

Utilização de redes neuronais para estimação de canal nas futuras redes 6G

Ricardo Alexandre Cajado Gaspar

Mestrado em Engenharia de Telecomunicações e Informática

Orientador(a):

Doutor João Carlos Marques Silva, Professor Auxiliar,
ISCTE-IUL

Co-Orientador(a):

Doutor Nuno Manuel Branco Souto, Professor Associado com
Agregação,
ISCTE-IUL

Outubro, 2023

Departamento de Ciências e Tecnologias da Informação

Utilização de redes neuronais para estimação de canal nas futuras redes 6G

Ricardo Alexandre Cajado Gaspar

Mestrado em Engenharia de Telecomunicações e Informática

Orientador(a):

Doutor João Carlos Marques Silva, Professor Auxiliar,
ISCTE-IUL

Co-Orientador(a):

Doutor Nuno Manuel Branco Souto, Professor Associado com
Agregação,
ISCTE-IUL

Outubro, 2023

Agradecimentos

Em primeiro lugar, é com uma imensa gratidão que dedico estas palavras de agradecimento à minha família, o alicerce primordial e essencial que me permitiu a realização desta dissertação. Aos meus pais, agradecer a dedicação, encorajamento e esforço que me permitiram realizar este curso, a eles dedico esta conquista. À minha irmã, que sempre esteve do meu lado, agradeço a compreensão e consciência do desafio que foi este percurso. Reconheço que foi o meu porto seguro e confiante, e que me incentivou a nunca desistir. A ela o meu muito obrigado!

Não posso deixar de referenciar obviamente o papel dos meus dois orientadores, o professor João Silva e o professor Nuno Souto, que me deram orientação e apoio ao longo do desenvolvimento da dissertação. Agradecer toda a ajuda, desde o conhecimento à experiência que me forneceram ao longo de todo este percurso.

Aos meus amigos e colegas, agradecer todos os bons momentos e apoio durante este longo percurso que foi a realização deste curso.

Por fim, quero agradecer a todos o que contribuíram neste projeto, todo o apoio e incentivo foram fundamentais para a realização desta dissertação.

Resumo

A tecnologia 6G será a sucessora da tecnologia 5G. Com a tecnologia 5G, a Modulação por Divisão de Frequência Ortogonal (OFDM) foi adotada e tornou-se um dos principais candidatos para os futuros sistemas de comunicação. Nesse sentido, a principal tarefa é resolver os problemas encontrados na tecnologia 5G, e um dos objetivos desta dissertação é focar na solução desses obstáculos e adaptá-los à tecnologia 6G, usando redes neurais e Deep Learning para inferir a estimação de canal. Para inferir a estimação de canal, foi adotado um modelo Matlab de uma rede neuronal convolucional (CNN), que utiliza símbolos piloto OFDM, de forma a garantir uma transmissão de dados precisa e robusta. A partir desses símbolos é gerado o sinal OFDM, que será transmitido através de um canal TDL, onde depois através de interpolação linear será efetuada a estimação de canal. O OFDM é uma técnica de modulação que tem sido amplamente usada em sistemas de comunicações sem fio, que o tornam desse modo muito atraente para os sistemas de comunicações futuros. Várias são as características do OFDM que podem contribuir para a evolução das comunicações tais como: uma largura de banda eficiente, uma vez que este modelo permite uma eficiente utilização de largura de banda espectral, através da divisão do canal em subcanais ortogonais, o que proporciona a transmissão simultânea de múltiplos canais; a simplicidade na implementação, sendo uma técnica relativamente simples de implementar; a capacidade de suportar altas taxas de transmissões de dados, e por fim a fácil adaptação a canais móveis e dinâmicos, devido à capacidade de dividir o canal, o que facilita a adaptação às mudanças nas condições do canal. Dessa maneira, o OFDM é um forte candidato para o futuro dos sistemas de comunicação, e será utilizado nesta dissertação para ser feita a estimação de canal, como referido anteriormente. Através do uso da rede neuronal do exemplo dado pelo Matlab, foram explorados diferentes tipos de camadas, como as camadas Depthwise, para tentar reduzir a complexidade e verificar se a rede mantém o desempenho em termos de Erro Quadrático Médio (MSE) em função da Relação Sinal-Ruído (SNR). Essas novas redes neurais criadas irão ser comparadas com um estimador prático e com a rede pré-definida pelo exemplo do Matlab que foi usado, considerando também a complexidade de cada rede. No final, será possível verificar que as redes com camadas Depthwise acabam por ter um desempenho muito semelhante ao da rede pré-definida, mesmo sendo menos complexas.

Palavras-chaves: Complexidade, Desempenho, MSE, Estimação de Canal, Deep Learning, Redes Neurais Convolucionais, OFDM.

Abstract

6G technology will be the successor to 5G technology. With 5G technology, Orthogonal Frequency Division Modulation (OFDM) was adopted and became one of the main candidates for future communication systems. In this sense, the main task is to solve the problems encountered in 5G technology, and one of the aims of this thesis is to focus on solving these obstacles and adapting them to 6G technology, using neural networks and deep learning to infer channel estimation. For channel estimation, a Matlab model of a convolutional neural network (CNN) was adopted, which uses OFDM pilot symbols to ensure accurate and robust data transmission. These symbols are used to generate the OFDM signal that is transmitted over a TDL channel, where channel estimation is then performed using linear interpolation. OFDM is a modulation technique that has been widely used in wireless communication systems, making it very attractive for future communication systems. There are several characteristics that can contribute to the evolution of communications, such as: bandwidth efficiency, since this model allows efficient use of spectral bandwidth by dividing the channel into orthogonal subchannels, which allows simultaneous transmission of multiple channels; ease of implementation, since it is a relatively simple technique to implement; the ability to support high data rates and easy adaptation to mobile and dynamic channels, since the ability to divide the channel makes it easier to adapt to changes in channel conditions. In this way, OFDM is a strong candidate for the future of communication systems and will be used in this dissertation for channel estimation as mentioned above. By using the neural network in the example given by Matlab, different types of layers were explored, such as Depthwise layers, to try and reduce complexity and to check that the network maintained its performance in terms of Mean Square Error (MSE) as a function of Signal to Noise Ratio (SNR). These newly created neural networks will be compared with a practical estimator and with the network predefined by the Matlab example used, also considering the complexity of each network. In the end, it will be possible to see that the networks with Depthwise layers ends up performing very similarly to the predefined network, even though they are less complex.

Keywords: Complexity, Performance, MSE, Channel Estimation, Deep Learning, Convolutional Neural Networks, OFDM.

Índice

Lista de Tabelas	xi
Lista de Figuras	xii
Lista de Acrónimos	xiii
1 Introdução	15
1.1 Conhecimento e Motivação	16
1.2 Questões de Pesquisa.....	17
1.3 Objetivos	18
1.4 Metodologia/Processo de Desenvolvimento	19
1.5 Planeamento	20
2 Estado de Arte/ Revisão de Literatura	21
2.1 Conceitos.....	21
2.2 Projetos Relacionados	23
3 Modelo do Sistema.....	28
3.1 Modelo de Treino	33
3.2 Complexidade e Desempenho	34
3.3 Stride	35
3.4 Padding.....	35
4 Resultados das Simulações.....	37
5 Conclusões e Trabalhos Futuros.....	45
5.1 Conclusões	45
5.2 Trabalhos Futuros.....	46
6 Anexos	47
6.1 Anexo A – Outras Simulações.....	47
6.2 Anexo B – Artigo	54
7 Referências.....	60

Lista de Tabelas

Tabela 1- Diferenças entre o DL e a transmissão sem fios [8].....	17
Tabela 2- Planeamento da realização da dissertação	20
Tabela 3- Complexidade e MSE acerca de todas as redes analisadas	42
Tabela 4- Valores obtidos para a heurística utilizada com um valor de $\alpha = 10 - 2$	43
Tabela 5- Valores obtidos para a heurística utilizada com um valor de $\alpha = 0,99$	43
Tabela 6- Valores obtidos para a heurística utilizada com um valor de $\alpha = 0,5$	43

Lista de Figuras

Figura 1- Roadmap do 6G [1]	16
Figura 2- Design Science Research Methodology (DSRM) [18].....	19
Figura 3-Utilização de blocos pilotos para comparar o desempenho entre o método baseado em DL e os métodos LS e MMSE [5]	24
Figura 4- Remoção do prefixo cíclico (CP) para comparar o método segundo a DNN e os métodos tradicionais [5]	24
Figura 5- Comparação da curva BER entre a DNN e o MMSE com todas as adversidades presentes [5]	25
Figura 6- Diferença entre o modelo LSTM e os métodos convencionais [6].....	26
Figura 7- Analogia entre SF-CNN e SFT-CNN com os métodos MMSE ideal e não-ideal [7]..	27
Figura 8- Curvas NMSE entre os vários tipos de CNN e dos métodos MMSE [7].....	28
Figura 9- Utilização de símbolos piloto para efetuar a estimação de canal [22]	29
Figura 10- Composição da rede base.....	30
Figura 11- Composição de um exemplo de rede com camadas Depthwise	31
Figura 12- Exemplo da composição de uma rede neuronal	32
Figura 13- Gráficos resultantes de treino efetuado na rede.....	33
Figura 14- MSE do estimador prático vs MSE da rede A, ambos com 256 sessões de treino	37
Figura 15- MSE da rede base vs MSE da rede A, ambas com 256 sessões de treino	38
Figura 16- MSE da rede base vs MSE da rede B, ambas com 256 sessões de treino.....	39
Figura 17- MSE da rede B vs MSE da rede A, ambas com 256 sessões de treino	39
Figura 20- MSE da rede base vs MSE da rede C, ambas com 4096 sessões de treino	41
Figura 21- MSE da rede base vs MSE da rede D, ambas com 4096 sessões de treino.....	42
Figura 22- MSE da rede base com Stride	47
Figura 23- MSE da Rede com apenas uma camada Depthwise vs MSE do estimador prático ...	48
Figura 24- MSE da rede com apenas uma camada Depthwise vs MSE da rede base	49
Figura 25- MSE da rede com a utilização de um canal (TDL-B) fora dos utilizados para o treino da rede vs MSE do estimador prático	50
Figura 26- MSE da rede com a utilização de um canal (TDL-B) fora dos utilizados para o treino da rede vs MSE da rede com canal utilizado no treino (TDL-A).....	50
Figura 27- MSE da rede com a utilização de um canal (TDL-E) fora dos utilizados para o treino da rede vs MSE do estimador prático	51
Figura 28 - MSE da rede com a utilização de um canal (TDL-E) fora dos utilizados para o treino da rede vs MSE da rede com canal utilizado no treino (TDL-A).....	52
Figura 29- MSE da rede base com 16384 sessões de treino vs MSE do estimador prático.....	53

Lista de Acrónimos

AI- Artificial Intelligence

AWGN- Additive White Gaussian Noise

BER- Bit Error Rate

CEU- Channel Estimation Unit

CNN- Convolutional Neural Network

CP- Cyclic Prefix

CSI- Channel State Information

DL- Deep Learning

DM-RS- Demodulation Reference Signal

DNN- Deep Neural Network

ISI- Inter-Symbol Interference

LS- Least Squares

LSTM- Long Short-Term Memory

MIMO- Multiple-Input Multiple Output

ML- Machine Learning

MMSE- Minimum Mean Square Error

MSE- Mean Squared Error

NFV- Network Functions Virtualization

NLOS- Non-Line Of Sight

OFDM- Orthogonal Frequency-Division Multiplexing

PDSCH- Physical Downlink Shared Channel

RNN- Recurrent Neural Network

SDN- Software Designed Networking

SF-CNN- Spatial-Frequency-Convolutional Neural Network

SFT-CNN- Spatial-Frequency-Temporal-Convolutional Neural Network

SNR- Signal-to-Noise Ratio

SPR-CNN- Spatial Pilot-Reduced-Convolutional Neural Network

TDL- Tap Delay Line

UMa- Urban Macro

1 Introdução

A recente proliferação de novos dispositivos móveis, alimentados por inteligência artificial, está a suscitar um debate sobre a forma como as redes móveis irão evoluir no futuro. Após a criação da tecnologia 5G, a nova tecnologia 6G está a ser contextualizada com a perspectiva de poder suportar serviços de IA para dispositivos de rede, para além de suportar a Internet móvel. Por outras palavras, a 6G está a ser concebida com um sistema de informação inteligente para impulsionar as modernas tecnologias de IA [1]. Com a tecnologia 5G foi adotada a modulação ortogonal por divisão de frequência (OFDM), e que se tornou um dos principais candidatos para os futuros sistemas de comunicação [21]. Nesse sentido, o principal papel passa por resolver os problemas obtidos na tecnologia 5G, sendo um dos objetivos desta dissertação centrar-se em resolver esses obstáculos e adaptá-los para a tecnologia 6G, com o uso de redes neuronais e de *Deep Learning* de modo a inferir a estimação de canal. Desse modo, será elaborado um programa em *Matlab*, para a estimação de canal nas futuras redes 6G.

O Deep Learning obteve um sucesso impressionante quando aplicado a redes neuronais. Mesmo para redes neuronais complexas que podem ser necessárias para resolver problemas de grande escala, espera-se que as arquiteturas paralelas distribuídas e as ferramentas de aceleração do Deep Learning resultem numa elevada eficiência computacional. Devido a estas vantagens, o Deep Learning está a ser aplicado às comunicações sem fios [8].

Neste documento, serão efetuados vários testes utilizando redes neuronais convolucionais (CNN) para a estimação de canais. Para analisar o desempenho das redes neuronais estudadas, será analisado o erro quadrático médio (MSE) em função do rácio sinal-ruído (SNR) em comparação com um estimador prático.

Existe já uma rede exemplo pré-definida constituída por 5 camadas convolucionais, que servirá de base de comparação com as novas redes testadas, com o objetivo de melhorar a rede neuronal para reduzir a sua complexidade e manter um bom desempenho relativamente a um estimador prático também testado.

No final de todos os testes efetuados, chega-se a uma conclusão sobre quais as melhores redes a utilizar, tendo sempre em conta o nível de complexidade e o desempenho da rede.

1.1 Conhecimento e Motivação

O mais recente aumento de novos dispositivos móveis, suportados por Inteligência Artificial, origina debate sobre como será a evolução das redes móveis no futuro. Após a criação da tecnologia 5G, começou-se a contextualizar a nova tecnologia 6G, com perspectivas de que esta além de suportar internet móvel, poder ainda oferecer suporte a serviços de AI para os dispositivos da rede. Posto isto, conceitualiza-se o 6G com um sistema de informação inteligente de forma a impulsionar as modernas tecnologias de AI.[1]

Em [1], foi representado uma ideia de *roadmap* para a nova tecnologia 6G, projetado com base no 5G, com a previsão de como essa tecnologia irá evoluir ao longo dos anos.

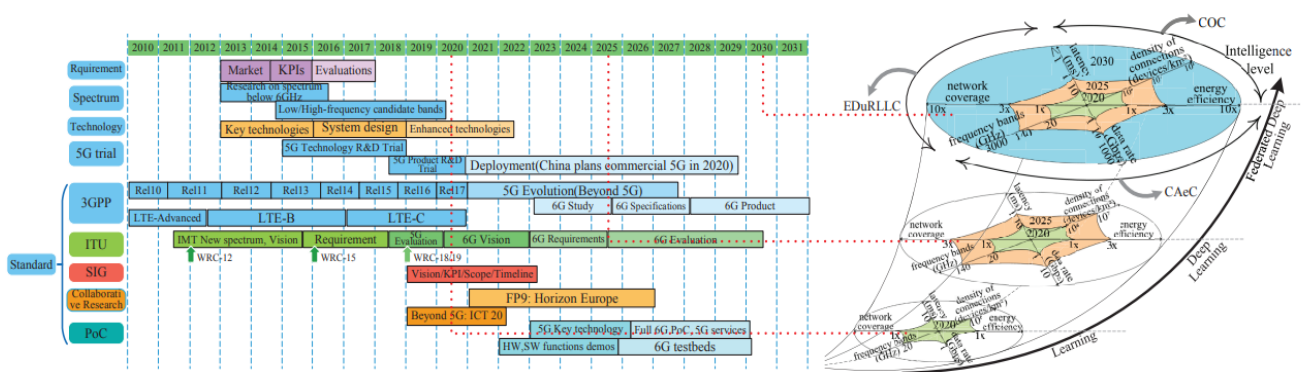


Figura 1- Roadmap do 6G [1]

Na tecnologia 5G deu-se especial importância ao aspeto “não-rádio”, tendo especificamente sido utilizado duas tecnologias chave para o 5G: Redes Definidas por Software (SDN) e funções de rede de virtualização (NFV), que mudou as redes de comunicações modernas para redes baseadas em software [1]. Porém estas tecnologias não serão suficientes para sustentar aplicações baseadas em IA, e dessa forma a nova rede 6G tem de suportar diversos recursos, tais como o aumento da largura de banda, onde se espera obter uma largura de banda na ordem dos Terahertz, por volta dos 0,3-3 THz [2].

Através do uso de técnicas de DL, que engloba diferentes tipos de redes neuronais, como redes neuronais profundas (DNNs), redes neuronais recorrentes (RNNs) e redes neuronais convolucionais (CNNs) é possível efetuar a estimação de canal para as futuras redes 6G. Em [3] foi adotada uma DNN capaz de integrar o DL para um sistema maciço MIMO de modo a verificar a estimação de canal, obtendo através dessa DNN, informação do estado do canal (CSI) em tempo real. Foram ainda fornecidos diferentes tipos de análise de performance, através do cálculo do

erro quadrático médio (MSE) e a taxa de erro de bit (BER) de forma a ter uma melhor precisão sobre a estimação de canal. Os resultados obtidos demonstram que um esquema baseado em DL apresenta uma maior eficiência e robustez que o esquema baseado nos métodos convencionais.

Deep Learning tem alcançado um sucesso impressionante, quando utilizado em redes neuronais. Até para redes neuronais complexas, que podem ser necessárias para resolver problemas em larga escala, as arquiteturas paralelas distribuídas e as ferramentas de aceleração do *Deep Learning* são esperadas para resultar em uma alta eficiência computacional. Devido a estas vantagens, DL é assim usado para comunicações sem fio. Num trabalho realizado em [8], foi elaborada uma tabela com as diferenças entre o DL e a transmissão sem fios.

	Transmissão sem fios	Deep Learning
Modelo matemático	Necessita de modelo matemático preciso	Não necessita de um modelo matemático preciso
Abordagem do projeto	Otimiza cada módulo separadamente com o uso de derivação matemática	Treina os parâmetros da DNN como um todo.
Interpretação	Intuitivo	Não intuitivo
Capacidade de generalização	Amplamente aplicável	Específico de cada aplicação
Desafios chave	Suposições idealizadas e simplificadas no seu modelo matemático	Muitos parâmetros

Tabela 1- Diferenças entre o DL e a transmissão sem fios [8]

1.2 Questões de Pesquisa

No âmbito do tema que irá ser abordado, surgem algumas questões que podem motivar o estudo do tema:

- Qual das diferentes redes neuronais será a mais adequada para a tecnologia 6G?
- Que tipos de algoritmos de *Machine/Deep Learning* serão necessários para efetuar a estimação de canal para as futuras redes 6G?
- Será possível adaptar sistemas OFDM oriundos da tecnologia 5G para a tecnologia 6G?
- Através da implementação de Inteligência Artificial na tecnologia 6G, que tipo de aplicações irão ser possíveis de realizar?

1.3 Objetivos

O principal objetivo desta dissertação passa por estudar e avaliar o uso de redes neuronais juntamente com técnicas de *Deep Learning* como abordagem eficaz para a estimação de canal nas futuras redes 6G. Para isso será desenvolvido um programa em *Matlab* que permita atingir esse objetivo.

Com a utilização das redes neuronais, espera-se que se atinja assim uma melhor precisão da estimação de canal, uma vez que estas são capazes de modelar a distribuição subjacente do canal e assim estimar os coeficientes do canal com alta precisão. As redes neuronais podem ainda ser usadas para integrar a estimação de canal com outras tarefas de comunicação, como modulação e codificação, de modo a melhorar o desempenho geral do sistema de comunicação. Através da correlação das redes neuronais com técnicas MIMO tentar-se-á melhorar a estimação de canal em canais altamente direcionais.

Através da utilização do *Deep Learning* não supervisionado [4], juntamente com as redes neuronais será permitido obter uma melhor precisão da estimação de canal, sendo que o uso de técnicas de *Deep Learning* permitirá uma redução de custos.

De forma geral, irá se verificar quais as redes neuronais e técnicas de *Deep Learning* mais adequadas para estabelecer a estimação de canal para a tecnologia de 6G, de forma a atingir o melhor desempenho do sistema.

1.4 Metodologia/Processo de Desenvolvimento

O método utilizado para a investigação deste trabalho foi a *Design Science Research Methodology* (DSRM) [18]. Este método caracteriza-se como sendo um tipo de pesquisa em desenvolvimento, contribuindo na construção de protótipos e artefactos relevantes.

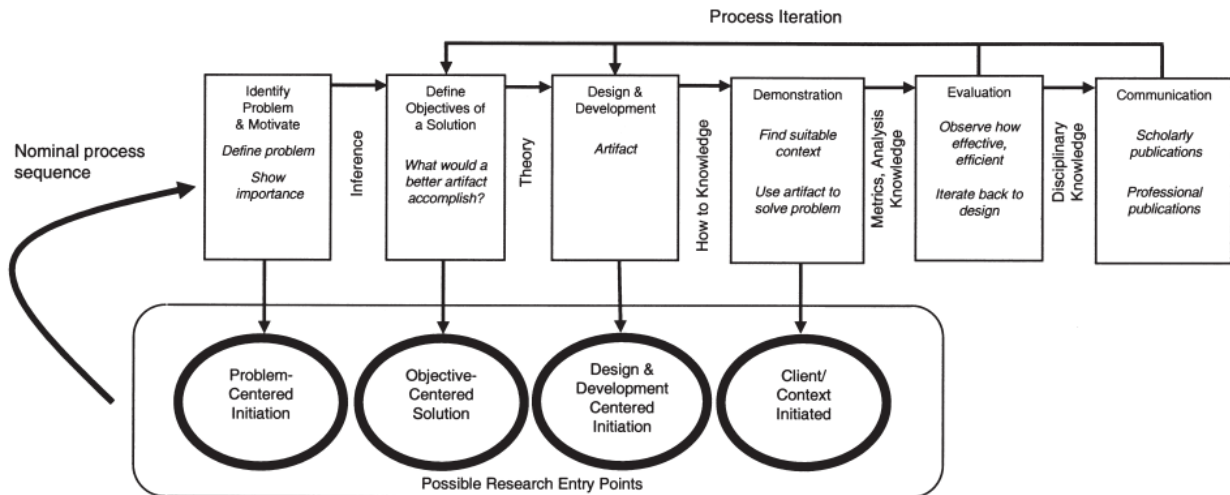


Figura 2- Design Science Research Methodology (DSRM) [18]

Este método é dividido por quatro diferentes fases que desenvolvem assim seis atividades também diferentes:

- **Abordagem centrada no problema:** Onde irá ser definido o problema específico, sendo necessário ao investigador recorrer a projetos relacionados com o estado do problema e interpretar algumas soluções previamente realizadas e assim tentar encontrar a melhor solução. Esta atividade é realizada na secção 2.2.
- **Solução centrada no objetivo:** Nesta fase, serão deduzidos quais os objetivos para a definição do problema através do conhecimento adquirido. Foram indicados nesta secção, 1.3, os objetivos pretendidos a alcançar na realização deste trabalho, de forma qualitativa, de modo a abordar novos protótipos para suportar o problema do trabalho.
- **Abordagem centrada no projeto e desenvolvimento:** Esta parte abrange o estabelecimento e desenvolvimento de mecanismos necessários para a realização do projeto. Nesta fase será indicada como as redes neuronais foram implementadas para realizar a estimação de canal para a rede 6G.
- **Solução iniciada pelo contexto:** Esta fase será a última parte do projeto, onde irão ser utilizados os mecanismos estabelecidos na fase anterior, e com estes demonstrar e avaliar a solução encontrada para o problema. Nesta fase, serão abordados os resultados obtidos e concluindo acerca do problema em causa

1.5 Planeamento

De forma a planear o tempo estabelecido para cada parte da dissertação, foi estabelecido um plano para orientar cada tarefa.

Precedentemente, foi **discutido com os orientadores acerca do tema da dissertação**. No decurso de toda a dissertação foram realizadas **reuniões bimensais com os orientadores**, para interpretar o trabalho realizado. A **escrita da dissertação** foi também efetuada durante toda a dissertação, sendo atualizada constantemente à medida que iam sendo efetuados progressos.

Durante a realização da dissertação, teve de ser **efetuado um relatório**, onde foram realizados os dois primeiros capítulos da dissertação, que envolvem a introdução e a revisão de literatura.

Após a realização desse relatório, avança-se para o **estudo das redes neuronais**, de modo a interpretar que metodologia existente será a melhor para a realização do trabalho. Será então **estudado o exemplo do Matlab já efetuado e reproduzindo o exemplo**, de forma a extrair dados relevantes para posteriormente **implementar o mecanismo** para resolver o problema exposto no tema da dissertação. Após essa implementação, será **testado o mecanismo** efetuado para a realização do problema.

Por fim, será efetuada a **escrita de um artigo**, onde serão alocados todos os mecanismos desenvolvidos e conclusões obtidas com esses mecanismos.

Tarefas	Início	Fim	Duração
Discussão do tema com os orientadores	Agosto de 2022	Setembro de 2022	1 mês
Reuniões com os orientadores	Outubro de 2022	Setembro de 2023	11 meses
Elaboração de relatório	Outubro de 2022	Fevereiro de 2023	4 meses
Estudo das redes neuronais	Fevereiro de 2023	Março de 2023	1 mês
Estudo e reprodução do exemplo do Matlab	Março de 2023	Abril de 2023	1 mês
Implementação em Matlab do mecanismo realizado	Abril de 2023	Julho de 2023	3 meses
Teste do modelo desenvolvido	Julho de 2023	Setembro de 2023	2 meses
Escrita de artigo	Agosto de 2023	Setembro de 2023	1 mês
Escrita da dissertação	Outubro de 2022	Setembro de 2023	11 meses

Tabela 2- Planeamento da realização da dissertação

2 Estado de Arte/ Revisão de Literatura

2.1 Conceitos

Para alcançar os objetivos para a dissertação definida é necessário identificar e interpretar certos conceitos importantes. Esses conceitos abrangem:

- **Modulação de canal:** uma vez que serão usadas redes neurais para a estimação de canal, é necessário um modelo que suporte essa utilização, sendo a modulação de canal um modelo ideal para o fazer. É neste processo que irão ser identificadas diferentes características do canal, tais como, perda de caminho (*path loss*), desvanecimento de caminhos múltiplos (*multipath fading*) e sombreamento (*shadowing*).
- **Machine/Deep Learning:** que é também um conceito chave no que diz respeito à utilização de redes neurais para a estimação de canal. Serão usadas técnicas de *Deep Learning* para treinar as redes neurais, e através disso usar o modelo treinado para estimar os coeficientes do canal.
- **Redes Neurais:** que são um tipo de modelo de *Machine Learning*, compostas por camadas de nós interconectados (neurónios) que permitem assim modelar a distribuição subjacente do canal.
- **Overfitting:** que é um problema muito habitual quando se treinam redes neurais, ocorrendo quando um modelo se adapta perfeitamente aos dados de treino, mas não consegue apresentar o mesmo desempenho para novos dados, e dessa forma é necessário utilizar técnicas de regularização, como *dropout* e *early stopping* para evitar esse *overfitting*, que permite limitar a capacidade do modelo para ajustar os dados de treino.
- **Verificação dos dados para treino e dados para validação:** é necessário dividir os dados em 2 diferentes conjuntos, sendo um usado para treinar a rede neuronal e outro conjunto de validação, para avaliar o desempenho do modelo treinado.
- **Retropropagação/Gradiente Descendente:** que é o método utilizado para ajustar as características da rede neuronal com base nos coeficientes de canal previstos e os reais, através do algoritmo de otimização Gradiente Descendente. Este método possibilita dessa forma minimizar o erro e melhorar a precisão da estimação de canal.

Este trabalho assenta na estimação de canal das futuras redes 6G. Este processo determina as características do canal (atenuação, distorção, largura de banda, etc) que são definidas através de amostras de sinais transmitidos. Desta forma permite que estes sinais possam ser corrigidos de forma a melhorar a qualidade dos sistemas de comunicações. Existem vários métodos para fazer estimação de canal, tais como métodos baseados em:

- **Um sinal piloto**: Um sinal que é conhecido (sinal piloto) vai ser transmitido pelo emissor e recebido pelo recetor, que irá utilizar a informação oriunda desse sinal para estimar as características do canal.
- **Múltiplas amostras de sinais**: Neste método, o emissor irá enviar múltiplos sinais diferentes, e o recetor irá receber esses sinais, e usufruindo da informação destes, irá estimar as características do canal.
- **Machine/Deep Learning**: O recetor usa algoritmos de *Deep Learning* para efetuar a estimação de canal através das amostras de sinais recebidos.
- **Mínimos Quadrados (LS)**: Neste caso, o recetor usa um algoritmo de mínimos quadrados para estimar as propriedades do canal através dos sinais recebidos. É um método matemático que tem como objetivo minimizar a soma dos quadrados das diferenças entre os valores observados e os valores estimados pelo modelo.
- **Mínimo Erro Quadrático Médio (MMSE)**: Neste método, o recetor usa o algoritmo do mínimo erro quadrático médio para estimar as características do canal através dos sinais recebidos. Este método dedica-se a minimizar a média dos erros quadráticos entre o valor estimado e o valor real, sendo um método de estimação linear, onde a estimação desse valor irá ser dada pela combinação linear das amostras.

2.2 Projetos Relacionados

Um trabalho realizado em [5] analisou a estimação de canal em sistemas OFDM com o uso de *Deep Learning*. Neste trabalho foram consideradas várias experiências para demonstrar o desempenho do método de *Deep Learning* em comparação com métodos tradicionais usados para efetuar a estimação de canal. Dessa forma é utilizada uma rede neuronal DNN, e são feitas comparações com esses métodos tradicionais em termos de taxa de erros de bits (BERs) sob diferentes relações sinal-ruído (SNRs).

Primeiramente fez-se uma experiência onde são utilizados blocos piloto para estimar o canal, sendo utilizados 64 e 8 pilotos respectivamente. Foi então comparada a DNN com dois modelos tradicionais, um baseado em mínimos quadrados (LS) e outro baseado no mínimo erro quadrático médio (MMSE). Quando utilizados 64 pilotos, verificou-se que o método LS foi o que apresentou pior desempenho (curva BER a tracejado vermelho na figura 3), sendo o método baseado em MMSE o que obteve um melhor desempenho, porém a abordagem baseada em DL tem um desempenho comparável com o método MMSE, tendo até para os valores mais baixos de SNR (de 5 a 15 dB), uma curva BER coincidente com a curva do método MMSE. Todavia quando são reduzidos o número de blocos piloto para 8, verifica-se que os dois métodos tradicionais, MMSE e LS, apresentam curvas BER que saturam para uma relação sinal-ruído mais elevada, por volta dos 10 dB, enquanto o método assente em DL possui a capacidade de reduzir o BER quando a SNR aumenta, e como é possível verificar em todas as curvas BER de todos os métodos na Figura 3, apenas a curva da DNN mantém o mesmo comportamento ao visto quando foram utilizados os 64 blocos piloto, o que constata que a DNN apresenta uma maior robustez para um menor número de pilotos usados para a estimação de canal.

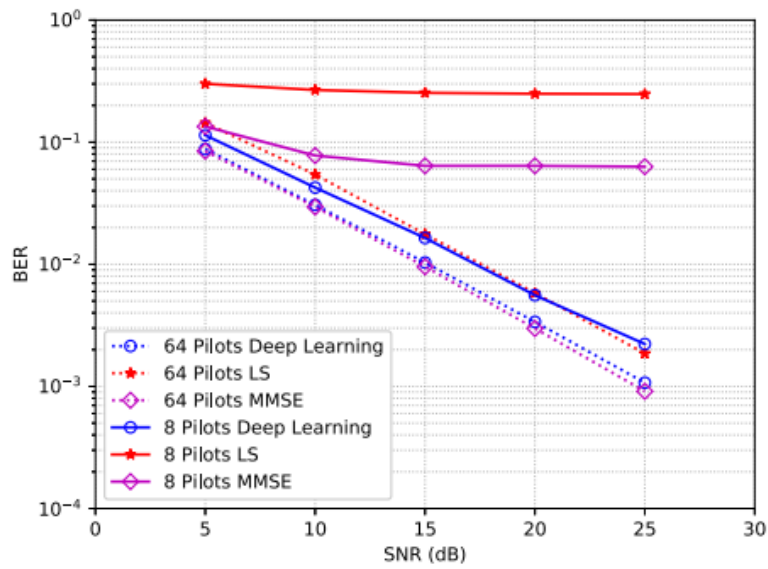


Figura 3-Utilização de blocos pilotos para comparar o desempenho entre o método baseado em DL e os métodos LS e MMSE [5]

A segunda simulação é feita com base na remoção do prefixo cíclico (CP), que tem como função mitigar a interferência intersimbólica (ISI). Ao remover-se o CP, denota-se que nenhum dos métodos tradicionais usados para a estimação de canal (MMSE e LS) podem efetivamente fazer a estimação de canal, uma vez que a precisão (curva BER) satura para SNRs superiores a 15 dB. Em contraste, o método baseado em DL demonstra um bom desempenho ao longo do aumento da relação sinal-ruído. Este melhor desempenho pode ser explicado por as características dos canais sem fio poderem serem aprendidas com base nos dados de treino gerados a partir do modelo da DNN.

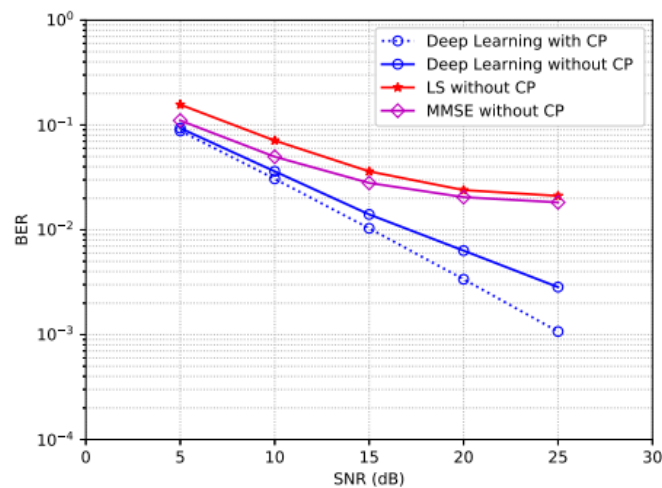


Figura 4- Remoção do prefixo cíclico (CP) para comparar o método segundo a DNN e os métodos tradicionais [5]

Foi então efetuado um treino, onde é inserido ruído de corte para averiguar o comportamento do método baseado em MMSE e do método baseado em *Deep Learning*. Analisando os resultados obtidos, foi possível apurar que a DNN revela um melhor desempenho que o método MMSE, quando a relação sinal-ruído atinge valores mais elevados, o que indica que o método de DL é mais robusto ao ruído de corte.

Por fim, foram testadas todas as adversidades em conjunto (Figura 3), ou seja, apenas 8 pilotos são usados, o CP é removido e há ruído de corte, em que se verificou que a DNN evidencia um desempenho bem superior ao método tradicional MMSE. Dessa forma, constata-se que o Deep Learning tem vantagens quando são adicionadas distorções e interferências, o que indicia que as DNNs possuem a capacidade de lembrar e analisar as características dos canais sem fio.

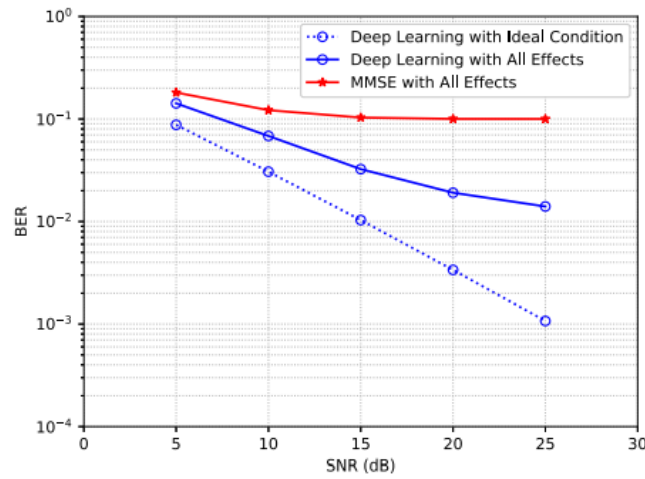


Figura 5- Comparação da curva BER entre a DNN e o MMSE com todas as adversidades presentes [5]

Um trabalho realizado por [6], foram interpretadas as mesmas adversidades, à exceção da inserção do ruído de corte, mas ao invés do modelo baseado em DL ser utilizada uma DNN, optaram por utilizar o modelo LSTM. Este modelo é assente numa arquitetura de uma rede neuronal recorrente (RNN). O LSTM é composto por 3 camadas: camada de entrada, camada oculta e camada de saída. A camada de entrada é onde irão ser tratados os dados de entrada, ao cargo que as células LSTM estão encarregues da camada oculta. Os resultados serão então exibidos na camada de saída. Como indicado, neste artigo foram abordados igualmente os 2 métodos tradicionais MMSE e LS em comparação com o método baseado em DL, o LSTM. Tal como verificado em [5], o modelo de *Deep Learning* provou ser mais robusto que os outros métodos utilizados como pode ser verificado na figura 4, seja para uma simulação com um número de pilotos menor ou para o uso de um prefixo cíclico (CP) menor. Desta forma, verifica-se que o DL pode ser um método promissor para estimação de canal em sistemas celulares e provavelmente na futura tecnologia 6G.

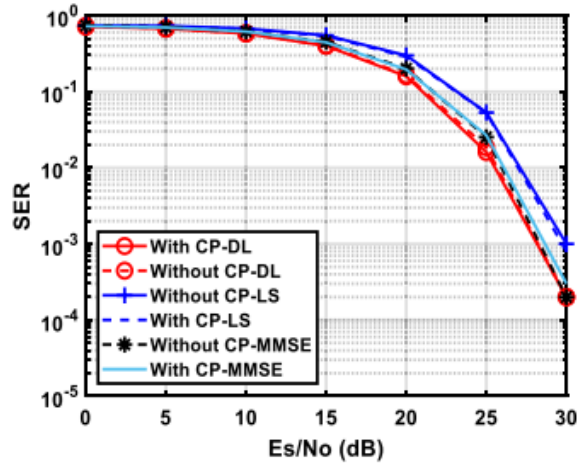


Figura 6- Diferença entre o modelo LSTM e os métodos convencionais [6]

Em [7], foram efetuadas diferentes formas de estimação de canal baseadas em redes neurais convolucionais (CNN). Implementou-se primeiro uma estimação de canal baseada em uma CNN de frequência espacial (SF-CNN), explorando assim a correlação espacial e de frequência. Posteriormente, foi analisada a correlação temporal em canais variantes no tempo, uma CNN de frequência espacial-temporal (SFT-CNN), de forma a melhorar a precisão. Por fim, foi projetada uma CNN reduzida por piloto espacial (SPR-CNN), com finalidade de controlar a sobrecarga do número piloto espacial para a estimação de canal. Estas 3 abordagens serão comparadas com o método MMSE não ideal e o estimador MMSE ideal. Foi efetuada a experiência para a estimação de canal baseada em SFT-CNN, juntamente com SF-CNN e o método MMSE. Tanto para a abordagem MMSE como para a SFT-CNN foi utilizada a informação do canal do intervalo de coerência anterior para efetuar a estimação de canal, enquanto para a SF-CNN não foi utilizada correlação temporal. Com a introdução da correlação temporal, denota-se que há uma melhoria do desempenho na estimação de canal baseada em SFT-CNN, superando o SF-CNN e o MMSE não ideal de forma significativa. O método SFT-CNN aproxima-se do desempenho do MMSE ideal, principalmente para SNRs baixos e médios. De seguida, foi introduzido um cenário de rua sem linha de vista (NLOS) macro urbano (UMa), para verificar o comportamento dos métodos SF-CNN e SFT-CNN em relação aos métodos MMSE, e concluiu-se que ambos superaram os métodos tradicionais MMSE, havendo uma degradação no desempenho desses métodos, enquanto os métodos baseados em CNNs exibiram uma robustez superior para esse cenário diferente.

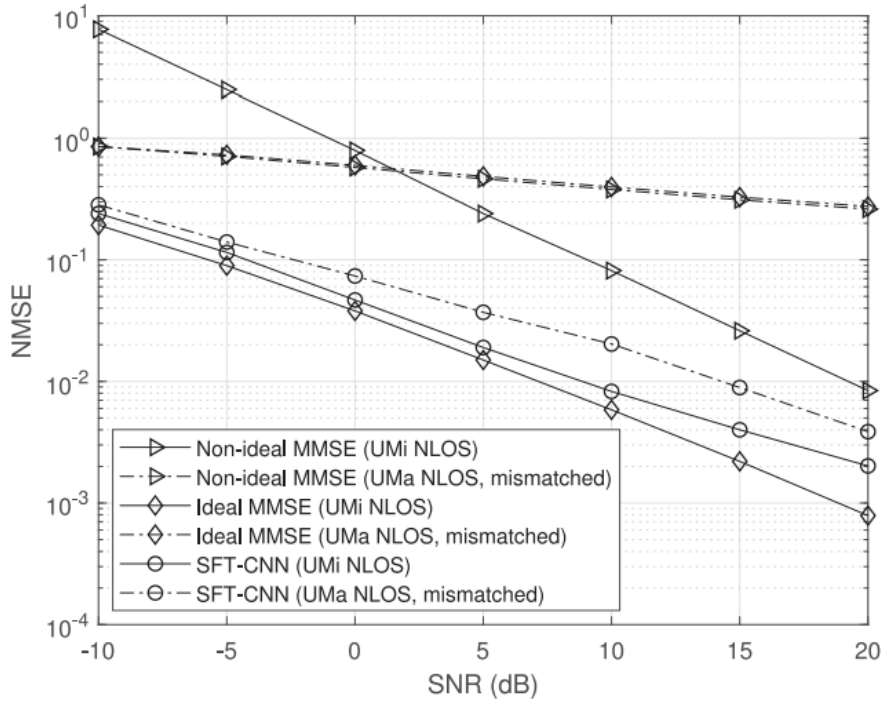


Figura 7- Analogia entre SF-CNN e SFT-CNN com os métodos MMSE ideal e não-ideal [7]

Por último, foi então adicionada a estimação de canal baseada em SPR-CNN, onde os canais serão agrupados e estimados por uma unidade de estimação de canal (CEU) com memória em 4 intervalos sucessivos de coerência, impondo apenas uma sobrecarga no piloto espacial de apenas 1/3. As abordagens baseadas no MMSE ideal e não ideal usam também as mesmas configurações que o SPR-CNN. No SF-CNN e SFT-CNN, as abordagens foram feitas utilizando toda a sobrecarga no piloto espacial. Observando os resultados da simulação, constata-se que a estimação de canal baseada em SPR-CNN atingiu um desempenho semelhante às abordagens baseadas em SF-CNN e SFT-CNN, especialmente nos valores de SNR baixos e médios, tendo em conta que apenas necessita de cerca de um terço da sobrecarga do piloto espacial ao custo da complexidade. Pelo contrário ambos os métodos MMSE demonstram um mau desempenho usando a mesma informação de correlação de canal que o SPR-CNN, o que comprova que o SPR-CNN resiste à redução da sobrecarga do piloto espacial, significando que este método utilizou com eficiência a frequência adicional e a correlação temporal. Em suma, conclui-se que as abordagens baseadas na CNN superam o estimador MMSE não ideal, e atingem um desempenho muito próximo do MMSE ideal, mesmo requerendo uma menor complexidade.

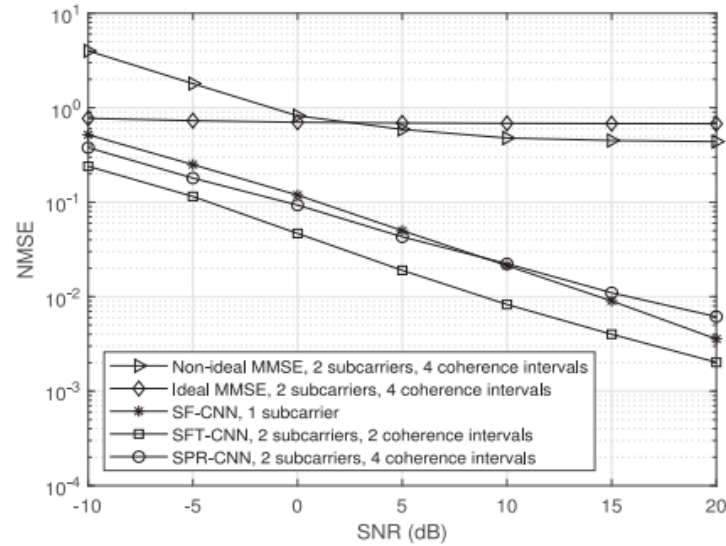


Figura 8- Curvas NMSE entre os vários tipos de CNN e dos métodos MMSE [7]

3 Modelo do Sistema

O estudo da estimação de canais em futuras redes 6G irá basear-se num modelo Matlab relacionado com a criação de dados e o treino de uma rede neuronal convolucional (CNN) para a estimação de canais em sistemas 5G. O modelo do sistema consiste em várias etapas, incluindo a criação de dados, o treino da rede neuronal e a avaliação dos resultados.

Em primeiro lugar, é configurado o ambiente onde são definidos parâmetros como o tamanho dos dados de treino, a taxa de amostragem, o tamanho da grelha, o número de antenas de transmissão/receção e a configuração PDSCH (Pilot Specific Downlink Symbols).

Posteriormente é utilizada uma função que é responsável pela criação de dados de treino para a estimação de canal onde é efetuada uma simulação para cada exemplo de treino: É criado um modelo de canal TDL (Tap Delay Line) com diferentes definições, como o perfil de atraso e a propagação do atraso. É gerado um sinal OFDM (Orthogonal Frequency Division Multiplexing) com símbolos DM-RS (Downlink Specific Pilot Symbols) inseridos. O sinal é transmitido através do canal TDL, é adicionado AWGN (Additive White Gaussian Noise) e a receção é simulada. A estimação do canal é efetuada por interpolação linear dos símbolos DM-RS recebidos. O sinal recebido e a estimação de canal são armazenados como exemplos de treino.

O OFDM é uma técnica de modulação avançada que utiliza símbolos piloto para corrigir a distorção do canal, garantindo uma transmissão de dados precisa, através de uma subportadora flexível e de atribuição de potência. É uma técnica promissora muito devido à grande capacidade da transmissão de dados, com elevada eficiência de largura de banda e à sua robustez face ao

atraso de multipercursos [21]. Ao intercalar estes sinais de referência com os dados transmitidos, o sistema OFDM supera desafios como o desvanecimento e a interferência, garantindo uma comunicação fiável mesmo em condições adversas. Desta forma, o sistema utilizado nesta dissertação utiliza símbolos piloto OFDM para obter uma transmissão de dados precisa e robusta. A abordagem à estimação de canal irá consistir em utilizar os símbolos piloto de referência conhecidos na transmissão, e em seguida interpolar o resto da resposta do canal através do uso desses símbolos piloto (Figura 9).

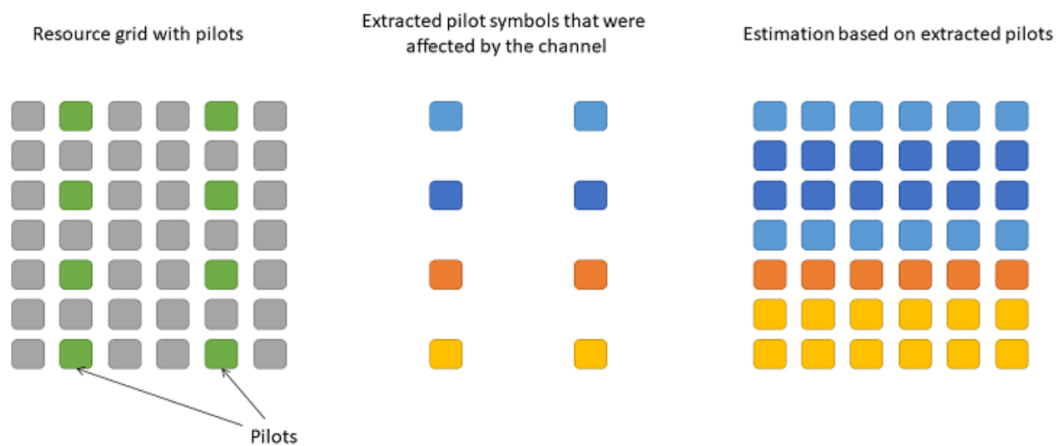


Figura 9- Utilização de símbolos piloto para efetuar a estimação de canal [22]

Depois de os dados serem gerados, é definida a arquitetura da rede neuronal, onde a arquitetura da CNN é definida na variável de camada. A CNN foi concebida para processar as imagens (partes reais e imaginárias) da grelha do canal como entrada e prever a estimação do canal como saída.

Finalmente, é definido o treino da rede neuronal, sendo especificado todos os dados necessários para esse ser efetuado. Os dados de treino e de validação são separados. As opções de treino, o otimizador e a taxa de aprendizagem são definidos. A CNN é treinada utilizando os dados de treino e de validação.

```
layers = [ ...  
    imageInputLayer([612 14 1], 'Normalization', 'none')  
    convolution2dLayer(9,64, 'Padding',4)  
    reluLayer  
    convolution2dLayer(5,64, 'Padding',2, 'NumChannels',64)  
    reluLayer  
    convolution2dLayer(5,64, 'Padding',2, 'NumChannels',64)  
    reluLayer  
    convolution2dLayer(5,32, 'Padding',2, 'NumChannels',64)  
    reluLayer  
    convolution2dLayer(5,1, 'Padding',2, 'NumChannels',32)  
    resize3dLayer('OutputSize',[612 14 1]);  
    regressionLayer  
];
```

Figura 10- Composição da rede base

Ao longo do desenvolvimento do estudo das redes neurais vai se partir de “benchmark” uma rede neuronal definida anteriormente pelo Matlab, constituída por 5 camadas convolucionais (convolution2dLayer na Figura 10), em que os componentes são expressos pelo número e tamanho dos filtros utilizados, pelo número de canais de saída nessa camada e pelos parâmetros que influenciam a rede, como o Padding e o Stride. Cada camada convolucional é seguida pela função de ativação ReLU, que é utilizada para introduzir a não linearidade na rede neuronal.

```

layers = [ ...
    imageInputLayer([612 14 1], 'Normalization', 'none')
    convolution2dLayer(9, 64, 'Padding', 'same')
    reluLayer
    convolution2dLayer(5, 64, 'Padding', 'same', 'NumChannels', 64)
    reluLayer
    convolution2dLayer(1, 64, 'Padding', 'same')
    reluLayer
    groupedConvolution2dLayer(5, 1, 'channel-wise', 'Padding', 'same')
    reluLayer
    convolution2dLayer(1, 64, 'Padding', 'same')
    reluLayer
    convolution2dLayer(5, 32, 'Padding', 'same', 'NumChannels', 64)
    reluLayer
    convolution2dLayer(5, 1, 'Padding', 'same', 'NumChannels', 32)
    resize3dLayer('OutputSize', [612 14 1]);
    regressionLayer
];

```

Figura 11- Composição de um exemplo de rede com camadas Depthwise

Um dos objetivos desta dissertação é utilizar a rede pré-definida pelo modelo como base, tentando substituir tanto quanto possível as camadas convolucionais, por camadas Depthwise, de forma a tentar reduzir a complexidade da rede e manter um bom desempenho da mesma. Estas camadas são assim mais eficientes computacionalmente, sendo as camadas ideais para dispositivos com recursos computacionais mais limitados, como dispositivos móveis. Na Figura 11 verifica-se então que cada camada Depthwise é composta por duas camadas convolucionais de filtros 1x1, reagrupadas juntamente com uma camada denominada de groupedConvolution2dLayer, formando assim a camada Depthwise. Tanto a rede base como as outras redes criadas são configuradas com uma camada resize3dLayer que serve para redimensionar a saída da camada anterior para ter o tamanho da camada de entrada, neste caso 612x14x1. Esta camada é apenas útil para quando é utilizado o parâmetro Stride. A rede termina

então com a camada `regressionLayer` usada para realizar a regressão da rede. Um exemplo da composição de uma rede neuronal utilizada é mostrado na Figura 12.



Figura 12- Exemplo da composição de uma rede neuronal

3.1 Modelo de Treino

O processo de teste de uma rede neuronal envolve 3 processos diferentes: treino da rede, validação da rede e, finalmente, teste da rede. O treino da rede é o mais importante desses processos. Durante o treino da rede, o número de Epochs usados para treinar a rede pode ser especificado e, quanto mais Epochs forem usados, mais iterações serão usadas durante o treino da rede. Depois de definir os parâmetros da rede para iniciar o treino, abre-se uma janela (Figura 13) onde se pode verificar a variação da raiz do erro quadrático médio (RMSE) e as perdas totais ao longo dos Epochs. A partir destes valores de RMSE, são calculados os valores de MSE, que são utilizados para analisar o desempenho da rede relativamente à relação sinal/ruído.

Se não se pretender treinar a rede para analisar os resultados, existe um modelo pré-treinado com 16.384 exemplos de treino. A criação de novos dados é pré-definida para produzir 256 exemplos de treino. Essa quantidade de dados é suficiente para treinar uma rede de estimação de canal funcional em uma CPU em um período razoável.

No entanto, é possível aumentar o número de exemplos de treino e, quanto maior o número, mais bem treinada será a rede e melhor será seu desempenho.

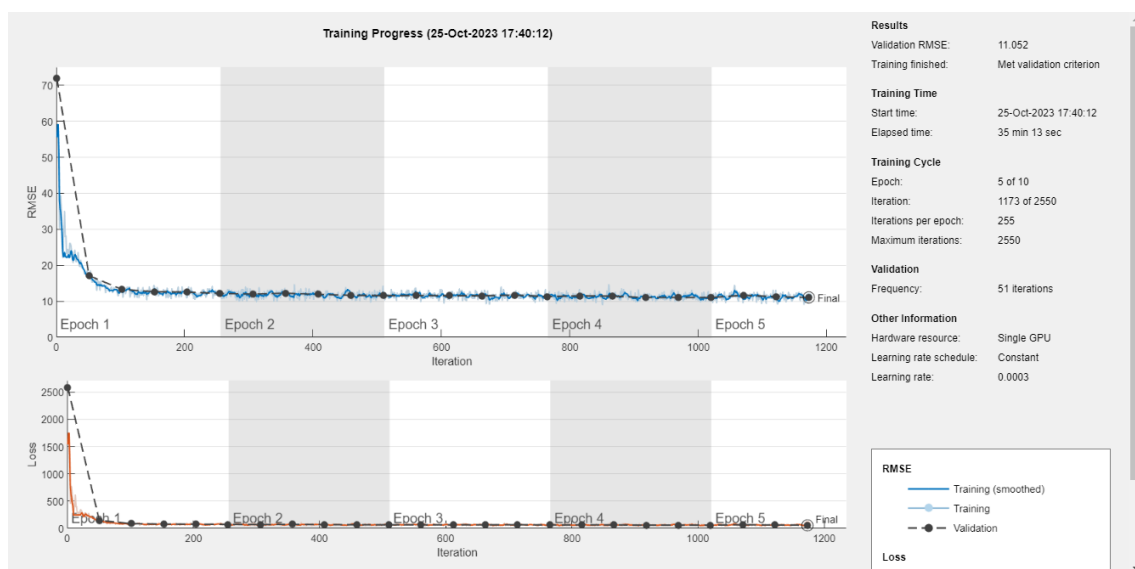


Figura 13- Gráficos resultantes de treino efetuado na rede

3.2 Complexidade e Desempenho

Uma forma de avaliar as redes neurais utilizadas tem a ver com a complexidade da rede, exigindo um equilíbrio entre a complexidade e o nível de desempenho. Se uma rede for mais complexa, espera-se que tenha melhor desempenho, mas se uma rede com menor complexidade não for tão eficiente quanto uma rede mais complexa, a melhor rede deve ser avaliada considerando ambas as variáveis, desempenho e complexidade.

Para calcular a complexidade da rede, são utilizadas apenas as operações de multiplicação de cada camada da rede neuronal. A complexidade é então calculada multiplicando o tamanho do filtro na vertical e na horizontal, neste caso o mesmo, pelo número de filtros utilizados. Este valor é depois multiplicado pelo tamanho da saída da camada anterior, e assim sucessivamente entre camadas. A comparação da complexidade entre redes neurais é então efetuada através da diferença percentual entre as duas.

Para demonstrar um exemplo de como foi calculada a complexidade de cada rede neuronal, vamos demonstrar o exemplo da rede base (Figura 10). A rede base é primeiro definida com uma camada de entrada com tamanho $612 \times 14 \times 1$, formada por um único canal. De seguida, a rede passa por uma camada onde é realizada convolução, `convolution2dLayer`, sobre os parâmetros de entrada. A convolução passa por aplicar filtros de forma a extrair características da rede. Neste caso são aplicados 64 filtros de tamanho 9×9 , sendo que cada filtro se associa a um canal de saída. A complexidade é então calculada fazendo a multiplicação do tamanho do filtro na vertical e horizontal, neste caso é o mesmo, com a multiplicação do número de filtros usados. A esse valor multiplica-se então o tamanho de saída da camada anterior, ou seja, na primeira camada multiplica-se por $612 \times 14 \times 1$ que é o tamanho da camada de entrada. Assim sendo, após passar a primeira camada, a rede tem uma complexidade de $9 \times 9 \times 64 \times 612 \times 14 \times 1 = 44,416,512$. Na segunda camada da rede é aplicada 64 filtros de tamanho 5, e dessa forma a complexidade da rede passa a ser $5 \times 5 \times 64 \times 612 \times 14 \times 64 = 877,363,200$. A terceira camada da rede apresenta os mesmos parâmetros que a segunda, consequentemente tem a mesma complexidade que essa camada. Na quarta camada da rede varia apenas o número de filtros usados, passando de 64 para 32 filtros, o que resulta numa complexidade de $5 \times 5 \times 32 \times 612 \times 14 \times 64 = 438,681,200$. Por último a rede é composta por uma camada convolucional de 32 filtros de tamanho 1, com uma complexidade igual a $1 \times 1 \times 32 \times 612 \times 14 \times 32 = 8,773,632$. Para obter a complexidade total da rede basta somar a complexidade de todas as camadas que constituem a rede, obtendo assim uma complexidade igual a 2,244,678,912.

3.3 Stride

Inicialmente, a rede base foi testada, mas, em vez de se impor o parâmetro Padding à rede, foi adicionado Stride à rede para verificar se ela poderia manter um bom desempenho. Esse parâmetro refere-se ao tamanho do passo que o filtro usa para percorrer a entrada de cada convolução, ou seja, o Stride determina em quantas posições o filtro é usado, horizontal ou verticalmente. O Stride é, por conseguinte, introduzido em cada camada da rede neuronal para reduzir ainda mais a complexidade e analisar a eficiência da rede. Assim, foi introduzido um Stride de 2 em cada camada da rede neuronal. Quando o passo é introduzido na rede neuronal, o desempenho da rede diminui significativamente e é pior do que o estimador prático para todos os valores de SNR analisados, como se pode confirmar nos gráficos indicados na secção Anexos.

3.4 Padding

Depois de verificar que o Stride degradou significativamente o desempenho da rede em comparação com o estimador prático, simplesmente introduzimos o Padding, um parâmetro usado para ajustar o tamanho das entradas, com o parâmetro Padding definido como "same", para que a operação de convolução tenha o mesmo tamanho que a entrada, para manter dimensões consistentes em todas as camadas da rede. Este parâmetro permite adicionar camadas ao redor da entradas. Dessa forma, permite controlar assim o tamanho de saída das operações de convoluções, e salvaguarda as informações correspondentes às camadas de entrada.

Dentro deste parâmetro, pode ser definido o valor que se quer atribuir, desde ao tamanho introduzido em nível de número de bordas que irão ser aplicadas, neste caso como dito anteriormente foi definido como "same", ou pode ser colocado um valor numérico que irá corresponder ao número de bordas adicionado pelo Padding. Neste caso ao ser utilizado o Padding como "same", todas as camadas adicionadas têm valor igual a 0, fazendo com que dessa forma que a saída da camada de convolução tenha as mesmas dimensões que a entrada original. Ao invés disso podem ser usadas camadas com outros valores diferentes de 0, variando a variável PaddingValue no Matlab, porém esta iria alterar as características das redes neuronais criadas, além de prejudicar o desempenho da rede, sendo que por isso optou-se pela opção do Same Padding.

Em seguida, foram testadas duas redes neuronais, além da rede base, em que algumas das camadas convolucionais da rede de base são substituídas por camadas Depthwise. A rede A

substitui 2 camadas convolucionais por um filtro de tamanho 5 com 2 camadas Depthwise. Na rede B, foi acrescentada uma camada Depthwise adicional à rede A. Desta forma, o desempenho da rede neuronal melhora à medida que a complexidade da rede é reduzida, e quanto mais camadas Depthwise a rede neuronal tiver, menor será a sua complexidade. Os resultados obtidos estão expressos na próxima secção, a secção 6 correspondente aos Resultados.

4 Resultados das Simulações

Várias experiências foram realizadas para demonstrar o desempenho das redes neurais utilizadas e do estimador prático. As simulações são avaliadas por meio de um gráfico que quantifica o MSE em função da relação sinal-ruído, numa faixa entre 0 e 20 dB. Os testes realizados identificaram quais as redes que poderiam ser utilizadas como alternativas à rede base sem comprometer o desempenho. Dessa forma nesta secção apenas serão apresentadas as redes neurais criadas que apresentaram um desempenho semelhante ou superior à rede base considerada. Todas as outras redes criadas que não foram tão eficientes quanto era esperado, terão os seus resultados demonstrados na secção Anexos.

Foram realizados dois testes idênticos, mas com diferentes quantidades de treino. Primeiro, todas as redes foram testadas com apenas 256 sessões de treino e, em seguida, o número de sessões de treino foi aumentado para 4096.

Foi então desenvolvida primeira uma rede neuronal formada por 4 camadas convolucionais e substituiu-se 1 camada convolucional correspondente à rede base por 2 camadas Depthwise, de forma a reduzir a complexidade da rede, e de seguida verificou-se se essa rede conseguiu ser tão eficiente quanto a rede base.

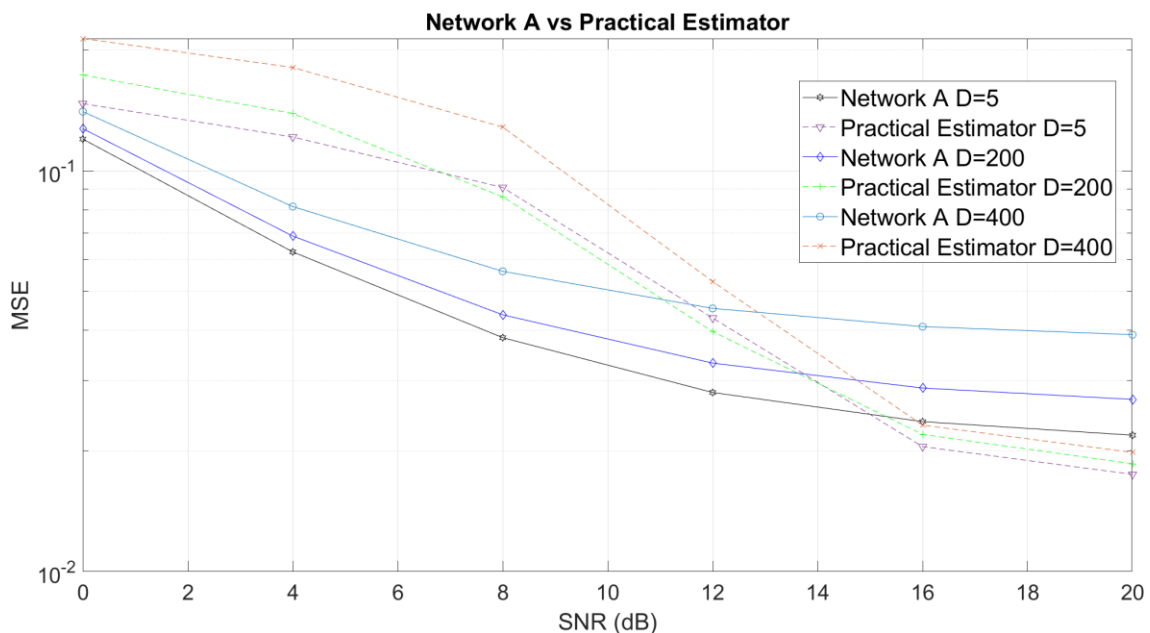


Figura 14- MSE do estimador prático vs MSE da rede A, ambos com 256 sessões de treino

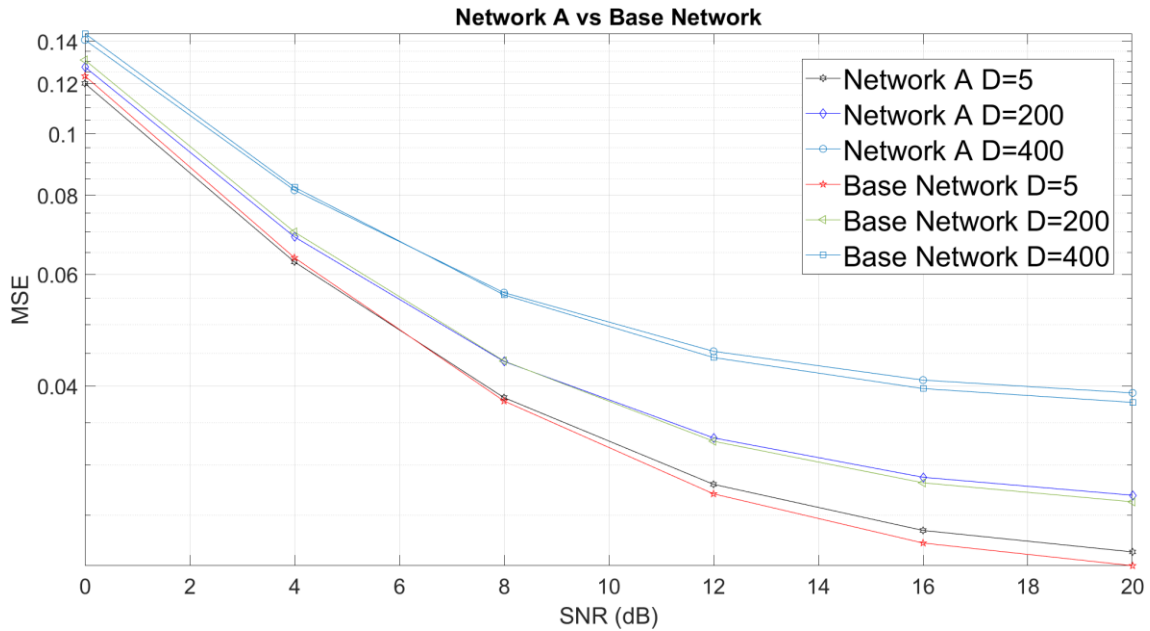


Figura 15- MSE da rede base vs MSE da rede A, ambas com 256 sessões de treino

A Figura 14 mostra que, como esperado, a rede A supera significativamente o estimador prático em valores mais baixos de SNR (com mais ruído), tal como a rede base. Analisando os resultados obtidos em comparação com a rede dada no exemplo (Figura 15), a rede A, com uma complexidade inferior à da rede de base, tem um desempenho tão bom como o da rede de base, e até melhor, para valores mais baixos de SNR, sendo apenas superada para valores com menor ruído. Porém para todas as velocidades de Doppler, verificou-se que a rede base acaba por superar ligeiramente em todos os casos o desempenho da rede A. De um contexto geral, as Figuras 14 e 15 demonstraram que o desempenho das redes é sempre superior para velocidades de Doppler mais baixas.

Avaliando as conclusões da experiência anterior, foi adicionada uma nova camada Depthwise e removida outra camada convolucional, com o intuito de reduzir ainda mais a complexidade, deixando 3 camadas Depthwise na rede, formando assim a Rede B, para analisar se a rede ainda podia melhorar em comparação com a rede de base e com a Rede A.

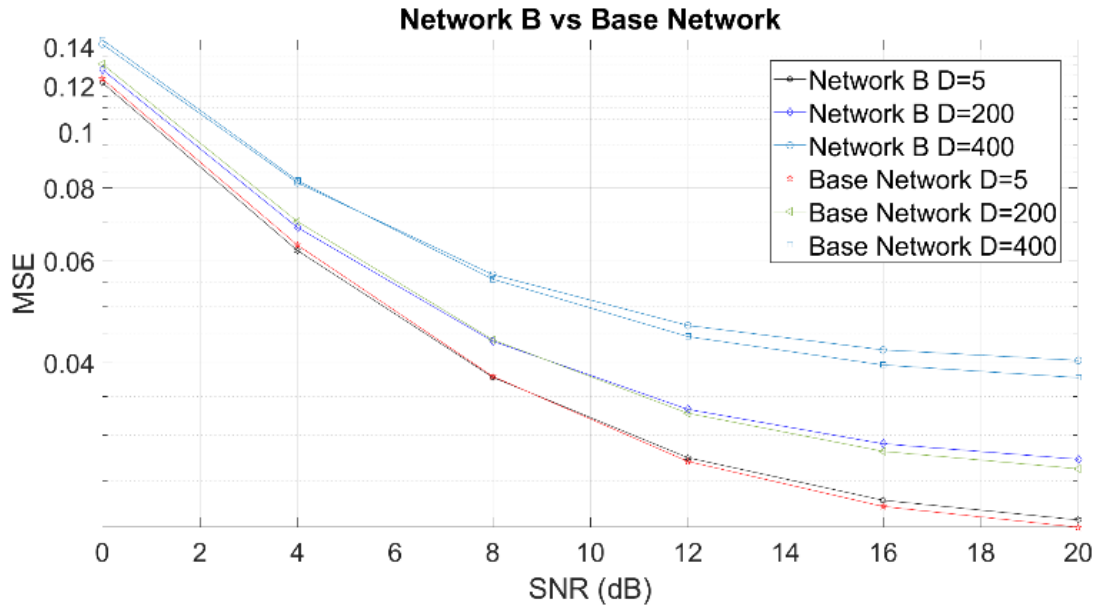


Figura 16- MSE da rede base vs MSE da rede B, ambas com 256 sessões de treino

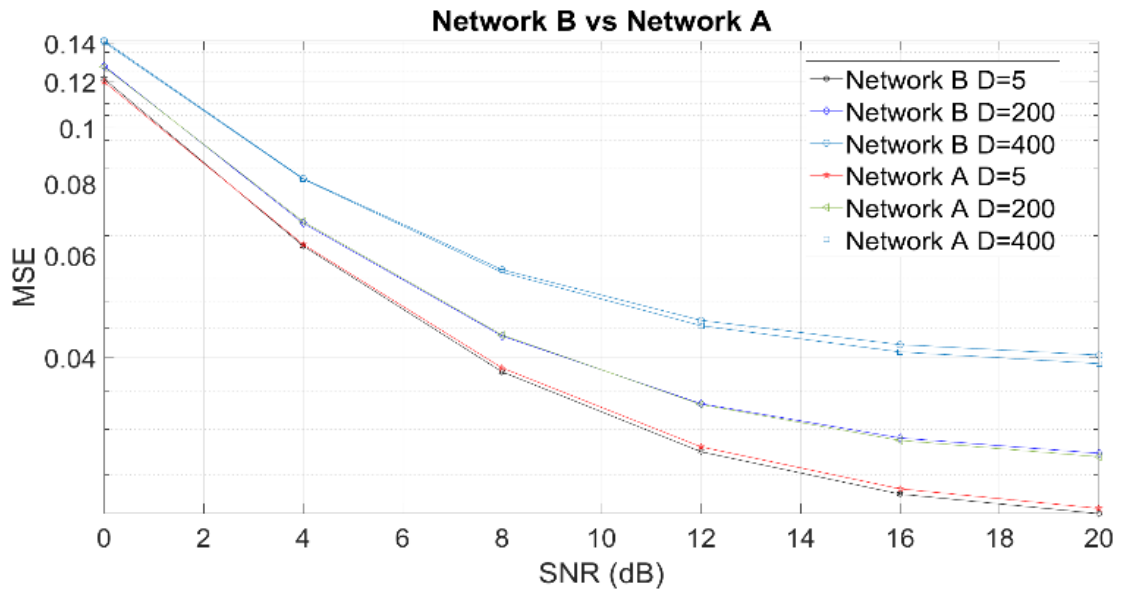


Figura 17- MSE da rede B vs MSE da rede A, ambas com 256 sessões de treino

Como pode ser visto nas Figuras 16 e 17, a Rede B acrescentou pouco em termos de desempenho da rede em comparação com a rede base e a Rede A, tanto que nenhuma velocidade de Doppler testada, a rede B superou a rede base, e em comparação com a rede A (Figura 17) apenas na velocidade de Doppler mais baixa ($D=5$) a rede B supera essa rede. Posto isto entendeu-se que não valia a pena continuar a tentar reduzir a complexidade das redes, pois o desempenho destas acabaria por diminuir em comparação com as outras redes mais complexas.

Depois de experimentar com 256 sessões de treino, o número de treinos foi aumentado para 4096 para ver se as redes com camadas Depthwise com mais treino superavam a rede base.

Base Network with 4096 training seasons vs Network A with 4096 training seasons

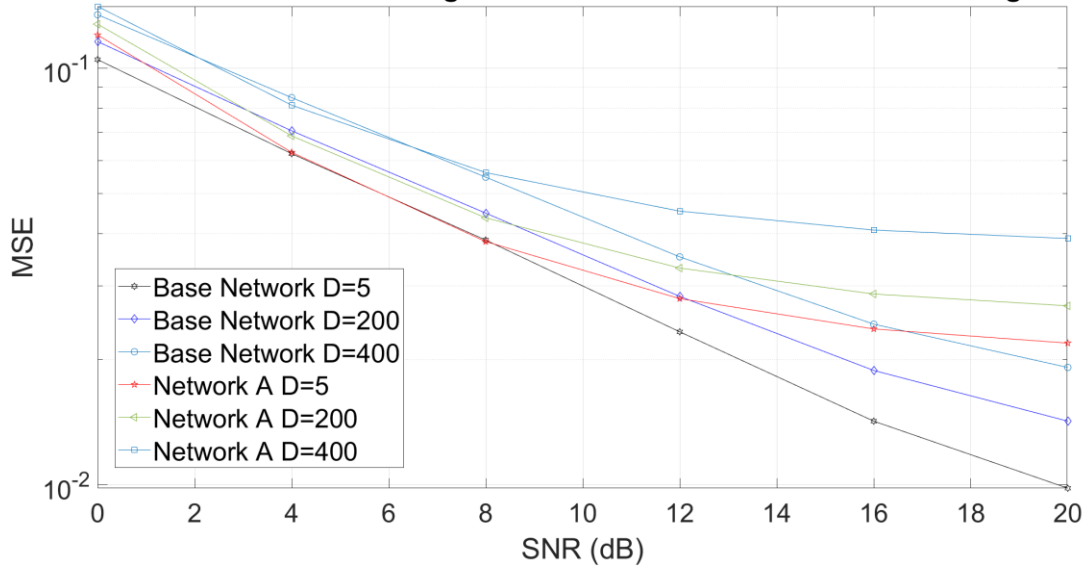


Figura 18- MSE da rede base vs MSE da rede A, ambas com 4096 sessões de treino

Base Network with 4096 training seasons vs Network B with 4096 training seasons

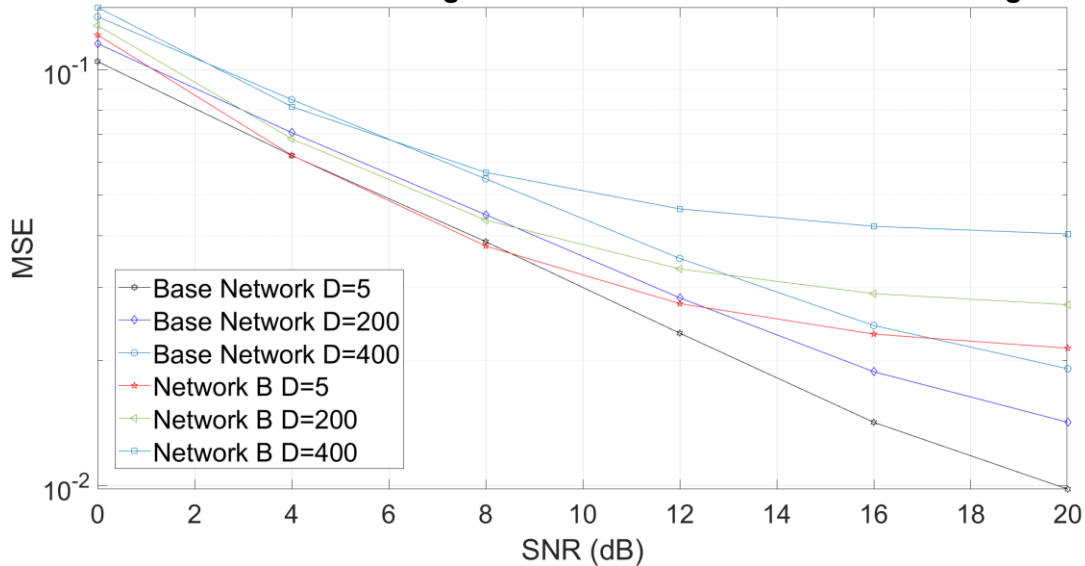


Figura 19- MSE da rede base vs MSE da rede B, ambas com 4096 sessões de treino

As figuras acima mostram comparações entre as redes A e B e a rede de base, todas com 4096 treinos. Os resultados mostram que a rede base com mais treino também melhorou seu desempenho no final, com melhores valores de MSE do que as outras redes com camadas Depthwise. Devido à maior complexidade da rede, foi possível concluir que a rede base acaba por ter um melhor desempenho que as outras redes (A e B) à medida que o número de treinos aumenta,

como se pode verificar no exemplo com 256 treinos, onde a rede B tem um comportamento muito semelhante à rede base, e neste caso com 4096 treinos acaba por ficar um pouco mais longe.

Desta forma, verificou-se que ao retirar a complexidade da rede pré-definida, as redes acabaram por reduzir o seu desempenho relativamente à rede pré-definida. No entanto, para mostrar que é possível melhorar a rede de forma que o seu desempenho melhore em relação à rede base, foram criadas a rede C, que possui as mesmas 5 camadas convolucionais da rede base, com mais 2 camadas de profundidade, e a rede D, que possui 6 camadas convolucionais. Estas redes acabam por ter uma complexidade superior à da rede de base e, por conseguinte, melhoram o desempenho em relação à rede de base (Figuras 20 e 21).

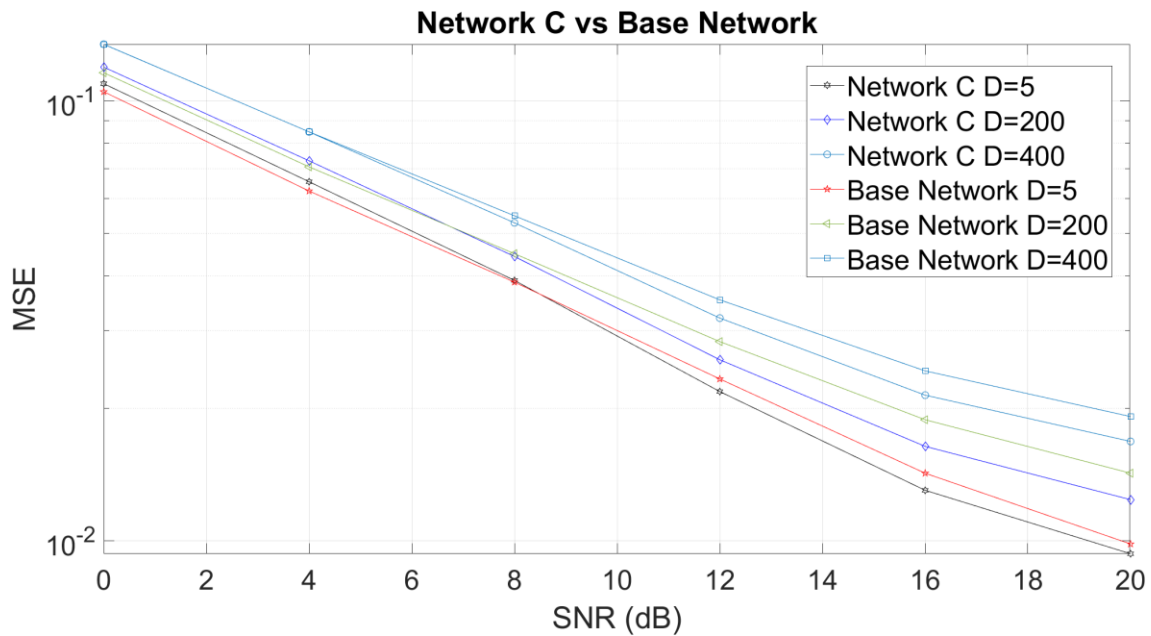


Figura 18- MSE da rede base vs MSE da rede C, ambas com 4096 sessões de treino

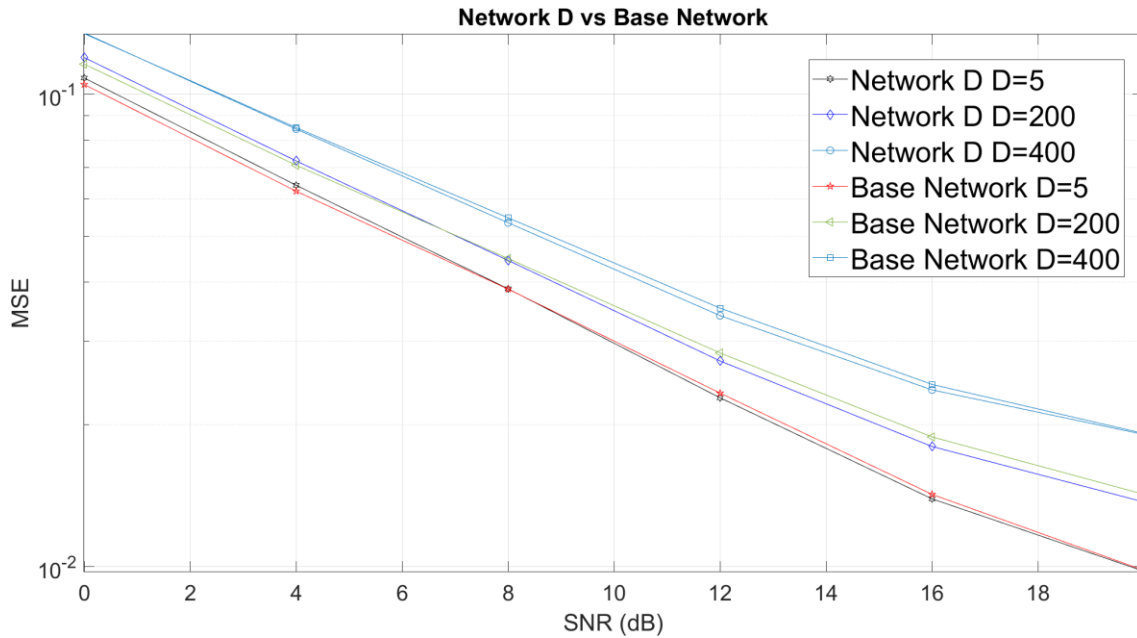


Figura 19- MSE da rede base vs MSE da rede D, ambas com 4096 sessões de treino

No entanto, é necessário criar heurísticas para avaliar o desempenho e a complexidade, uma vez que as redes A e B são muito menos complexas do que a rede de base e essas redes não têm um desempenho tão bom como a rede de base.

Por este motivo, foi criada uma heurística que consiste em quantificar o desempenho ou a complexidade, dependendo do peso que se queira dar a cada variável considerada. Dessa forma foram analisadas 3 perspectivas diferentes da heurística criada, cada uma das quais atribui um valor mais elevado a uma das variáveis, seja ela a complexidade ou o desempenho, para que seja possível ver em que aspeto é preferível utilizar as redes A e B em detrimento das redes C e D, bem como a rede de base, apesar de terem um desempenho inferior.

Redes	SNR 0 a 8 (MSE)	SNR 12 a 20 (MSE)	Complexidade
Rede A	0,0983684	0,04219345	1,389,658,816
Rede B	0,09906815	0,0433555	971,131,392
Rede C	0,09355935	0,0244259	2,273,193,216
Rede D	0,0940274	0,02651655	3,122,042,112
Rede Base	0,09451085	0,0271952	2,244,678,912

Tabela 3- Complexidade e MSE acerca de todas as redes analisadas

A tabela acima mostra os valores de complexidade das redes e os valores médios de MSE para cada um dos intervalos de SNR definidos. Com base nestes valores, são efetuados todos os cálculos para as 3 perspectivas consideradas.

$$Heuristic = \frac{\alpha}{MSE} + \frac{1 - \alpha}{Complexity * 10^{-9}} \quad (1)$$

A heurística acima é então utilizada para analisar as redes, variando apenas o valor da variável α , de modo a dar prioridade à complexidade ou ao desempenho das redes.

Para dar prioridade à complexidade, a heurística será utilizada com um valor de $\alpha = 10^{-2}$, obtendo-se os seguintes resultados:

Redes	Heurística (0 a 8 dB)	Heurística (12 a 20 dB)
Rede A	0.81	0.95
Rede B	1.12	1.25
Rede C	0.54	0.84
Rede D	0.42	0.69
Rede Base	0.55	0.81

Tabela 4- Valores obtidos para a heurística utilizada com um valor de $\alpha = 10^{-2}$

Por outro lado, para dar mais valor às redes com melhor desempenho, foi utilizado $\alpha=0,99$:

Redes	Heurística (0 a 8 dB)	Heurística (12 a 20 dB)
Rede A	10.07	23.47
Rede B	10	22.84
Rede C	10.59	40.54
Rede D	10.53	37.34
Rede Base	10.48	36.41

Tabela 5- Valores obtidos para a heurística utilizada com um valor de $\alpha = 0,99$

Para encontrar um equilíbrio entre os dois aspetos considerados, mas sempre privilegiando o desempenho da rede em detrimento da complexidade, utilizou-se $\alpha=0,5$:

Redes	Heurística 3 (0 a 8 dB)	Heurística 3 (12 a 20 dB)
Rede A	5.44	12.02
Rede B	5.56	12.05
Rede C	5.56	20.7
Rede D	5.48	19.02
Rede Base	5.51	18.61

Tabela 6- Valores obtidos para a heurística utilizada com um valor de $\alpha = 0,5$

Considerando todos os resultados obtidos para as diferentes operações efetuadas de acordo com as heurísticas criadas, podemos verificar que cada uma delas terá resultados diferentes. Se escolhermos uma situação em que o desempenho da rede é crucial, devemos optar pela rede C, pela rede D ou pela rede base, sendo que a rede C é a que tem melhor desempenho em relação a todas as outras redes.

Noutro contexto, em que o desempenho da rede não precisa de ser tão excelente, em que se prefere uma rede menos complexa, deve considerar-se as redes A ou B.

De um ponto de vista geral, em que é necessário encontrar um equilíbrio entre complexidade e desempenho, a rede que deve ser utilizada é a rede C.

Avaliando a precisão da heurística criada, esta fornece uma forma de priorizar cada variável que se queira considerar, seja a complexidade ou o desempenho (MSE). No futuro pode-se considerar esta heurística de forma a constatar em que situações é preferível utilizar redes com um desempenho inferior, mas com uma complexidade mais baixa. Dessa forma verifica-se que então em certas situações em que é exigido um poder de computação extremamente elevado e nesse sentido um desempenho excelente das redes neuronais, tais como, a transferência de grandes ficheiros, a realidade virtual ou a computação gráfica, as redes menos complexas não conseguem superar esses requisitos. Em sentido contrário, em que o poder de computação imposto não necessite de ser muito elevado, as redes A e B, que são redes com uma complexidade mais baixa, podem perfeitamente ser utilizadas, nomeadamente dispositivos móveis com

5 Conclusões e Trabalhos Futuros

5.1 Conclusões

Esta dissertação analisa a transição da tecnologia 5G para a 6G, centrando-se na superação dos desafios encontrados na 5G. As Redes Neurais Convolucionais (CNN) foram utilizadas para a estimação de canais, com o objetivo de adaptar estas técnicas ao 6G. Foram testadas diferentes arquiteturas de rede, incluindo camadas Depthwise, para reduzir a complexidade sem sacrificar o desempenho. Analisando o MSE em função da SNR, verificou-se que redes como a A e a B, embora menos complexas, podem não corresponder às expectativas de desempenho em determinadas situações, como a transferência de ficheiros, a computação gráfica e a realidade virtual, ou a gestão de grandes empresas, situações em que se exige um excelente desempenho das redes utilizadas.

Para avaliar as redes, foi criada uma heurística que avalia tanto o desempenho como a complexidade. Se a complexidade for a prioridade, as redes A e B destacam-se em situações em que o desempenho da rede não precisa de ser excelente, como os dispositivos móveis (Smartwatches, Air pods, etc.). Se valorizarmos o desempenho, destacam-se as redes C e D, bem como a rede de base. Se procura um equilíbrio, a rede C é a escolha mais sensata, oferecendo o equilíbrio ideal entre complexidade e desempenho.

Através do trabalho elaborado nesta dissertação, foi possível verificar que através do uso dos símbolos piloto OFDM é possível efetuar a estimação de canal, sendo que por isso pode perfeitamente ser adotado nos dois sistemas tanto MIMO-OFDM como SIMO-OFDM apresentados na secção 5.2. Através do uso da heurística criada pode se também retirar conclusões mais efetivas do efeito que as redes neurais poderão ter quando acionados os efeitos não lineares na comunicação.

Em conclusão, esta dissertação salienta a necessidade de uma abordagem equilibrada, tendo sempre em conta as circunstâncias em que as redes neurais são necessárias, quer num capítulo em que o desempenho da rede é crucial, quer quando é possível utilizar uma rede que satisfaça os requisitos e tenha uma menor complexidade.

5.2 Trabalhos Futuros

No futuro pode-se dar desenvolvimento ao tema abordado nesta dissertação, de forma a expandir ainda mais este tema:

- A extensão do esquema para comunicações MIMO – Em [31] discute-se a importância da estimação de canal em sistemas MIMO-OFDM e o seu impacto significativo no desempenho geral do sistema. É destacado os desafios colocados pelos canais rápidos que variam no tempo, onde as características do canal mudam rapidamente dentro da duração do símbolo, levando a complicações na estimação de canal. Neste artigo são explorados várias técnicas de estimação de canal em sistemas MIMO-OFDM, através do uso de diferentes filtros adaptativos.
- A inclusão de efeitos não lineares na comunicação – É proposta uma técnica de estimação de canal baseado em Deep Learning para sistemas SIMO-OFDM assistidos por uma Superfície Inteligente Reconfigurável (RIS) com limitações de hardware. A RIS é uma superfície plana constituída por elementos refletores passivos que podem alterar as fases dos sinais de entrada para melhorar o desempenho do sistema. O método proposto utiliza uma Rede Neuronal Profunda (DNN) não treinada para eliminar o ruído do canal efetivo obtido a partir da estimação convencional por mínimos quadrados (LS) baseada em pilotos. O canal sem ruído é então utilizado para encontrar os canais diretos e em cascata. A DNN não requer treino, garantindo baixa complexidade e latência reduzida. O método proposto é avaliado através de simulações e comparado com as técnicas convencionais de estimação de canais, demonstrando a sua elevada precisão e robustez face a deficiências de hardware [32].

6 Anexos

Nesta secção serão colocadas todas as simulações que não tiveram um efeito tão positivo quanto o esperado, além de outras experiências que foram realizadas de modo a tirar conclusões sobre algum conteúdo. Será também colocado um artigo que foi realizado através do âmbito desta dissertação.

6.1 Anexo A – Outras Simulações

Como referido anteriormente na secção 5.1, passou por impor-se Stride em cada camada da rede neuronal, de forma a baixar ainda mais a complexidade e analisar a eficiência da rede, uma vez que o Stride se refere ao tamanho do passo que o filtro utiliza para se mover pela entrada de cada convolução, ou seja o Stride determina quantas posições o filtro irá ser utilizado seja de forma horizontal ou vertical. Neste caso utilizou-se um Stride de 2 em cada camada da rede neuronal.

Considerando os gráficos abaixo, percebe-se que ao introduzir Stride na rede neuronal base, o desempenho da rede decaiu significativamente, passando a apresentar piores resultados que o estimador prático para todos os valores de SNR analisados.

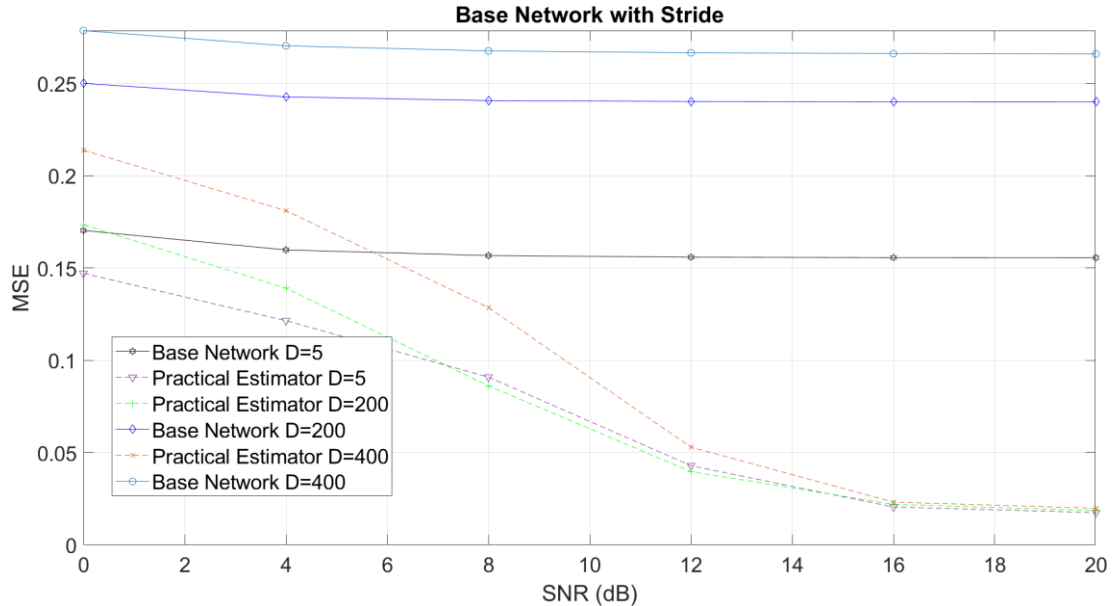


Figura 20- MSE da rede base com Stride

Ainda que a complexidade da rede baixe com a utilização de Stride, não é útil essa utilização, dado que a rede neuronal apresenta valores inferiores comparativamente ao estimador prático. De forma que ao se impor Stride a rede mantenha um bom desempenho comparativamente ao estimador prático, teria de se aumentar o tamanho do filtro, além do número de filtros utilizados por camada, o que faria aumentar a complexidade, não trazendo melhorias em comparação com as experiências anteriores.

Dessa forma a utilização do Stride foi descartada, sendo que todas as experiências para diante foram efetuadas através apenas da introdução de Padding.

O primeiro teste analisado foi feita a comparação entre a rede pré-definida no exemplo do Matlab, constituída por 5 camadas convolucionais em que é aplicada um Padding de tamanho 2 e entre uma rede onde uma dessas camadas é substituída por uma camada Depthwise e apenas mantida 4 das camadas convolucionais, de modo a tentar reduzir a complexidade da rede.

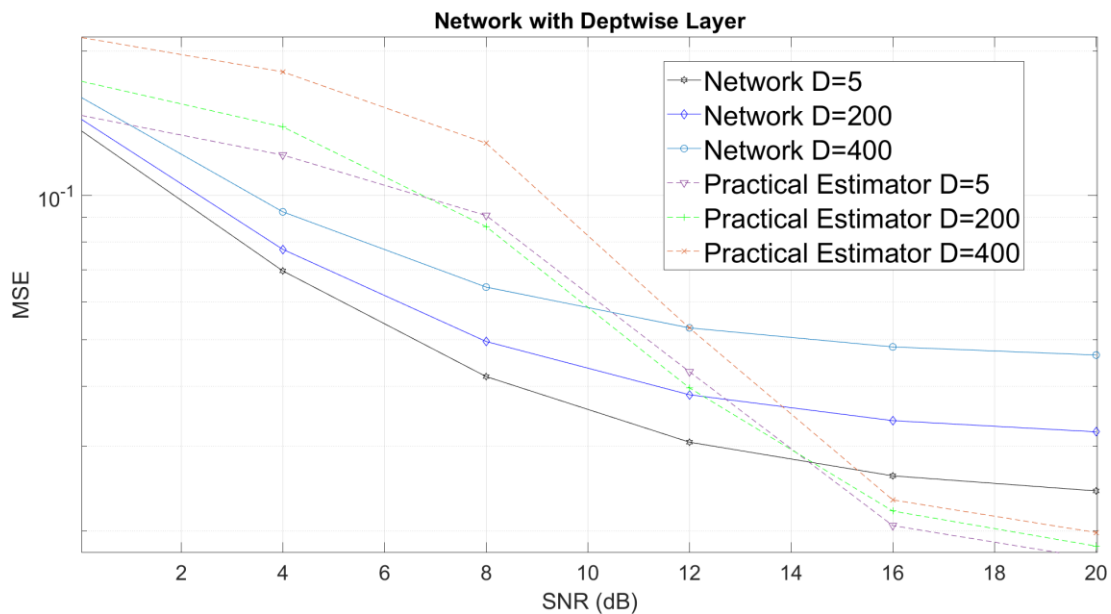


Figura 21- MSE da Rede com apenas uma camada Depthwise vs MSE do estimador prático

Observando a figura 23, que compara a rede com a camada Depthwise com o estimador prático, verifica-se que a rede neuronal apresenta um melhor desempenho que o estimador prático para valores de SNR mais baixo (com + ruído). À medida que o SNR vai aumentando, e a partir dos 16 dB o estimador prático passa a ter valores mais baixos de MSE comparativamente à rede neuronal para qualquer das 3 velocidades de Doppler utilizadas.

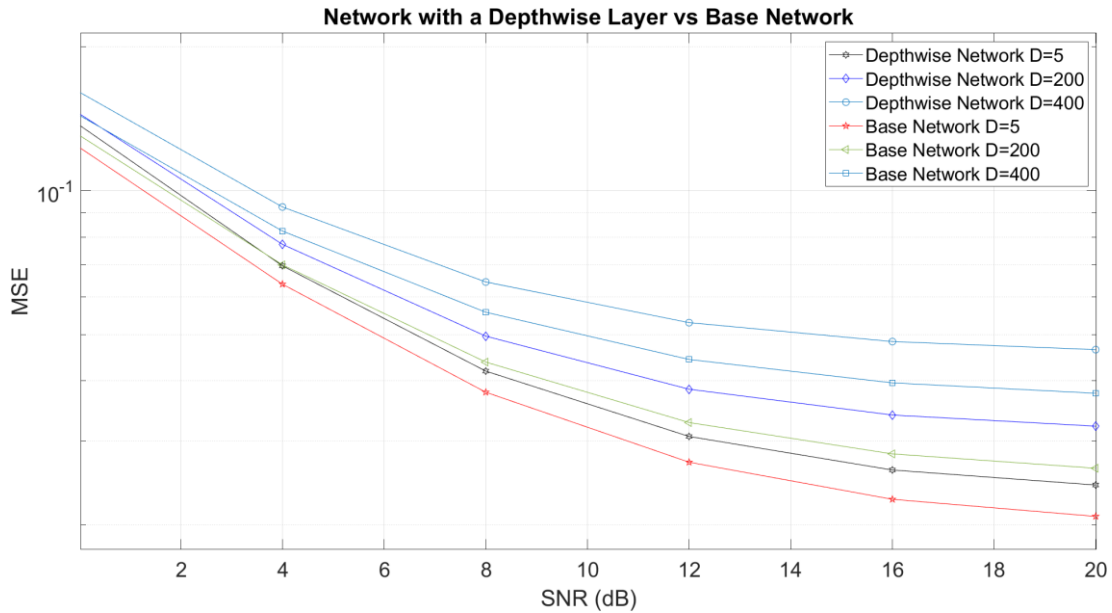


Figura 22- MSE da rede com apenas uma camada Depthwise vs MSE da rede base

Porém ao compararmos a rede com a rede base (Figura 24), denota-se que para todas as velocidades de Doppler analisadas, a rede decai o desempenho em todas, e ainda que tenha uma complexidade muito inferior que a rede base, não é o desempenho pretendido, e dessa forma procedeu-se a outros testes com mais camadas Depthwise adicionadas à rede, de forma a melhorar o desempenho da rede e manter uma complexidade inferior à rede base.

Outros testes que foram efetuados, foi a alteração do tipo de canal utilizado durante o treino da rede. Com o intuito de perceber se um canal fora dos selecionados para os que são usados para treinar a rede, mantém o desempenho da rede, utilizou-se o canal TDL-B, que é um canal com linha de vista, tal como o TDL-A, e que foi o utilizado anteriormente nas experiências anteriores e que é um dos canais usados para treino, para verificar o comportamento da rede com um canal que tem algumas características diferentes dos canais aplicados no treino.

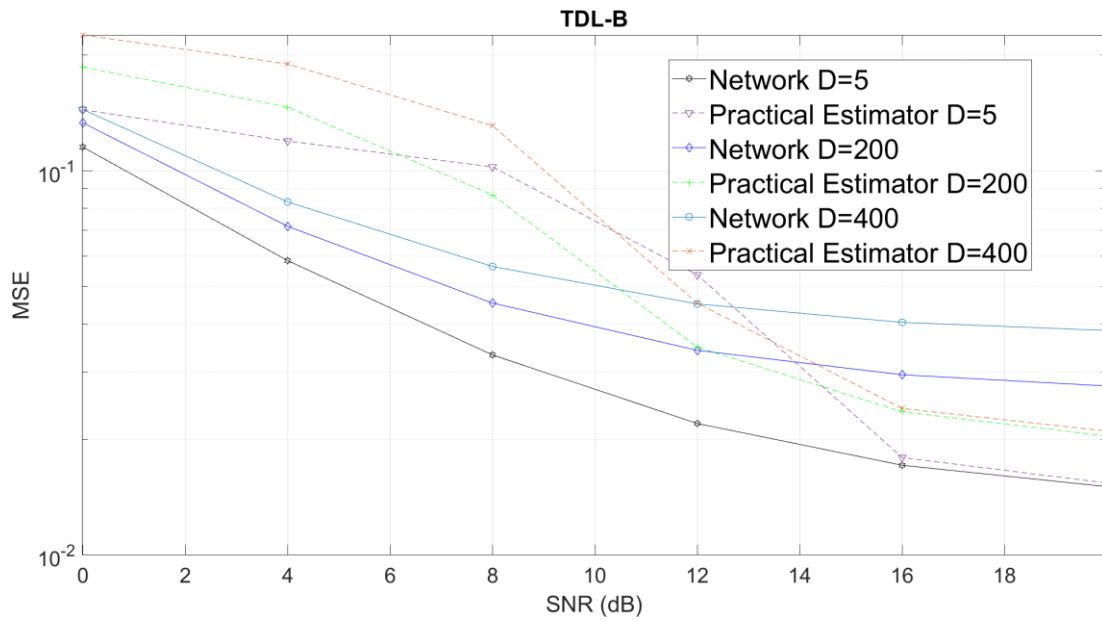


Figura 23- MSE da rede com a utilização de um canal (TDL-B) fora dos utilizados para o treino da rede vs MSE do estimador prático

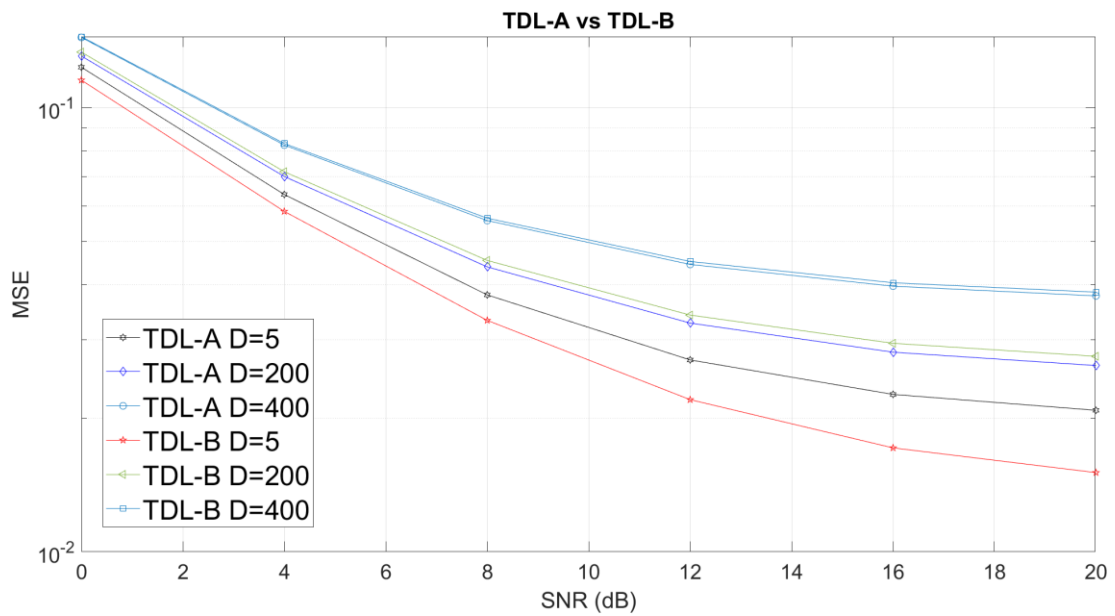


Figura 24- MSE da rede com a utilização de um canal (TDL-B) fora dos utilizados para o treino da rede vs MSE da rede com canal utilizado no treino (TDL-A)

Analisando as figuras 25 e 26, averigua-se que a rede com o canal TDL-B apresenta melhores valores que o estimador prático para todas as velocidades, e que apresenta valores bastante semelhantes com a rede onde o canal TDL-A é que é utilizado, o que evidencia que a rede consegue lidar bem com canais idênticos, porém que não tenham sido utilizados no treino.

Após verificar como seria o comportamento da rede para um canal que não é treinado, passou-se por testar a rede com um canal sem linha de vista, NLOS. Neste caso foi utilizado o canal TDL-E, e depois comparou-se os valores obtidos com o canal usado na rede base, o TDL-A.

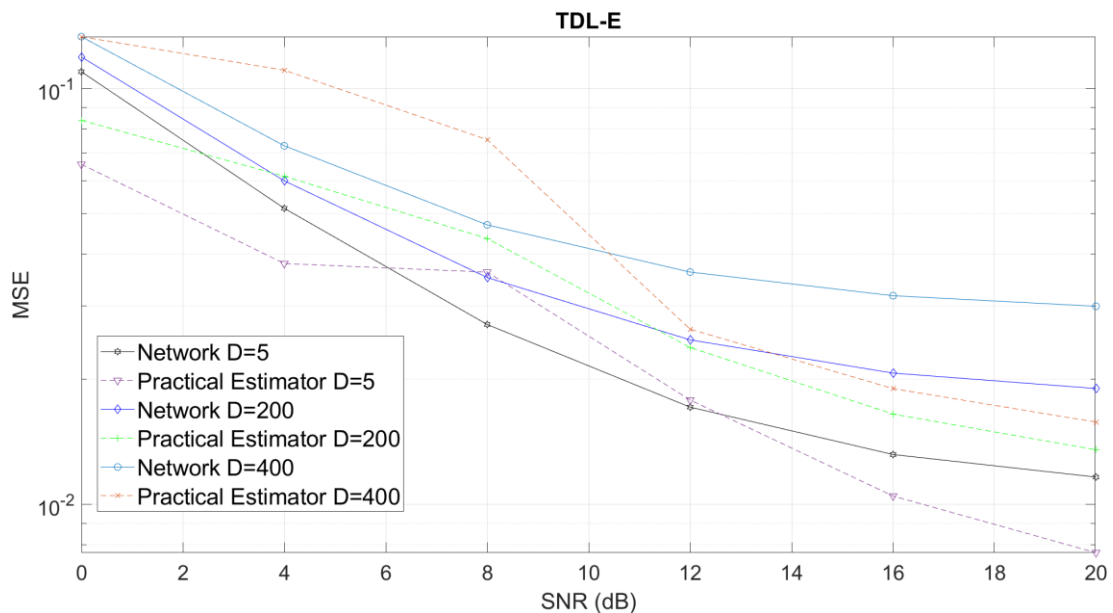


Figura 25- MSE da rede com a utilização de um canal (TDL-E) fora dos utilizados para o treino da rede vs MSE do estimador prático

Considerando a figura acima (Figura 27), verifica-se que a rede neuronal com o canal TDL-E acaba por revelar um desempenho mais baixo que o estimador prático nessas características, como podemos observar na figura 17, sendo que para uma velocidade de Doppler mais elevada, $D=400$, a rede já possui valores de MSE mais baixos relativamente ao estimador prático para valores de SNR mais baixos. Para a velocidade de Doppler mais baixa que foi testada, $D=5$, a rede tanto para situações com maior ou menor ruído tem uma eficiência inferior em relação ao estimador prático

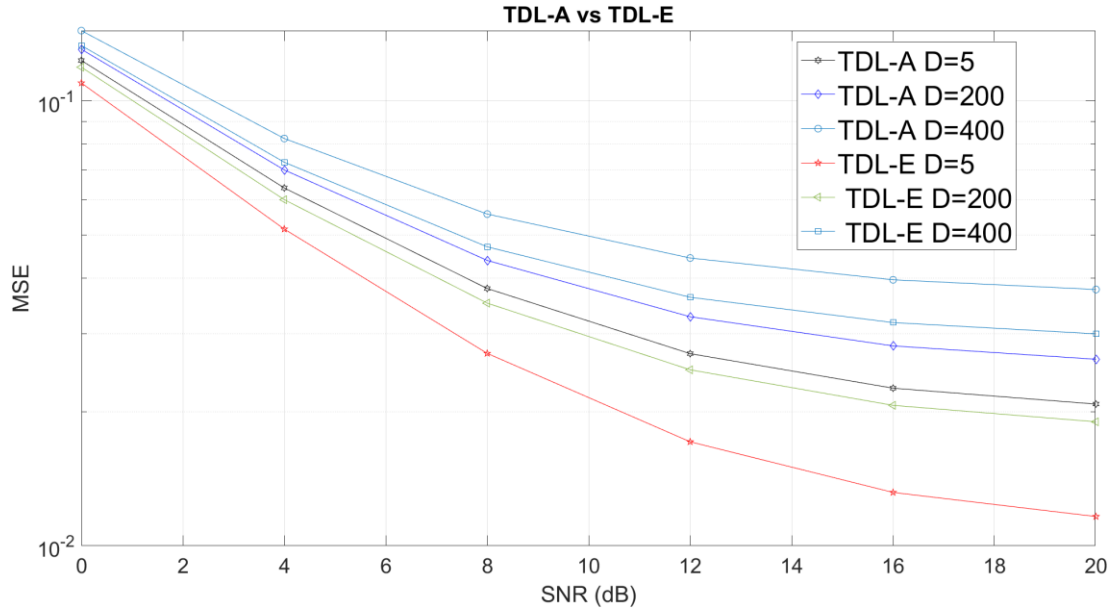


Figura 26 - MSE da rede com a utilização de um canal (TDL-E) fora dos utilizados para o treino da rede vs MSE da rede com canal utilizado no treino (TDL-A)

Analisando a figura 28, onde é comparado o desempenho da rede neuronal com o canal TDL-E em contraste com a rede neuronal com o canal TDL-A, averigua-se que a rede TDL-E até apresenta melhores resultados de MSE para todas as velocidades de Doppler testadas. Todavia, como visto anteriormente, a rede neuronal com o canal TDL-A apresenta um melhor desempenho que o estimador prático, ao passo que a rede neuronal com o canal TDL-E piora em comparação com o estimador prático. Dessa forma, não faz sentido utilizar o canal NLOS, uma vez que o estimador prático acaba sempre por ter melhores desempenhos que a rede neuronal. Foi então sempre considerado o canal TDL-A em todas as simulações efetuadas.

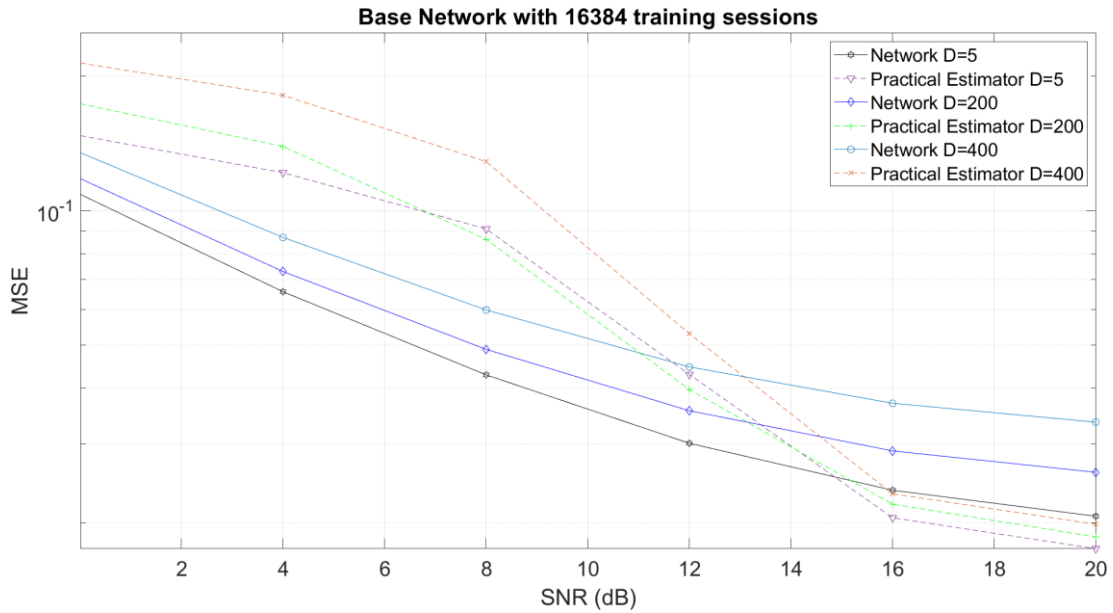


Figura 27- MSE da rede base com 16384 sessões de treino vs MSE do estimador prático

Para constatar se fazia sentido aumentar mais ainda os exemplos de treino para verificar se a rede acabaria por se tornar mais eficiente que o estimador prático para valores de SNR mais altos, utilizou-se o modelo pré-treinado do exemplo que utiliza 16384 exemplos de treino. Analisando a figura acima (Figura 29), averigua-se que mesmo com esse elevado número de exemplos de treino o estimador prático, a partir dos 16 dB torna-se mais eficiente que a rede neuronal. Assim sendo considera-se que para um nível de ruído mais baixo (SNR mais elevado), o estimador prático acaba por permanecer sempre mais eficiente que as redes neuronais.

6.2 Anexo B – Artigo

Use of Neural Networks for channel estimation in future 6G Networks

Ricardo Gaspar
ISCTE- Department of Information Sciences and Technologies
Samora Correia, Portugal
rcgaspar22@gmail.com

João Silva
ISCTE- Department of Information Sciences and Technologies
Lisboa, Portugal
joao.silva@iscte-iul.pt

Nuno Souto
ISCTE- Department of Information Sciences and Technologies
Lisboa, Portugal
nuno.souto@iscte-iul.pt

Abstract—6G will be the successor to 5G technology. With 5G technology, Orthogonal Frequency Division Modulation (OFDM) has been adopted and has become one of the main candidates for future communication systems. In this sense, the main task is to solve the problems encountered in 5G technology, and one of the aims of this article is to focus on solving these obstacles and adapting them to 6G technology, using neural networks and deep learning to infer channel estimation. To infer channel estimation, a Matlab model of a convolutional neural network (CNN) was adopted. Using this basic neural network, different types of layers were explored, such as Depthwise layers, to try to reduce the complexity and verify that the network maintains the same performance in terms of Mean Square Error (MSE) as a function of Signal to Noise Ratio (SNR). In the end, it will be possible to verify that the network with Depthwise layers ends up having a very similar performance to the predefined network.

Keywords— Complexity, MSE, channel estimation.

I. INTRODUCTION

The recent proliferation of new mobile devices, powered by artificial intelligence, is sparking debate about how mobile networks will evolve in the future. After the creation of 5G technology, the new 6G technology is being contextualised with the prospect that it could support AI services for network devices in addition to supporting mobile internet. In other words, 6G is being designed with an intelligent information system to drive modern AI technologies.

In 5G technology, particular emphasis has been placed on the "non-radio" aspect, and in particular two key technologies have been used for 5G: Software Defined Networks (SDN) and Network Functions Virtualisation (NFV), which have transformed modern communication networks into software-based networks [1]. However, these technologies will not be sufficient to support AI-based applications, so the new 6G

network will need to support a number of features, such as increased bandwidth, which is expected to be in the terahertz range, around 0.3-3 THz [2].

Deep learning has achieved impressive success when applied to neural networks. Even for complex neural networks that may be required to solve large-scale problems, the distributed parallel architectures and acceleration tools of deep learning are expected to result in high computational efficiency. Because of these advantages, DL is being applied to wireless communications.

In this paper, several tests have been carried out using convolutional neural networks (CNNs) for channel estimation [7]. To analyse the performance of the neural networks studied, the mean square error (MSE) as a function of the signal-to-noise ratio (SNR) was analysed in comparison with a practical estimator [5].

There is already a pre-defined example network made up of 5 convolutional layers, which will serve as a basis for comparison with the new networks tested, with the aim of improving the neural network to reduce its complexity and maintain good performance with respect to a practical estimator also tested.

At the end of all the tests carried out, a conclusion is drawn as to which networks are the best to use, always considering the level of complexity and performance of the network.

II. SYSTEM MODEL

The study of channel estimation in future 6G networks is based on a Matlab model related to data generation and training of a Convolutional Neural Network (CNN) for channel estimation in 5G systems. The system model consists of several steps, including data generation, training of the neural network and evaluation of the results.

First, the environment is set up where parameters such as training data size, sampling rate, grid size, number of transmit/receive antennas, PDSCH (Pilot Specific Downlink Symbols) configuration, etc. are defined.

Afterwards, a function is used that is responsible for generating training data for channel estimation. A simulation is run for each training example: A TDL (Tap Delay Line) channel model is created with different settings such as delay profile, delay spread, etc. An OFDM (Orthogonal Frequency Division Multiplexing) signal is generated with DM-RS (Downlink Specific Pilot Symbols) symbols inserted. The signal is transmitted over the TDL channel, AWGN (Additive White Gaussian Noise) is added, and reception is simulated. Channel estimation is performed by linear interpolation of the received DM-RS symbols. The received signal and the channel estimation are stored as training examples.

OFDM is an advanced modulation technique that uses pilot symbols to correct for channel distortion, ensuring accurate data transmission. It is a promising technique due to its high data transmission capacity, high bandwidth efficiency and robustness to multipath delay [18]. By interspersing these reference signals with the transmitted data, the OFDM system overcomes challenges such as fading and interference, ensuring reliable communication even in adverse conditions. In this way, the system used in this article uses OFDM pilot symbols to achieve accurate and robust data transmission.

After the data is generated, the architecture of the neural network is defined, where the architecture of the CNN is defined in the layer variable. The CNN is designed to process the images (real and imaginary parts) of the channel grid as input and predict the channel estimate as output.

Finally, the training of the neural network is defined, specifying all the data necessary to carry it out. The training and validation data are separated. The training options, optimiser and learning rate are set. The CNN is trained using the training and validation data.

III. NEURAL NETWORK-BASED CHANNEL ESTIMATION

The neural network architecture specified by the Matlab model consists of 5 convolutional layers, where the components are expressed by the number and size of the filters used, the number of output channels in that layer, and the parameters that influence the network, such as padding and stride. Each

convolutional layer is followed by the ReLU activation function, which is used to introduce non-linearity into the neural network.

One of the objectives of this article is to use the network predefined by the model as a basis, trying to replace as much as possible the convolutional layers, by Depthwise layers, in order to try to reduce the complexity of the network and maintain a good performance of the network. Thus, each Depthwise layer is composed of two convolutional layers of 1x1 filters. An example of the composition of a neural network used is shown in Figure 1.



Figure 1- Example of a network composition

IV. MODEL TRAINING

The process of testing a neural network involves 3 different processes: training the network, validating the network, and finally testing the network. Network training is the most important of these processes. During network training, the number of epochs used to train the network can be specified, and the more epochs used, the more iterations will be used during network training. After setting the network parameters to start training, a window will open (Figure 2) where you can check the variation of the root mean square error (RMSE) and the total losses over the epochs. From these RMSE values, the MSE values are then calculated and used to analyse the performance of the network over the signal-to-noise ratio.

The generation of new data is predefined to produce 256 training examples. This amount of data is sufficient to train a working channel estimation network on a CPU in a reasonable amount of time. However, it is possible to increase the number of training examples, and the higher the number, the better trained the network will be, and the better the network will perform.

V. COMPLEXITY AND PERFORMANCE

One way of evaluating the neural networks used has to do with the complexity of the network, requiring a balance between complexity and performance level. If a network is more complex, it is expected to perform better, but if a network with less complexity is not as efficient as a more complex network,

the best network must be evaluated considering both variables, performance and complexity.

To calculate the complexity of the network, only the multiplication operations for each layer in the neural network are used. The complexity is then calculated by multiplying the size of the filter vertically and horizontally, in this case the same, by the number of filters used. This value is then multiplied by the output size of the previous layer, and so on between layers. The comparison of complexity between neural networks is then made using the percentage difference between the two.

A. Stride

Initially, the base network was tested, but instead of imposing the Padding parameter on the network, Stride was added to the network to see if the network could maintain good performance. This parameter refers to the step size that the filter uses to move through the input of each convolution, i.e., Stride determines how many positions the filter is used in, either horizontally or vertically. The stride is therefore introduced into each layer of the neural network to further reduce complexity and analyse the efficiency of the network. A stride of 2 has therefore been introduced into each layer of the neural network. When the stride is introduced into the neural network, the performance of the network decreases significantly and performs worse than the practical estimator for all SNR values analysed.

B. Padding

After verifying that Stride significantly degraded the performance of the network compared to the practical estimator, we simply introduced Padding, a parameter used to adjust the size of the inputs, with the Padding parameter set to "same" so that the convolution operation has the same size as the input, to maintain consistent dimensions throughout the network layers. Two neural networks were then tested, in addition to the base network, where some of the convolutional layers of the base network are replaced by Depthwise layers. Network A replaces 2 convolutional layers with a size 5 filter with 2 Depthwise layers. In network B, an additional Depthwise layer was added to network A. In this way, the performance of the neural network improves as the complexity of the network is reduced, and the more Depthwise layers the neural network has, the lower its complexity.

VI. SIMULATION RESULTS

Several experiments have been carried out to demonstrate the performance of the neural networks used and the practical estimator. The simulations are evaluated using a graph that quantifies the MSE as a function of the signal-to-noise ratio, in a range between 0 and 20 dB. The tests carried out identified

which networks could be used as alternatives to the base network without compromising performance.

Two identical tests were performed, but with different amounts of training. First, all networks were tested with only 256 training sessions, and then the number of training sessions was increased to 4096.

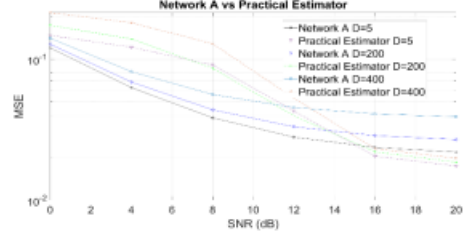


Figure 2- MSE of the practical estimator vs MSE of network A both with 256 training sessions

Figure 3 shows that, as expected, network A significantly outperforms the practical estimator at lower SNR values (with less noise), just like the base network. Analysing the results obtained in comparison with the network given in the example (Figure 4), network A, with a lower complexity than the base network, performs as well as the base network, even better for lower SNR values, only being outperformed for values with lower noise.

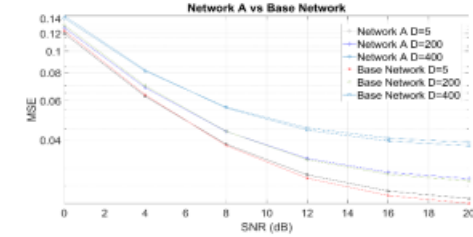


Figure 3- MSE of the base network vs MSE of network A both with 256 training sessions

Evaluating the conclusions of the previous experiment, a new Depthwise layer was added, leaving 3 Depthwise layers in the network, thus forming Network B, to analyse whether the network could still improve compared to the base network and compared to Network A.

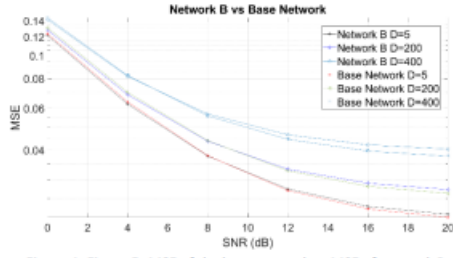


Figure 4- Figure 5- MSE of the base network vs MSE of network B both with 256 training sessions

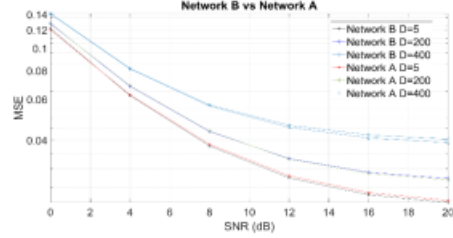


Figure 6- MSE of network B vs MSE of network A both with 256 training sessions

As can be seen in Figures 5 and 6, Network B added little in terms of network performance compared to the base network and Network A.

After experimenting with 256 training sessions, the number of trainings was increased to 4096 to see if the networks with Depthwise layers with more training outperformed the base network.

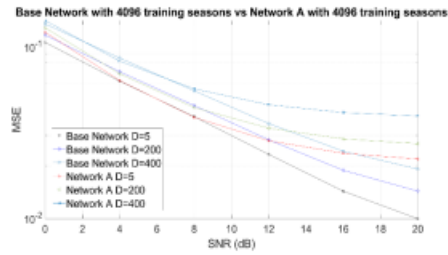


Figure 8- MSE of the base network vs MSE of network A both with 4096 training sessions

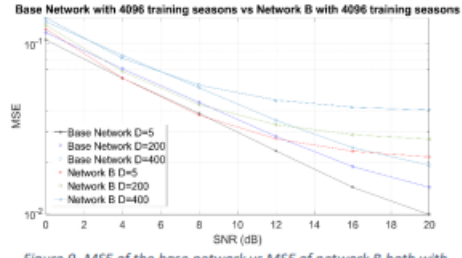


Figure 9- MSE of the base network vs MSE of network B both with 4096 training sessions

The figures above show comparisons between networks A and B and the base network, all with 4096 trainings. The results show that the base network with more training also improved its performance in the end, with better MSE values than the other networks with Depthwise layers. Due to the greater complexity of the network, it was possible to conclude that the base network ends up performing better than the other networks (A and B) as the number of training sessions increases, as can be seen in the example with 256 training sessions, where network B behaves very similarly to the base network, and in this case with 4096 training sessions it ends up a little further away. In this way, it was found that by removing complexity from the pre-defined network, the networks ended up reducing their performance relative to the pre-defined network. However, to show that it is possible to improve the network so that its performance improves relative to the base network, network C was created, which has the same 5 convolutional layers as the base network, plus 2 depth layers, and network D, which has 6 convolutional layers. These networks therefore end up having a higher complexity than the base network and therefore improve performance compared to the base network.

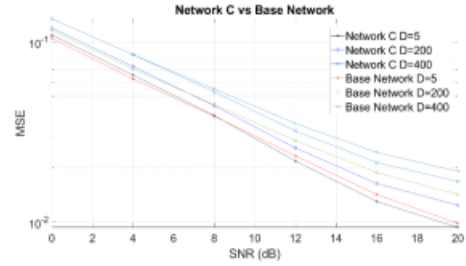


Figure 7- MSE of the base network vs MSE of network C both with 4096 training sessions

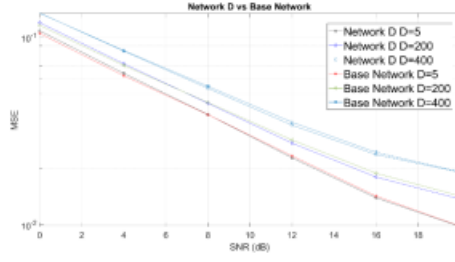


Figure 10- MSE of the base network vs MSE of network C both with 4096 training sessions

However, it is necessary to create heuristics to evaluate performance and complexity, since networks A and B are much less complex than the base network and these networks do not perform as well as the base network.

For this reason, a heuristic has been created which consists of quantifying performance or complexity depending on the weight to be given to each variable considered. In this way, 3 different perspectives of the heuristic created were analysed, each of which gives a higher value to one of the variables, be it complexity or performance, so that it is possible to see in which aspect it is preferable to use networks A and B to the detriment of networks C and D, as well as the basic network, despite their lower performance.

Networks	SNR 0 to 8 (MSE)	SNR 12 to 20 (MSE)	Complexity
Network A	0.0983684	0.04219345	1,389,658,816
Network B	0.09906815	0.0433555	971,131,392
Network C	0.09355935	0.0244259	2,273,193,216
Network D	0.0940274	0.02651655	3,122,042,112
Base Network	0.09451085	0.0271952	2,244,678,912

Table 1 - Complexity and MSE for all the networks analysed

The table above shows the complexity values of the networks and the average MSE values for each of the SNR ranges defined. On the basis of these values, all the calculations are made for the 3 different heuristics created.

$$Heuristic = \frac{\alpha}{MSE} + \frac{1 - \alpha}{Complexity * 10^{-9}} \quad (1)$$

The above heuristic is then used to analyse the networks, varying only the value of the variable α in order to prioritise either the complexity or the performance of the networks.

To prioritise complexity, the heuristic will be used with a value of $\alpha = 10^{-2}$ giving the following results:

Networks	Heuristic (0 to 8 dB)	Heuristic (12 to 20 dB)
Network A	0.81	0.95
Network B	1.12	1.25
Network C	0.54	0.84
Network D	0.42	0.69
Base Network	0.55	0.81

Table 2 - Values obtained for the heuristic used with a value of $\alpha = 10^{-2}$

Conversely, to give more value to the best performing networks, $\alpha = 0.99$ was used:

Networks	Heuristic (0 to 8 dB)	Heuristic (12 to 20 dB)
Network A	10.07	23.47
Network B	10	22.84
Network C	10.59	40.54
Network D	10.53	37.34
Base Network	10.48	36.41

Table 3 - Values obtained for the heuristic used with a value of $\alpha = 0.99$

In order to find a balance between the two aspects considered, but always favouring network performance over complexity, $\alpha = 0.5$ is used:

Networks	Heuristic 3 (0 to 8 dB)	Heuristic 3 (12 to 20 dB)
Network A	5.44	12.02
Network B	5.56	12.05
Network C	5.56	20.7
Network D	5.48	19.02
Base Network	5.51	18.61

Table 4 - Values obtained for the heuristic used with a value of $\alpha = 0.5$

Considering all the results obtained for the different operations carried out according to the heuristics created, we can see that each one will have different results. If we choose a situation where the performance of the network is crucial, we should opt for network C, network D or the base network, with network C having the best performance compared to all the other networks.

In another context, where the performance of the network doesn't need to be so excellent, where a less complex network is preferred, you should consider networks A or B.

From a general point of view, where a balance has to be struck between complexity and performance, the network that should be used is network C.

VII. CONCLUSION

This article examines the transition from 5G to 6G technology, focusing on overcoming the challenges encountered in 5G. Convolutional Neural Networks (CNNs) were used for channel estimation, with the aim of adapting these techniques to 6G. Different network architectures were tested, including Depthwise layers, to reduce complexity without sacrificing performance. By analysing MSE as a function of SNR, we found that networks such as A and B, although less complex,

may not meet performance expectations in certain situations, such as file transfer, computer graphics and virtual reality, or the management of large enterprises, situations in which excellent performance is required from the networks used.

In order to evaluate the networks, a heuristic has been created to assess both performance and complexity. If complexity is the priority, networks A and B stand out in situations where network performance does not need to be excellent, such as mobile devices (Smartwatches, Air pods, etc.). If you value performance, networks C and D stand out, along with the base network. If you are looking for a balance, the C network is the most sensible choice, offering the ideal balance between complexity and performance.

In conclusion, this article has highlighted the need for a balanced approach, always considering the circumstances in which neural networks are needed, whether in a chapter where the performance of the network is crucial, or where a network can be used that meets the requirements and has a lower complexity.

REFERENCES

- [1] Khaled B. Letaief, Wei Chen, Yuanming Shi, Jun Zhang, and Ying-Jun Angela Zhang, "The Roadmap to 6G – AI Empowered Wireless Networks", *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84-90, 2019.
- [2] Demos Serghiou, Mohsen Khalily, Tim W. C. Brown, and Rahim Tafazolli, "Terahertz Channel Propagation Phenomena, Measurement Techniques and Modeling for 6G Wireless Communication Applications: A Survey, Open Challenges and Future Research Directions", *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 1957-1996, 2022.
- [3] Hongji Huang, Jie Yang, Hao Huang, Yiwei Song, and Guan Gui, "Deep Learning for Super-Resolution Channel Estimation and DOA Estimation Based Massive MIMO System", *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549-8560, 2018.
- [4] Dong Liu, Chengjian Sun, Chenyang Yang, and Lajos Hanzo, "Optimizing Wireless Systems Using Unsupervised and Reinforced-Unsupervised Deep Learning", *IEEE Network*, vol. 34, no. 4, pp. 270-277, 2020.
- [5] Hao Ye, Geoffrey Ye Li, and Bing-Hwang Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems", *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114-117, 2018.
- [6] Aliaa Said Mousa Mohammed, Abdelkarim Ibrahim Abdelkarim Taman, Ayman M. Hassan, and Abdelhalim Zekry, "Deep Learning Channel Estimation for OFDM 5G Systems with Different Channel Models", *Wireless Pers Commun*, 2022.
- [7] Peihao Deng, Hua Zhang, Geoffrey Ye Li, Ivan Simões Gaspar, and Navid Naderi Alizadeh, "Deep CNN-Based Channel Estimation for mmWave Massive MIMO Systems", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 989-1000, 2019.
- [8] C. Nguyen, T. M. Hoang and A. A. Cheema, "Channel Estimation Using CNN-LSTM in RIS-NOMA Assisted 6G Network," in *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 1, pp. 43-60, 2023.
- [9] C. Silpa, A. Vani and K. Rama Naidu, "Implementation of MIMO-OFDM system with deep learning-based channel estimation and channel equalization," *2022 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, pp. 173-177, 2022.
- [10] Redhwan Q. Shaddad, Esam M. Saif, Husam M. Saif, Zaid Y. Mohammed, and Ahmed H. Farhan, "Channel Estimation for Intelligent Reflecting Surface in 6G Wireless Network via Deep Learning Technique", *1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, 2021.
- [11] Xiao Chen, Miao Liu, Guan Gui, Bamidele Adebsi, Haris Gacanan, and Hikmet Sari, "Complex Deep Neural Network Based Intelligent Signal Detection Methods for OFDM-IM Systems", *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit): Physical Layer and Fundamentals (PHY)*, 2021.
- [12] Ravinder Kaur, and Dr. Rinkesg Mittal, "Implementation of Neural Network for Channel Estimation in OFDM Network", *2nd International Conference on Contemporary Computing and Informatics*, 2016.
- [13] Elisa Zimaglia, Daniel G. Riviello, Roberto Garelli, and Roberto Fantini, "A Deep Learning-based Approach to 5G-New Radio Channel Estimation", *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit): Physical Layer and Fundamentals (PHY)*, 2021.
- [14] H. Tahir, M. Shahbaz Khan and M. Owais Tariq, "Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles," *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 587-594, 2021.
- [15] G. Priyadharshini and D. R. Judie Dolly, "Comparative Investigations on Tomato Leaf Disease Detection and Classification Using CNN, R-CNN, Fast R-CNN and Faster R-CNN," *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1540-1545, 2023.
- [16] P. Rost et al., "Mobile network architecture evolution toward 5G," *Infocommunications J.*, vol. 9, no. 1, pp. 24-31, 2017.
- [17] Sukanya Samanta and Venkata Sridhar T., "Modified Slow Fading Channel Estimation Technique and Fast Fading Channel Estimation Technique For OFDM Systems", *3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018)*, pp.1638-1643, 2018.
- [18] Ján Šterba and Dušan Kocur, "Pilot symbol aided channel estimation for OFDM system in frequency selective Rayleigh fading channel", *19th International Conference Radioelektronika*, 2009.

7 Referências

- [1] Khaled B. Letaief, Wei Chen, Yuanming Shi, Jun Zhang, and Ying-Jun Angela Zhang, “The Roadmap to 6G – AI Empowered Wireless Networks “, *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84-90, 2019.
- [2] Demos Serghiou, Mohsen Khalily, Tim W. C. Brown, and Rahim Tafazolli, “Terahertz Channel Propagation Phenomena, Measurement Techniques and Modeling for 6G Wireless Communication Applications: A Survey, Open Challenges and Future Research Directions “, *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 1957-1996, 2022.
- [3] Hongji Huang, Jie Yang, Hao Huang, Yiwei Song, and Guan Gui, “Deep Learning for Super-Resolution Channel Estimation and DOA Estimation Based Massive MIMO System “, *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549-8560, 2018.
- [4] Dong Liu, Chengjian Sun, Chenyang Yang, and Lajos Hanzo, “Optimizing Wireless Systems Using Unsupervised and Reinforced-Unsupervised Deep Learning “, *IEEE Network*, vol. 34, no. 4, pp. 270-277, 2020.
- [5] Hao Ye, Geoffrey Ye Li, and Bing-Hwang Juang, “Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems “, *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114-117, 2018.
- [6] Aliaa Said Mousa Mohammed, Abdelkarim Ibrahim Abdelkarim Taman, Ayman M.Hassan, and Abdelhalim Zekry, “Deep Learning Channel Estimation for OFDM 5G Systems with Different Channel Models “, *Wireless Pers Commun*, 2022.
- [7] Peihao Dung, Hua Zhang, Geoffrey Ye Li, Ivan Simões Gaspar, and Navid Naderi Alizadeh, “Deep CNN-Based Channel Estimation for mmWave Massive MIMO Systems “, *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 989-1000, 2019.
- [8] Linglong Dai, Ruicheng Jiao, Fumiyuki Adachi, H. Vincent Poor, and Lajos Hanzo, “Deep Learning for Wireless Communications: An Emerging Interdisciplinary Paradigm “, *IEEE Wireless Communications*, vol. 27, no. 4, pp. 133-139, 2020.
- [9] Elisa Zimaglia, Daniel G. Riviello, Roberto Garello, and Roberto Fantini, “A Deep Learning-based Approach to 5G-New Radio Channel Estimation “, *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit): Physical Layer and Fundamentals (PHY)*, 2021.
- [10] Redhwan Q. Shaddad, Esam M. Saif, Husam M. Saif, Zaid Y. Mohammed, and Ahmed H. Farhan, “Channel Estimation for Intelligent Reflecting Surface in 6G Wireless Network via Deep Learning Technique”, *1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, 2021.
- [11] Xiao Chen, Miao Liu, Guan Gui, Bamidele Adebisi, Haris Gacanin, and Hikmet Sari, “Complex Deep Neural Network Based Intelligent Signal Detection Methods for OFDM-IM Systems “, *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit): Physical Layer and Fundamentals (PHY)*, 2021.
- [12] Ami Munshi, and Srija Unnikrishnan, “Compressive Sensing in Channel Estimation for SISO and MIMO OFDM Systems “, *Proceedings of TENCON 2018*, 2018.

- [13] Ravinder Kaur, and Dr. Rinkesg Mittal, "Implementation of Neural Network for Channel Estimation in OFDM Network ", *2nd International Conference on Contemporary Computing and Informatics*, 2016.
- [14] Eric Pierre Simon, Laurent Ros, Hussein Hijazi, and Mounir Ghogho, "Joint Carrier Frequency Offset and Channel Estimation for OFDM Systems via the EM Algorithm in the Presence of Very High Mobility ", *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 754-765, 2012.
- [15] Yu Tian, Gaofeng Pan, and Mohamed-Slim Alouini, "Applying Deep-Learning-Based Computer Vision to Wireless Communications: Methodologies, Opportunities, and Challenges ", *IEEE Open Journal of the Communications Society*, vol. 2, pp. 132-143, 2020.
- [16] Guangxu Zhu, Dongzhu Liu, Yuqing Du, Changsheng You, Jun Zhang, and Kaibin Huang, "Towards an Intelligent Edge: Wireless Communication Meets Machine Learning ", *IEEE Communications Magazine*, vol. 58, no. 1, pp. 19-25, 2020.
- [17] Yinchuan Li, Xiaodong Wang, and Robert L. Olsen, "Unfolded Deep Neural Network (UDNN) for High Mobility Channel Estimation ", *IEEE Wireless Communications and Networking Conference (WCNC)*, 2021.
- [18] Ken Peffers, Tuure Tuunanen, and Marcus A. Rothenberger, "A design science research methodology for information systems research ", *Journal of Management Information Systems*, 2007.
- [19] P. Rost *et al.*, "Mobile network architecture evolution toward 5G," *Infocommunications J.*, vol. 9, no. 1, pp. 24-31, 2017.
- [20] Sukanya Samanta and Venkata Sridhar T., "Modified Slow Fading Channel Estimation Technique and Fast Fading Channel Estimation Technique For OFDM Systems", *3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018)*, pp.1638-1643, 2018.
- [21] Ján Šterba and Dušan Kocur, "Pilot symbol aided channel estimation for OFDM system in frequency selective Rayleigh fading channel", *19th International Conference Radioelektronika*, 2009.
- [22] 1994-2023 The MathWorks, Inc. *Deep Learning Data Synthesis for 5G Channel Estimation*. [Online]. Disponivel em: <https://www.mathworks.com/help/5g/ug/deep-learning-data-synthesis-for-5g-channel-estimation>.
- [23] Z. Xie, X. Chen, Y. Liu and M. Defense, "A channel estimation algorithm for MIMO-SCFDE systems over fast time-varying multipath channels," *2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pp. 1-6, 2017.
- [24] S. Montaha, S. Azam, A. K. M. R. H. Rafid, M. Z. Hasan, A. Karim and A. Islam, "TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study," in *IEEE Access*, vol. 10, pp. 60039-60059, 2022.
- [25] C. Nguyen, T. M. Hoang and A. A. Cheema, "Channel Estimation Using CNN-LSTM in RIS-NOMA Assisted 6G Network," in *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 1, pp. 43-60, 2023.

- [26] C. Silpa, A. Vani and K. Rama Naidu, "Implementation of MIMO-OFDM system with deep learning-based channel estimation and channel equalization," *2022 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, pp. 173-177, 2022.
- [27] R. U. Murshed, Z. B. Ashraf, A. H. Hridhon, K. Munasinghe, A. Jamalipour and M. F. Hossain, "A CNN-LSTM-Based Fusion Separation Deep Neural Network for 6G Ultra-Massive MIMO Hybrid Beamforming," in *IEEE Access*, vol. 11, pp. 38614-38630, 2023
- [28] Q. Lu, T. Lin and Y. Zhu, "Channel Estimation and Hybrid Precoding for Millimeter Wave Communications: A Deep Learning-Based Approach," in *IEEE Access*, vol. 9, pp. 120924-120939, 2021.
- [29] H. Tahir, M. Shahbaz Khan and M. Owais Tariq, "Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles," *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 587-594, 2021.
- [30] G. Priyadharshini and D. R. Judie Dolly, "Comparative Investigations on Tomato Leaf Disease Detection and Classification Using CNN, R-CNN, Fast R-CNN and Faster R-CNN," *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1540-1545, 2023.
- [31] H. Kaur, M. Khosla and R. K. Sarin, "Channel Estimation in MIMO-OFDM System: A Review," *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 974-980, 2018.
- [32] N. Ginige, K. B. S. Manosha, N. Rajatheva and M. Latva-aho, "Untrained DNN for Channel Estimation of RIS-Assisted Multi-User OFDM System with Hardware Impairments", *Center for Wireless Communications, University of Oulu, Finland*, 2021.