems.

The arrival of the sixth generation (6G) of mobile networks heralds a transformative era in wireless communication, promising unprecedented data rates, ultra-low latency and ubiquitous connectivity. Building on the foundation of 5G, 6G is set to revolutionize connectivity paradigms, enabling immersive experiences and seamless interactions between man and machine. In this context, a key challenge in the development of 6G is the accurate estimation of wireless channels, essential to optimize signal transmission and reception performance. Traditional techniques, such as LS and MMSE, face difficulties in modern wireless environments due to their dynamic and complex nature [17].

DL emerges as a promising solution to improve CE in 5G and pave the way for 6G networks. DL algorithms, in particular CNN, recurrent neural networks (RNNs), CAE and DnCNN, are excelling in extracting intricate patterns from huge data sets. Recent studies [18–20] have demonstrated the superiority of DL-based CE methods over conventional approaches, highlighting their accuracy and adaptability. Hybrid models, which integrate DL with domain-specific knowledge, further improve CE performance. As the telecom industry moves towards 6G, the integration of DL-based CE techniques is set to revolutionize next-generation network design, unlocking unprecedented levels of spectral efficiency, reliability and scalability, and fostering innovation in various applications and services.

Digital transformation has generated significant changes in several sectors, highlighting the importance of data analytics, especially in telecommunications. Here, different data analytics define key performance indicators (KPIs), from operations to customer experience and profitability. Telecom providers need constant monitoring to reduce costs and optimize revenue streams, where KPIs play a crucial role. Data analytics allows identifying problems, making improvements and capitalizing on growth opportunities, resulting in increased sales [21], higher customer retention and reduced fraud. KPIs are fed by data analytics, which provides the information needed to respond to them. This enables a holistic view of the organization, facilitating the identification of problems and opportunities for improvement, and the implementation of solutions that respond to customer needs.

The 3GPP submission to ITU-R also included a self-assessment of 5G performance against all IMT-2020 KPIs [21]. However, it does not include the detailed procedures and examples needed to understand how the results are obtained, therefore the development of this toolkit responds to the need to obtain KPIs from real time data for 5G systems by means of DL optimizing CEs and therefore improving system capacity and quality of service, which would lead us to take the next step towards 6G networks.

The aim of this paper is to present a set of tools designed to facilitate the process of evaluation and optimization of CE in mobile communications. The tool incorporates

DL techniques such as CAE, CNN and DnCNN, in order to achieve an approximation to 6G networks and to address the mentioned problems of traditional techniques. The challenge is to capture accurately the channel responses in the presence of phenomena such as attenuation, multipath losses, noise and delays that affect the signal transmission in a real scenario. The tool implemented in MATLAB, facilitates real-time evaluations covering line-of-sight (LoS) and non-line-of-sight (LoS) environments using a tapped delay line (TDL) model. It evaluates the performance of a Single Input Single Output (SISO) system against challenging scenarios that enable future KPI analysis from data-driven telecommunications analysis using metrics such as bit error rate (BER), error vector magnitude (EVM), estimation time and mean square error (MSE).

2 Materials and Methods

2.1 Communications System Architecture

The toolkit emulates scenarios of ODFM data transmission based on actual Physical Downstream Link Shared Channel (PDSCH) model of 5G. Figure 1 illustrates the layout of the proposed framework.

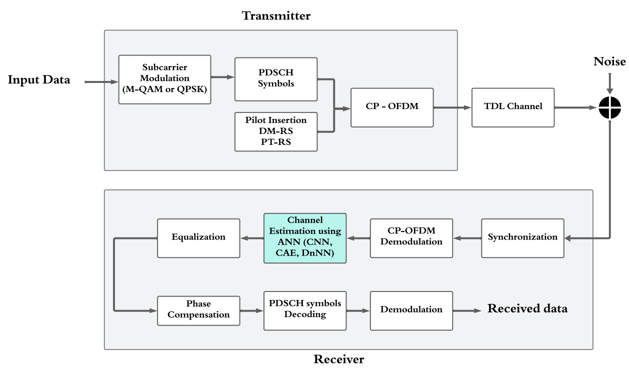


Fig. 1: 5G OFDM proposed block diagram

The process begins with the generation of binary data on the transmitter end, which is then sent to the modulation block. Here, it undergoes transformation into various subcarrier modulation schemes specified by the 5G standard [17], including QPSK, 16QAM, 64QAM, and 256QAM. Following modulation, these symbols are organized into a resource grid, which acts as a two-dimensional representation of time and frequency domains. Notably, this work employs piloted CE, necessitating the allocation of Demodulation Reference Signals (DM-RS) and Phase Tracking Reference Signals (PT-RS) within the resource grid. For a visual depiction of the distribution of drivers and symbols within the resource grid, refer to Fig. 2.

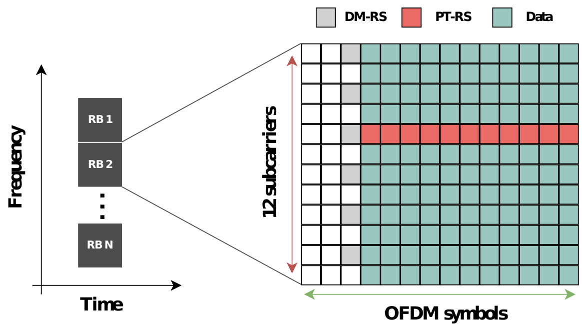


Fig. 2: Pilots and data allocation in the resource grid

Data transmission occurs in compact segments termed subframes, each lasting 1 millisecond. As depicted in the preceding illustration, the Resource Block (RB) comprises 12 subcarriers, with the quantity of RBs within a subframe adaptable to application contexts. The determination of the number of OFDM symbols N_{sym} within a subframe conforms to the OFDM numerology stipulated by the 5G standard. Furthermore, a slot consists of 14 OFDM symbols and a frame encompasses 10 slots.

Within the transmitter's progression lies the CP-OFDM procedure. Initially, a numerology, configuring carrier parameters like subcarrier spacing (SCS) and Cyclic Prefix (CP), must be selected. SCS denotes the interval between adjacent subcarriers and is determined by Equation 1, where μ ranges from 0 to 4. CP serves as a guard interval countering Inter-Symbol Interference (ISI) from multi-path propagation. This involves appending a copy of a symbol's final samples at the beginning of subsequent ones to preserve subcarrier orthogonality. CP types encompass *extended* (for $\mu = 2$) and *normal* (for other μ).

$$\mathrm{SCS} = 2^{\mu} \cdot 15 \text{ kHz} \tag{1}$$

The CP-OFDM block yields a baseband signal. The Inverse Fast Fourier Transform (IFFT) translates modulated data from resource grid $X(k)$ to $x(n)$ (Equation 2), where k denotes subcarrier index, l signifies time index, and N represents subcarrier count.

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi kl}{N}} \tag{2}$$

The base band generated signal is transmitted through the wireless channel. In this paper, the channel model is selected from the 3GPP standard [18]. The channel includes the effects of multi-path and Doppler shifting, which causes frequency and time selective fading, respectively. In particular, the channel is a Tapped Delay Line (TDL) model, which applies a signal processing technique to generate variable time delays in the transmitted signal $x(n)$, each variation has different amplitude and phase. TDL supports Non-Line-of-Sight (NLoS) and Line-of-Sight (LoS) scenarios. The delay profiles TDL-A, TDL-B, and TDL-C are used to represent NLoS; while, TDL-D and TDL-E have LoS conditions.

The Doppler shift experienced over the wireless channel is influenced by both the level of mobility and the carrier frequency. This relationship is defined by the maximum Doppler shift f_d as expressed in Equation 3.

$$f_d = \frac{v\,\frac{1000}{3600}}{c_0}\, f_c \tag{3}$$

Where v denotes the user's velocity in meters per second, f_c represents the carrier frequency in hertz, and $c_0 \approx 3 \times 10^8$ stands for the speed of light. In static conditions, $f_d = 0$, yet with increasing values of v and f_c, the Doppler shift escalates.

Following transmission across the channel, noise is introduced, resulting in the received OFDM signal $y(n)$:

$$y(n) = h(n) \circledast x(n) + \eta(n) \tag{4}$$

Upon time synchronization, the receiver conducts a Fast Fourier Transform (FFT), yielding:

$$Y(k) = H(k) \circledast X(K) + \eta(K) \tag{5}$$

Here, $Y(k)$ and $X(k)$ represent the received and transmitted symbols in the kth subcarrier respectively, $H(k)$ signifies the channel impulse response in the kth subcarrier, and $\eta(K)$ denotes Additive White Gaussian Noise (AWGN). Pilot signals are extracted from the frequency-time domain signal to facilitate CE. Subsequently, after estimating the channel response, the received signal undergoes equalization and phase compensation, crucial processes for mitigating channel distortion and aligning the phase between transmitted and received symbols. Data retrieval from the resource grid is followed by demodulation via the demodulation scheme, corresponding to the modulation utilized at the transmitter. Consequently, the proposed block diagram outputs a binary data sequence.

2.2 Channel Estimation Techniques

CE is a crucial step within modern wireless communications systems. Such techniques involve the characterization of a mathematically modeled channel. This characterization encompasses factors such as attenuation, propagation delay, and impulse response denoted as $H(k)$. CE methods are categorized into Pilot-Aided CE (PACE), Blind CE (BCE), and Decision Directed Channel Estimation (DDCE) [11]. This paper will focus in PACE techniques, especially LS estimation, which is widely used in wireless systems due to low complexity. Figure 3 illustrates the process made with PACE.

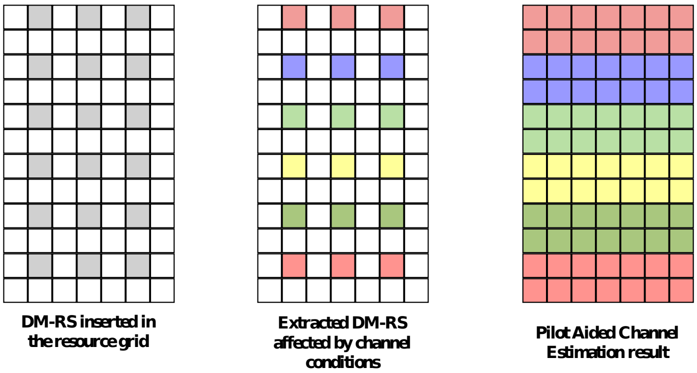


Fig. 3: Pilot Aided Channel Estimation

The LS estimator operates under the assumption that alterations in transmitted signals follow linear equations, leading to the acquisition of channel coefficients through linear interpolation. While the LS algorithm is straightforward, its precision diminishes notably in scenarios with elevated noise levels and mobility. Although Equation 5 describes the received signals in LS, it overlooks the noise contribution $\eta(K)$. To determine the channel response, $H(K)$ is cleansed, and the pilots are extracted using the expression:

$$\hat{H}_{LS} = X^{-1}Y = \left(\frac{X_p}{Y_p}\right)^T \quad \text{where} \quad p = 0, 1, 2, \cdots, N-1 \tag{6}$$

Here, Y represents the output pilot vector $Y_p = [y_0, y_1, y_2, \cdots, y_{N-1}]^T$, T denotes the transpose operation, X is the diagonal matrix containing the transmitted pilots, N signifies the number of pilots inserted across the time domain, and $\hat{H}_{LS}$ symbolizes the estimated channel response.

2.3 Available Models for CE in GUI using DLS5G

The developed tool is called Deep Learning Simulations 5G (DLS5G). The main objective of this proposal is to present an emulation of a real environment capable of integrating DL techniques in the 5G CE process in view of the new functionalities proposed for 6G networks. For this purpose the developed toolkit includes three classic and three DL methods for CE.

The available classic estimators that works for the proposed communications system architecture in Fig. 1 are :

- *Lineal Estimation :* This function performs a linear estimation of the channel using the interpolated channel grid. It does not involve the use of neural networks or other machine learning methods.
- *Practical Estimation :* Is performed using the `nrChannelEstimate` method provided by the MATLAB wireless communications package [22] this method is based on LS estimation. However, LS estimation can have the disadvantage that its ability to remove noise is not always sufficient, this practical estimation method uses the estimated channel impulse response (CIR) to counteract the noise.
- *Ideal Estimation :* Is performed using the `nrPerfectChannelEstimate`.The function first reconstructs the channel impulse response from the channel path gains and the path filter impulse response [23].

2.3.1 CNN Networks

CNN offers a solution to CE since they allow working with higher dimensional data, as is the case of the resource grid. The importance in its selection lies in the convolution operation between the filter and the channel sample, which allows identifying patterns in the channel regardless of the conditions it has (noise, Doppler shifting, fading, among others), managing to mitigate the effects present in the pilot signals generated by the transmission medium. In the toolkit there are 8 CNN type networks, each one, has variations in parameters such as size of the convolutional filter, the number of filters, padding size and the number of layers. Figure 4 presents the best modeled architecture called CNN6, which has three convolutional layers with the Rectified Linear Union (ReLU) function, additionally, the padding size is varying across the layers with the aim of guarantee the correct output dimension.

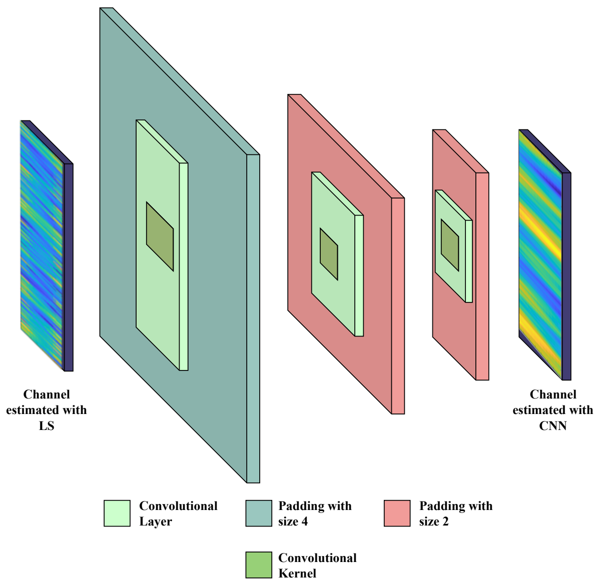


Fig. 4: Layers of the CNN

2.3.2 Convolutional Autoencoders

The CAE is a model that is capable of reducing the dimensions by extracting main features of the input and from these, recreates an output closer to the ideal output, thus eliminating redundancies in the data such as noise, translation or rotation. An advantage of this model over a standard CNN is that it is more robust because it compresses the information, resulting in a better prediction quality, but with a higher computational cost for its training. The architecture of CAE is shown in Fig. 5, there it can be identify two stages: coding and decoding. Coding layers are used to transform the input image into a compressed representation, here the size of filters are varied with the purpose of reduce the spatial dimension. In the output of the coding stage, it is obtained a set of main features. The decoding stage is in charge of reconstructing the image based on the latent space, at this point, it is necessary to use up-sampling layers, in order to reach the initial dimension.

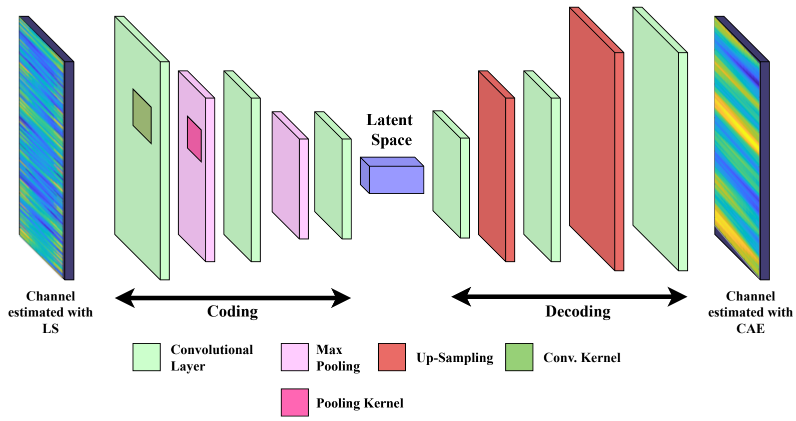


Fig. 5: Layers of the CAE

2.3.3 Denoising Networks

A DnCNN is a special model whose main functionality is to eliminate noise in images, so for this work it is useful in mitigating this effect on the estimated channel. A key feature of these networks is the use of residual connections, whose purpose is to include direct paths between the layers of the network, to avoid data degradation during the forward propagation. These DL algorithms have a combination of convolutional and batch normalization layers, so they have greater depth compared to previously designed models. Figure 6 illustrates the DnCNN architecture.

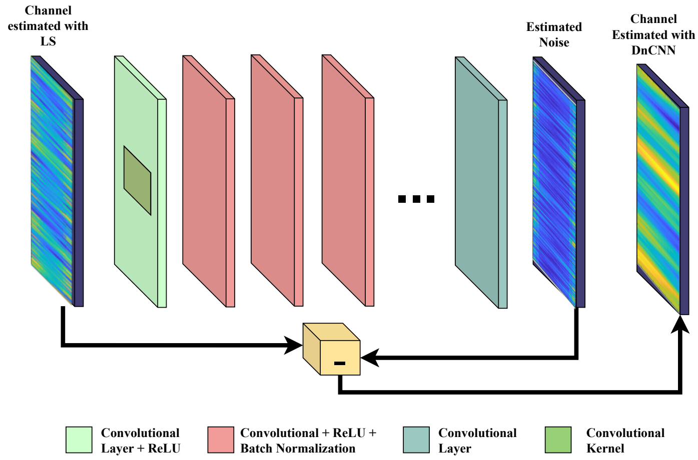


Fig. 6: Layers of the DnCNN

Since the model has a large number of layers, with greater robustness and complexity, it was decided to perform the training with two scenarios:

- Taking the generated X and Y data to learn the perfect estimated channel representations, similar to the process applied with standard CNNs and autoencoders. This model is going to be referred to throughout the paper as *Denoising 1*
- The other way it is trained is by varying the labels so that the model does not estimate the channel, but rather the noise. The data X has a noise level of η, the new labels are obtained by applying the subtraction $X - Y$ implying that the new labels contain the noise η. In the output, $Y = X - \eta$ is done to reduce the noise and optimize the quality of the estimation. It is referred to as *Denoising 2*.

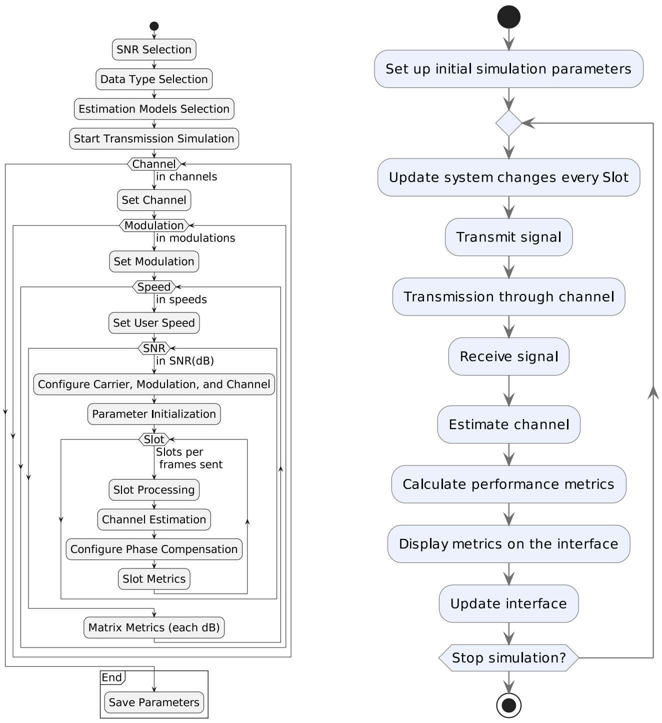
2.4 Data Structure and Modular Approach

In order to present a dynamic and adaptable tool, the development was carried out by independent functions, which allows migration to other programming languages such as Python to obtain better performance. Matlab was used as a first version for comparison with classical models using its 5G toolbox functions.

Within the toolkit, there exists a data structure containing all elements, denoted as *"Pam_sim"* this structure is stored in the main application object *"app"*. Modularity poses a significant challenge in terms of global control of variables. The ***"Pam_sim"*** structure addresses this issue, ensuring that the output of each function always returns the structure with the changes or new variables used in the function.

2.5 Processes and services

DLS5G includes two forms of simulation, the first one is called *"Simple Simulation"* and the other one is called *"Real Time"*. The first form enables three ways to be executed, namely: by using a script, through the Graphical User Interface (GUI) designed without display of metrics during simulation (less load on the system and faster results) and the last using GUI for full visualization of the simulation process. These forms could be considered as closed form since the user cannot interact with the process once the simulation has started. On the other, the "real time" simulation-emulation allows the user to interact with the system variables at any time and emulates a determinate transmission. The flowcharts that describe the *simple* and the *real time* operation are shown in Fig. 7.



(a) Simple simulation with or without GUI

(b) Real time emulation using GUI

Fig. 7: Toolkit process flow charts

2.6 Data Modeling Design

Data modeling design in the mobile communication CE application plays an important role in efficiently representing and manipulating the information necessary for the CE process. In this section it is described how the data is structured and represented in the toolkit. Figure 8 shows the directory layout used for the execution of the toolkit. There are 8 main folders distributed in 4 sections. Each of these folders, in turn, contains supplementary files and directories required by the APP. The toolkit includes a storage area that also functions as a database with which to graph the results obtained.

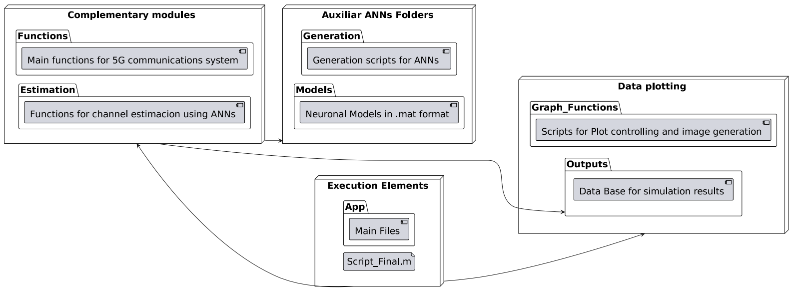


Fig. 8: General structure of app directories

In the *Generation* directory are the generator scripts of several networks and inside the *Models* directory are these models in format .mat, so it is possible to add any other CNN, CAE, DnCNN network that follows the layers deployment presented in the sections 2.3.1, 2.3.2 and 2.3.3.

Figure 9 presents the use case diagram of the application, there are 3 ways to use the developed tool, the first one is by means of the execution of the *"script_final"* in the MATLAB environment, the second one is to use the GUI by a tool *"Simple Simulation"* both of them implement the same algorithm described in Fig. 7a, the last way is by means of the GUI in the tab of *Real Time* this service works by means of a while cycle allowing the interaction of the user on the system of communications in real time, the algorithm is described in Fig. 7b

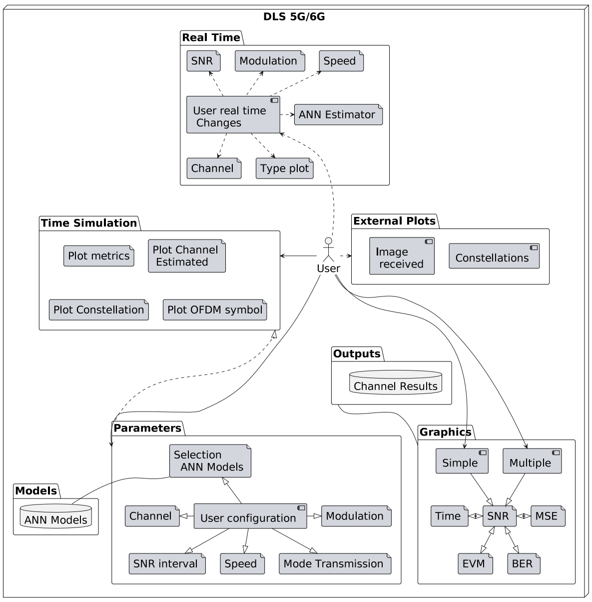


Fig. 9: Use case diagram of the DLS5G app

2.7 Evaluation Metrics Considerations

The calculations of MSE and EVM are performed for each processed slot. Per iteration, it will have a vector of size $(10\,\frac{\text{subframes}}{\text{frame}} \times N\,\text{frames}) \times \text{len(SNR)}$. For the graphs presented in the interfaces, it is not necessary to calculate this metric for each slot. Therefore, an average value is obtained over the transmission time in each iteration of the SNR sweep. For example, if the transmitted data is 7 frames over a OFDM system with numerology $\mu = 0$, the transmission of these frames is done for all the SNR values and the average value per iteration is graphed.

The term *estimation time* for the authors refers to the normalized value that takes to execute the CE instruction within the MATLAB processing sequence. This value is obtained using the *tic* and *toc* functions. When referring to the normalized value, it means it is a value brought to the theoretical realm, as the use of these functions allows seeing the execution time of the instruction based on the machine responsible for simulating the program. To avoid biasing the results to the processing capacity of the equipment used, this value was normalized considering the execution time of the CE of the ideal model as 1ms for $\mu = 0$. This is obtained by dividing the entire output time vector of the *tic* function by the vector of ideal estimation time, so it could be expressed as:

$$t_{norm} = \frac{t_{ANN}}{t_{ideal}} \tag{7}$$

2.8 Emulation of a real scenario using DLS5G toolkit

As previously mentioned, the toolkit models a real environment, therefore real scenarios will be presented in which the toolkit will be able to emulate the described environment and obtain performance metrics through its GUI. For the emulation all the available models will be used in order to contrast them and obtain dynamic and optimized methods for the CE, as well as to characterize urban environments with diverse channel profiles for mobile communications. Figures 10a and 10b show LoS and NLoS environments, respectively. As it can be seen, the first environment presents a direct path between the user equipment and the base station, even, the maximum Doppler shift is zero because the user is not moving. The aforementioned condition allows the use of high transmission rates. The NLoS shows a more challenging environment with fluctuating speeds and different obstacles between the transmitter and the receiver. The channel can be modeled with all the TDL power delay profiles. As observed, both the speed and SNR are constantly changing.

(a) LoS scenario

(b) NLoS scenario

Fig. 10: Types of environments in communications

Due to the conditions described for the scenarios, in this paper, the *simple simulation* is used to describe the first environment, while, the other is emulated in *real time*. The conditions are listed below.

- For simple simulation the power delay profile used is TDL-E, which has LoS parameters, the subcarrier modulation is 256QAM, the relative movement of the receiver is fixed in low-speed values, between 1 km/h and 5 km/h, and the SNR is equal to 10 dB.
- For the real time emulation the power delay profiles used are all the supported by 5G standard (TDL-A to TDL-E), the data is transmitted over 256QAM symbols, the speed is fixed in the range from 30 km/h to 100 km/h, and the levels of noise are changed between 0 dB and 30 dB.

3 Results and discussions

The toolkit has a total of five GUIs, the first one is shown in Fig. 11, which allows the setting of the communication system conditions, selection of models to be used for CE and the type of data to be transmitted. Both scenarios include sending images in order to emulate a real transmission. It is worth mentioning that in this first version of the toolkit, the communication is performed at the physical layer level, then, the toolkit also includes the functionality of sending a customized number of bits or frames.

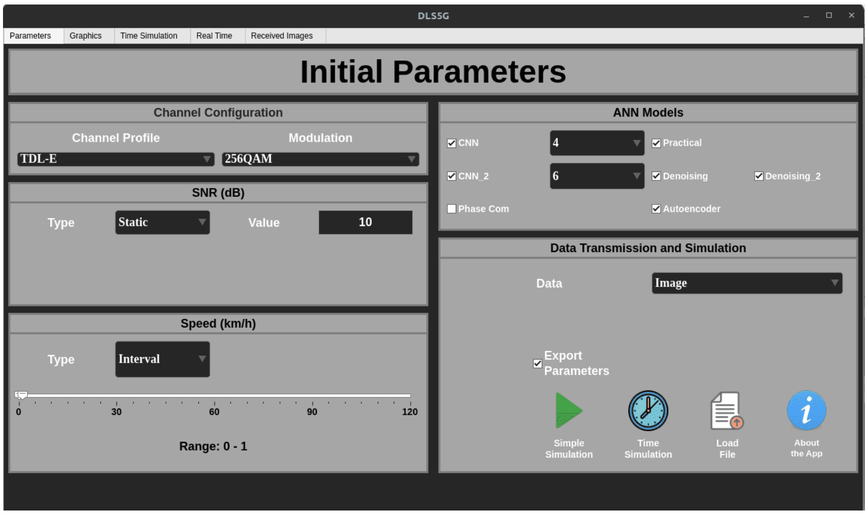


Fig. 11: Parameter configuration GUI for LoS scenario

Figure 12 shows the results obtained for the case shown in Fig. 10a, the graphs on the left correspond to the estimation of channels using CNN, CAE, and DnCNN, on the right there is a table with the values obtained for the last frame in its last transmitted slot. These values are displayed as the simulation progresses and change color dynamically being blue the best metric among the three that are being displayed and red the worst. Finally, a graph is presented with the information of a single OFDM symbol as a function of the subcarriers in this case shows the third OFDM symbol estimated by the different models.

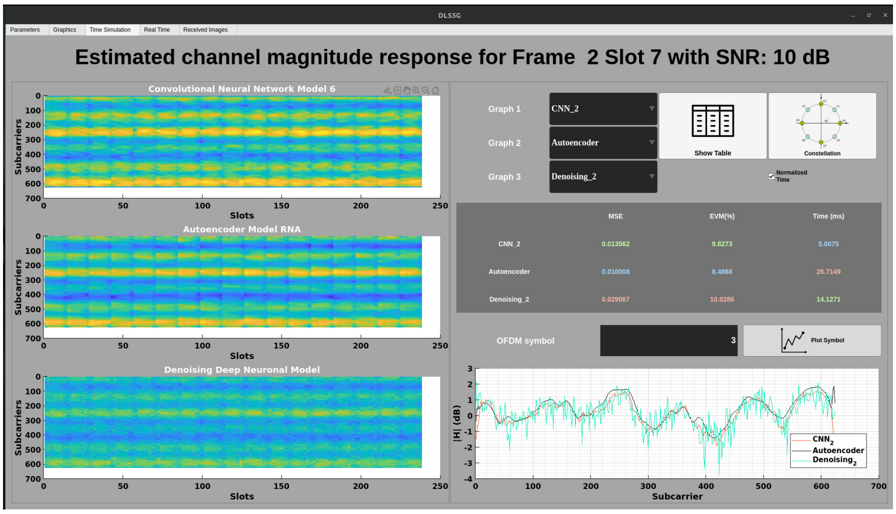


Fig. 12: Estimated Channel with CNN, CAE, and DnCNN

Figure 13 shows another GUI available within the developed toolkit. This part of the tool enables the visualization of the performance metrics of the executed simulation. In particular, it allows to observe in greater detail the results obtained as well as to contrast in the same graph with other simulations. This section has a database that characterizes the behaviors of all models for all channel profiles. A result obtained for a TDL-E channel with 256QAM modulation and a user's speed of 5 km/h is appreciated, as described in the scenario 10a, with the difference that it let the user to see the response of the proposed system across a range of SNR values.

Figure 13 illustrates the BER, EVM, MSE and estimation time metrics. The estimation of each model shows that the BER gradually decreases as the SNR increases. The modulation format used for transmission affects the received bits, this occurs due to the noise robustness of 256QAM because the closer the symbols are, the worse the response in noisy conditions is obtained.

In the range between 0 dB and 15 dB, all estimators obtained a BER value very close to the ideal, where the DnNN network models stand out slightly. After that limit and up to 25 dB, the LS improves gradually, which leads to be the best model, with a slight difference is the best neural model DnNN 2 and the practical estimator (PE), in general the models have a very good response indicating an accurate estimation in high and low noise conditions.

The EVM metric can be analyzed in 2 ranges, the first ranging from 0 dB to 15 dB where the results are good and close to ideal, the second interval up to 25 dB shows the incidence of scattering of the symbols, in this interval the LS models, DnNN 2 and PE obtained the best results, very close to the ideal. On the other hand the symbols estimated by the other models present greater dispersion leading to the observed irregular peaks which demonstrates a great inference of the noise reduction in the metric of EVM that is not improved apart from the 12 dB. This can be explained by the fact that the loss function has been determined to minimize errors in estimation rather than the EVM measurement. As for the MSE metric, all models share the same trend of MSE reduction as SNR increases, all estimate tend to zero. The speed effect does not affect the quality of the estimation, for each model both low and high speed curves have no significant changes, indicating that the SSM is independent of the maximum Doppler displacement. The linear estimator obtained the worst results up to 5 dB, being approximately 6 and 12 times higher compared to CAE and PE, respectively. The DnNN, CNN_2 outperformed the LS estimation over the entire simulation range and showed practically ideal results.

Finally, the estimation time is a valuable metric as it allows to visualize how a model with near ideal results such as the DnNN networks have a higher computational cost, the fastest and closest models to the practical estimation are the CNN and CAE. The LS estimator requires more time to generate an output, while the PE is faster. The CAE and CNN estimates the channel with 256QAM in an average time of 3.54 ms. The estimation time in CAE is less compared to LS, this shows the importance of max pooling layers, which reduce the complexity in the model.

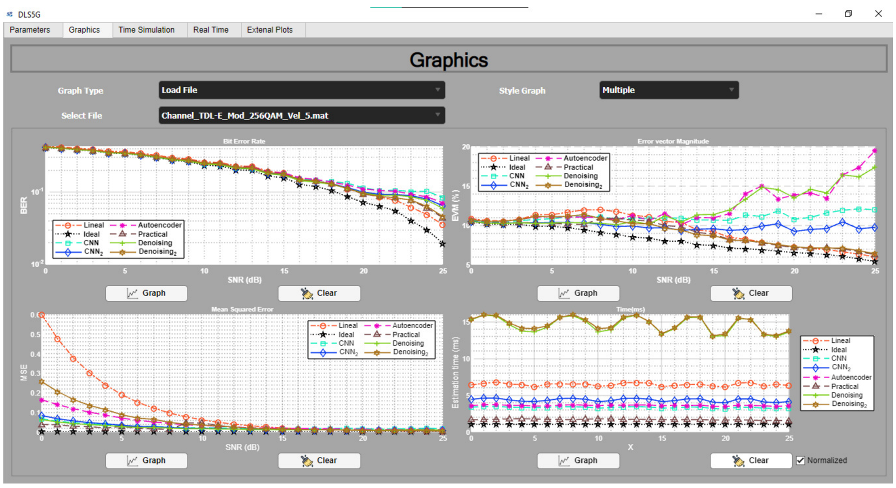


Fig. 13: Results of performance metrics

Continuing with the simulation, Fig. 14 presents the transmitted image and the set of received images based on the use of different CE techniques: ideal, practical, LS, and DL models. The results probe a slight superiority of the DL techniques offering a better estimation.

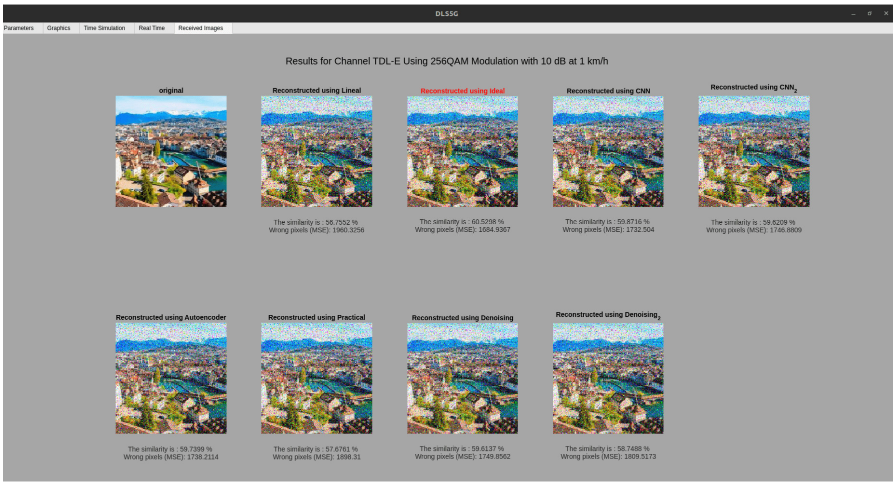


Fig. 14: Received images using DL estimators for a LoS scenario

Figure 15 presents the emulation of the scenario described in Fig. 10b, it corresponds to a variation of channel profiles for a user with a high-speed mobility immersed in a noisy environment. Due to variation in the power delay profiles, the image is divided into different colors, each color represents an specific power delay profile. Additionally, for the CE, two DL models were deployed: Denoising 2 and a CNN. With the aim of identify the response of the channel estimators, the color box was divided into two line styles: dashed for Denoising 2 and continuous for CNN. Furthermore, the real time variations can also be seen with the EVM and time estimation metrics. The scenario was emulated for 44 subframes, which corresponds to a transmission time of 44 ms using numerology $\mu = 0$. At the end of the emulation time, some changes were included with the purpose of showing the adaptability and response capacity of the designed DL models with different channel conditions, those changes include: the increase of the user's speed to 100 km/h, then the increase in the noise power until reach a SNR equal to 0 dB; finally, the last variation consists in the reduction of the noise and mobility levels, where the values were fixed in 30 dB and 30 km/h, respectively.

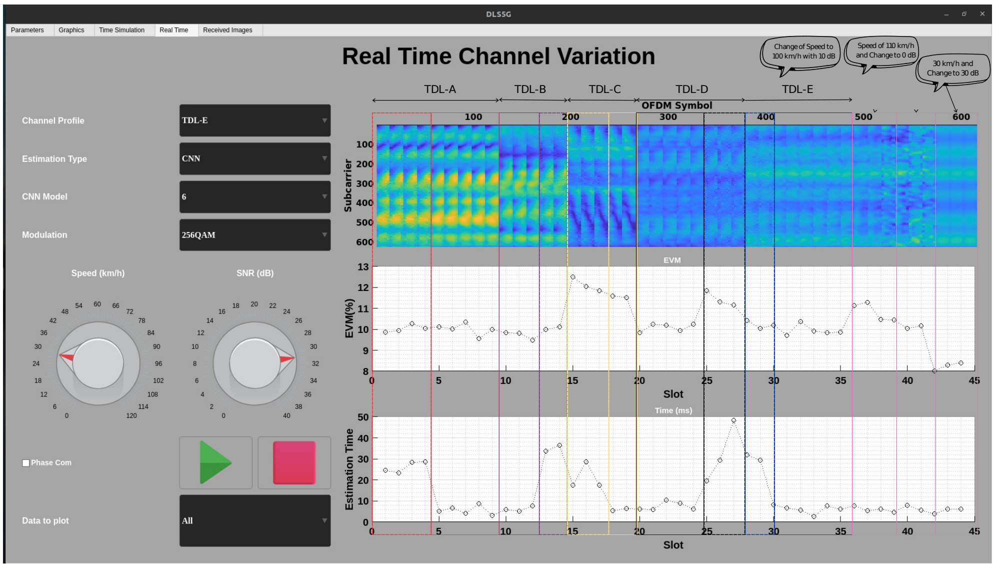


Fig. 15: Real-time TDL CE emulation

This emulation allows to analyze the strong impact of the change of TDL channel profile for metrics such as estimation time and EVM, being the TDL-C channel the one that presents a significant increase in the time required for the CE. This is due to the fact that, as shown in the EVM metric, the dispersion between symbols increased considerably. As for user speed, it is evident that it has a substantial impact on EVM, especially in extreme motion scenarios. However, the estimation models show an almost ideal ability to estimate the channel even at speeds above 100 km/h, indicating a considerable robustness of the system to dynamic conditions.

4 Conclusion

This paper presented a toolkit for the evaluation of deep learning techniques for channel estimation in 5G networks and beyond in NLoS and LoS environments. The obtained results reveals that the TDL channel can introduce significant distortions in the received signal, leading to increased EVM. As expected, neural models demonstrated superior performance in LoS scenarios due to the direct signal propagation path, showcasing optimal adaptability to diverse channel profiles. This suggests that neural models possess a high generalization capability, with a single model effectively operating across various environments. The toolkit facilitates efficient, consistent and precise recording of metric fluctuations concerning changes in the current 5G communication channel, employing estimators based on CNN, CAE and DnCNN. It enables continuous monitoring and optimizes the core process of channel estimation through Deep Learning, resulting in enhanced KPI metrics to mitigate costs, thereby presenting a significant stride towards sixth-generation network development. Overall, these findings affirm the efficacy and resilience of the communication system engineered within the DLS5G toolkit, demonstrating the neural estimation models' capacity to adapt and furnish accurate channel estimates across diverse operational conditions. These results not only herald the prospect of real-time system monitoring, but also point to areas for future improvements and optimizations. The metrics results show considerable performance in terms of EVM, MSE and BER, outperforming traditional techniques such as LS and PE. Notably, MSE performance exhibits negligible variations compared to PE across the 0 dB to 20 dB range. These findings reinforce the claim that deep learning models boast substantial generalization ability and robustness, effectively estimating channels with precision and low latency in 5G networks and beyond.

Acknowledgments. The authors wish to acknowledge and thanks the Universidad Distrital Francisco José de Caldas for supporting the development of this paper.

Conflicts of interest. The authors declare no conflicts of interest.

References

[1] Sudhamani, C., Roslee, M., Tiang, J.J., Rehman, A.U.: A survey on 5g coverage improvement techniques: Issues and future challenges. Sensors **23**(4) (2023) https://doi.org/10.3390/s23042356

[2] Pana, V.S., Babalola, O.P., Balyan, V.: 5g radio access networks: A survey. Array **14**, 100170 (2022) https://doi.org/10.1016/j.array.2022.100170

[3] Valgas, J., Monserrat, J.F., Arslan, H.: Flexible numerology in 5g nr: Interference quantification and proper selection depending on the scenario. Mobile Information Systems **2021**, 6651326 (2021) https://doi.org/10.1155/2021/6651326

[4] Jiang, X., Hu, Z., Wang, S., Zhang, Y.: Deep learning for medical image-based cancer diagnosis. Cancers **15**(14) (2023) https://doi.org/10.3390/cancers15143608

[5] Dellal-Hedjazi, B., Alimazighi, Z.: Deep learning for recommendation systems. In: 2020 6th IEEE Congress on Information Science and Technology (CiSt), pp. 90–97 (2020). https://doi.org/10.1109/CiSt49399.2021.9357241

[6] Alkhudaydi, O.A., Krichen, M., Alghamdi, A.D.: A deep learning methodology for predicting cybersecurity attacks on the internet of things. Information **14**(10) (2023) https://doi.org/10.3390/info14100550

[7] Bhalla, A., Nikhila, M.S., Singh, P.: Simulation of self-driving car using deep learning. In: 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), pp. 519–525 (2020). https://doi.org/10.1109/ICISS49785.2020.9315968

[8] Sarker, I.H.: Machine learning: Algorithms, real-world applications and research directions. SN Computer Science **2**(3), 160 (2021) https://doi.org/10.1007/s42979-021-00592-x

[9] Wong, K.J., Juwono, F.H., Reine, R.: Deep learning for channel estimation and signal detection in ofdm-based communication systems. ELKHA **14**(1), 52–59 (2022)

[10] Mohammed, A.S.M., Taman, A.I.A., Hassan, A.M., Zekry, A.: Deep learning channel estimation for ofdm 5g systems with different channel models. Wireless Personal Communications **128**(4), 2891–2912 (2023) https://doi.org/10.1007/s11277-022-10077-6

[11] Dash, L., Thampy, A.S.: Channel estimation using hybrid optimizer based recurrent neural network long short term memory for mimo communications in 5g network. SN Applied Sciences **5**(2), 60 (2023) https://doi.org/10.1007/s42452-022-05253-z

[12] Gupta, R., Gupta, J.: Channel estimation of 5g new radio using convolutional neural network. In: 2022 8th International Conference on Signal Processing and Communication (ICSC), pp. 58–62 (2022). https://doi.org/10.1109/ICSC56524.2022.10009096

[13] Wang, Y., Chang, J., Lu, Z., Yu, F., Wei, J., Xu, Y.: Channel estimation of 5g ofdm system based on convlstm network. In: 2022 7th International Conference on Communication, Image and Signal Processing (CCISP), pp. 62–66 (2022). https://doi.org/10.1109/CCISP55629.2022.9974588

[14] Yamada, Y., Ohtsuki, T.: Autoencoder-based pilot pattern design for cdl channels. In: 2022 27th Asia Pacific Conference on Communications (APCC), pp. 76–80 (2022). https://doi.org/10.1109/APCC55198.2022.9943621

[15] Thakkar, K., Goyal, A., Bhattacharyya, B.: Deep learning and channel estimation. In: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 745–751 (2020). https://doi.org/10.1109/ICACCS48705.2020.9074414

[16] Wang, M., Wang, A., Liu, Z., Chai, J.: Deep learning based channel estimation method for mine ofdm system. Scientific Reports **13**(1), 17105 (2023) https://doi.org/10.1038/s41598-023-43971-5

[17] Alsabah, M., Naser, M.A., Mahmmod, B.M., Abdulhussain, S.H., Eissa, M.R., Al-Baidhani, A., Noordin, N.K., Sait, S.M., Al-Utaibi, K.A., Hashim, F.: 6g wireless communications networks: A comprehensive survey. Ieee Access **9**, 148191–148243 (2021)

[18] Shaddad, R.Q., Saif, E.M., Saif, H.M., Mohammed, Z.Y., Farhan, A.H.: Channel estimation for intelligent reflecting surface in 6g wireless network via deep learning technique. In: 2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA), pp. 1–5 (2021). https://doi.org/10.1109/eSmarTA52612.2021.9515730

[19] Ohtsuki, T.: Machine learning in 6g wireless communications. IEICE Transactions on Communications **106**(2), 75–83 (2023)

[20] Shi, Y., Lian, L., Shi, Y., Wang, Z., Zhou, Y., Fu, L., Bai, L., Zhang, J., Zhang, W.: Machine learning for large-scale optimization in 6g wireless networks. IEEE Communications Surveys & Tutorials **25**(4), 2088–2132 (2023) https://doi.org/10.1109/COMST.2023.3300664

[21] Fuentes, M., Carcel, J.L., Dietrich, C., Yu, L., Garro, E., Pauli, V., Lazarakis, F.I., Grøndalen, O., Bulakci, O., Yu, J., Mohr, W., Gomez-Barquero, D.: 5g new radio evaluation against imt-2020 key performance indicators. IEEE Access **8**, 110880–110896 (2020) https://doi.org/10.1109/ACCESS.2020.3001641

[22] MathWorks: `nrChannelEstimate`. https://la.mathworks.com/help/5g/ref/nrchannelestimate.html

[23] MathWorks: `nrPerfectChannelEstimate`. https://la.mathworks.com/help/5g/ref/nrperfectchannelestimate.html