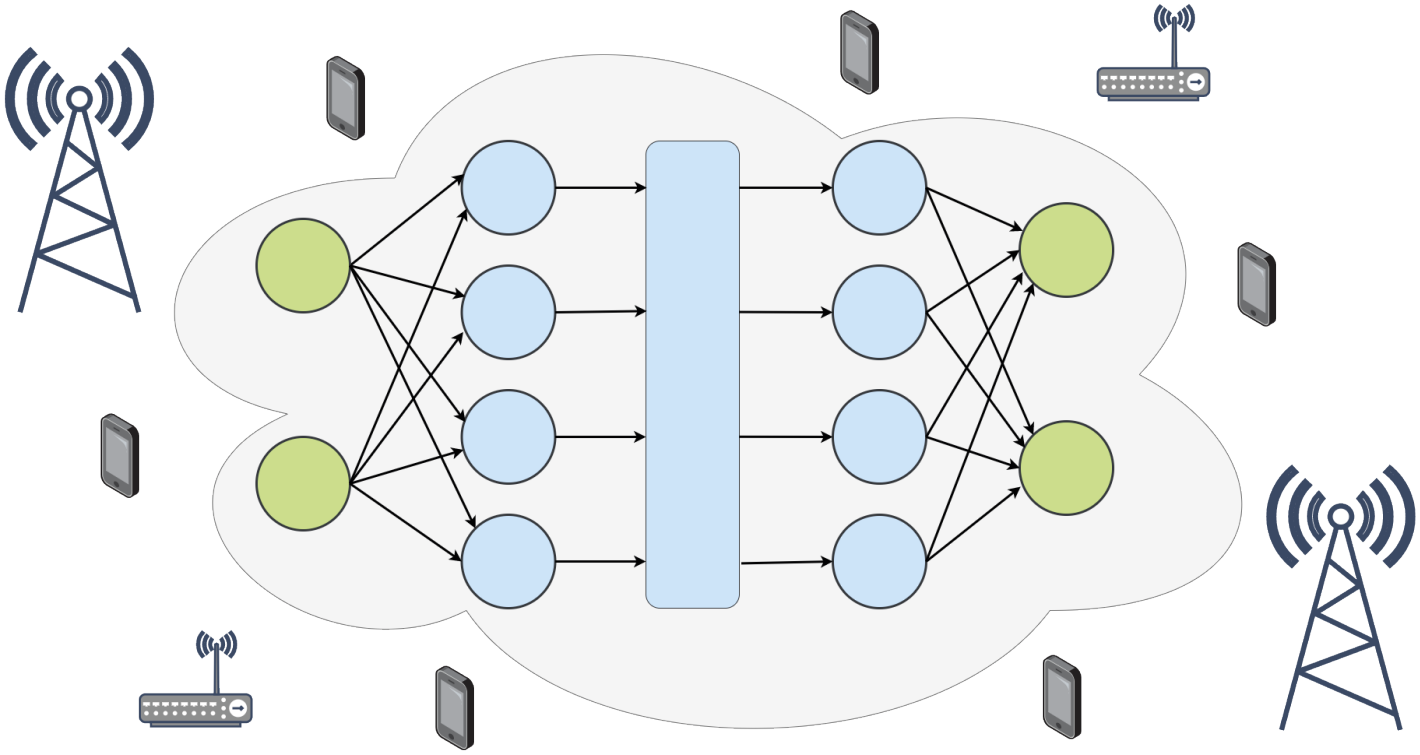
FOI

Overview of Machine Learning in Communication Systems

ERIK AXELL, PATRIK ELIARDSSON, KRISTOFFER HÄGGLUND,
PER BRÄNNSTRÖM, CAROLIN SVENSSON



FOI-R--5275--SE
ISSN 1650-1942 December 2021

Erik Axell, Patrik Eliardsson, Kristoffer Hägglund,
Per Brännström, Carolin Svensson

Overview of Machine Learning in Communication Systems

Titel	Översikt av maskininlärning i kommunikationssystem
Title	Overview of Machine Learning in Communication Systems
Rapportnr / Report No.	FOI-R--5275--SE
Månad / Month	December / December
Utgivningsår / Year	2021
Antal sidor / Pages	57
ISSN	1650-1942
Kund / Customer	FM
Forskningsområde	Ledningsteknologi
FoT område	Ledning och MSI
Projektnr / Project No.	A74029
Godkänd av / Approved by	Christian Jönsson
Ansvarig avdelning	Ledningssystem

Abstract

The advancement of artificial intelligence and machine learning (ML) have made a significant impact in many research areas during the last decade. ML techniques will most certainly be implemented in some parts of future wireless communication systems. The amount of published research about ML for communications has increased enormously during the last five years. ML provides new opportunities but also new vulnerabilities towards, for example, new types of adapted jamming and spoofing attacks and difficulties in predicting and guaranteeing system performance.

This report provides an overview of existing research results about using ML in communication systems. An extensive literature review has been conducted, covering many areas of communications. The large amount of research literature demonstrates that there are various applications where ML can be exploited. However, it is important to keep in mind that this does not necessarily imply that ML is the best choice, or even that it provides any benefits compared to traditional methods, for all types of applications. Applications where ML may be beneficial are, for example, such that require reduced computational complexity with near-optimal performance or model-free learning that can be adapted to phenomena that are completely or partially unknown. Examples of such algorithms include resource allocation at different levels of complex communication networks, traffic modelling, signal detection or classification of unknown signals, and end-to-end learning of channels that are unknown or difficult to model.

A crucial aspect for the development and use of ML algorithms, in any application, is the access to a sufficiently large set of training data with good enough quality. The creation of such data sets may be burdensome and costly, and it is of uttermost importance to pass that hurdle to be able to use ML algorithms in communication systems.

This report has only briefly touched upon the vulnerabilities of using ML techniques for commmunication applications. Future studies need to put more emphasis on exploitable vulnerabilities and the consequences of these on communication performance. For example, the overall knowledge is limited of adversarial attacks on this type of algorithms and defense mechanisms against such attacks.

Adversarial ML attacks may also be exploited for purposes that are benefitial in defense and security applications, such as LPI/LPD communications. Such applications should be further studied.

Keywords: AI, machine learning, deep learning, communication system.

Sammanfattning

Framstegen inom artificiell intelligens och maskininlärning (ML) har gjort ett stort genomslag på många forskningsområden under det senaste decenniet. ML-tekniker kommer med största sannolikhet att implementeras i vissa delar av framtida trådlösa kommunikationssystem. Mängden publicerad forskning om ML för kommunikation har ökat enormt under de senaste fem åren. ML ger nya möjligheter men också nya sårbarheter, mot exempelvis nya typer av anpassad störning och vilseledning, samt svårighet att garantera och prediktera prestanda i systemet.

Denna rapport ger en översikt av tillgängliga forskningsresultat om användning av ML för kommunikationssystem. En omfattande litteraturstudie har genomförts, som täcker många områden inom kommunikation. Den stora mängden forskningslitteratur visar att det finns flertalet tillämpningar där ML kan utnyttjas. Det är däremot viktigt att ha i åtanke att detta inte nödvändigtvis innebär att ML är det bästa alternativet, eller ens att det ger fördelar jämfört med traditionella tekniker, för alla typer av tillämpningar. Kommunikationstillämpningar där ML kan ge fördelar är exempelvis sådana som kräver reducerad beräkningskomplexitet med nära optimal prestanda eller databaserad inlärning av fenomen som är helt eller delvis okända. Exempel på sådana algoritmer är resursallokering på flera nivåer av komplexa kommunikationsnätverk, trafikmodellering samt signaldetektion eller -klassificering av okända signaler.

En avgörande aspekt av utvecklingen och utnyttjandet av ML-baserade algoritmer, för alla tillämpningar, är åtkomsten till stora mängder träningsdata av god kvalitet. Generering av sådana datamängder kan vara besvärlig och kostsam, och det är av yttersta vikt att överkomma detta hinder för att kunna utnyttja ML-algoritmer i kommunikationssystem.

Denna rapport har bara ytligt berört de sårbarheter som kommer av att använda ML för kommunikationstillämpningar. Framtida studier behöver lägga större tonvikt på exploaterbara sårbarheter och konsekvenserna av dessa på kommunikationsprestanda. Exempelvis är den sammantagna kunskapen om fientliga attacker mot dessa typer av algoritmer, samt försvarsmekanismer mot sådana attacker, begränsad.

Fientliga attacker mot ML-algoritmer kan också utnyttjas för syften som kan ge fördelar i försvar- och säkerhetstillämpningar, såsom smygradioegenskaper. Sådana tillämpningar bör studeras vidare.

Nyckelord: AI, maskininlärning, djupinlärning, kommunikationssystem.

Contents

List of Abbreviations

AI artificial intelligence 9, 10, 13
AWGN additive white Gaussian noise 16, 23, 30, 35

BCE binary cross entropy 20, 27
BEP bit error probability 16
BER bit error rate 15, 18, 35, 45
BP belief propagation 21, 22, 46
BPSK binary shift keying 15, 16, 29

CIR channel impulse response 34
CNN convolutional neural network 10, 17, 18, 19, 22, 24, 26, 27, 29, 30, 31, 32, 33, 34, 45, 46, 47, 48, 49, 50
CSI channel state information 17, 18, 19, 32, 33, 35, 44, 45, 50

DBN deep belief networks 15, 44
DIP deep image prior 19, 45
DL deep learning 10, 15, 16, 17, 18, 22, 24, 25, 26, 29, 35, 36, 38
DQPSK differential quadrature phase shift keying 28

E2E end-to-end 26, 27, 48

FCNN fully connected neural network 10, 18, 19, 24, 28, 29, 34, 37, 47, 50
FFT fast fourier transform 49
FGSM fast gradient sign method 35

GAN generative adversarial network 11, 24, 27, 45, 47, 48
GNN graph neural network 10, 24, 26, 47

HF high frequency 31, 49

IRS intelligent reflecting surface 25, 47

LDPC low-density parity-check 22
LO local oscillator 30
LOS line-of-sight 34
LS least square 17, 18, 19, 34, 45
LSTM long short-term memory 18, 21, 25, 29, 30, 45, 46, 47, 49

MAC medium access control 13, 24, 47
MAP maximum a posteriori 21
MIMO multiple input multiple output 17, 18, 19, 23, 24, 33, 35, 44, 45, 47
MISO multiple input single output 25, 47
ML machine learning 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 22, 28, 29, 30, 31, 34, 35, 36, 37, 38, 39, 41, 44, 50
MMSE minimum mean square error 17, 18, 19, 45
MPC multipath components 33
MSE mean square error 17, 20, 27
MUSIC multiple signal classification 32, 50

NBP neural belief propagation 21, 22, 46
NLOS non-line-of-sight 32, 34

NN neural network 17, 19, 20, 21, 24, 26, 27, 28, 29, 30, 34, 35, 36, 38, 44, 45, 46, 47, 48, 50
NND neural network decoder 21
NOMA non-orthogonal multiple access 24, 25, 47

OFDM orthogonal frequency-division multiplexing 18, 19
OFDMA orthogonal frequency-division multiple access 24
OTA over-the-air 27, 28, 30, 34, 35, 36

PCA Principal Component Analysis 31

QPSK quadrature phase shift keying 29

ResNet residual neural network 30, 49
RL reinforcement learning 20, 24, 25, 26, 27, 46, 47, 48
RMSE root mean square error 10
RNN recurrent neural network 10, 21, 22, 46
RSRP reference signal received power 34, 50
RSS receive signal strength 32

SAE stacked autoencoder 15, 44
SCM sample covariance matrix 29, 49
SINR signal-to-interference-plus-noise ratio 25, 47
SNR signal-to-noise ratio 16, 18, 20, 22, 29, 30, 31

TTN twice training network 15, 44

UE user equipment 18, 23, 47
UWB ultra wide-band 34

V2I vehicle-to-infrastructure 25, 47
V2V vehicle-to-vehicle 25, 47
VAE variational autoencoder 10

WLS weighted least squares 34

1 Introduction

The introduction of artificial intelligence (AI) and machine learning (ML) have made a significant impact in many research areas during the last decade. ML techniques will most certainly be implemented in some parts of future wireless communication systems. Figure 1.1 shows the number of publications in the IEEEXplore database during the last 20 years related to the search terms 'cognitive radio' and the combination of 'machine learning' and 'communication'. It is evident that the amount of published research about ML for communications has increased enormously during the last five years. ML provides new opportunities but also new vulnerabilities towards, for example, new types of adapted jamming and spoofing attacks and difficulties in predicting and guaranteeing system performance.

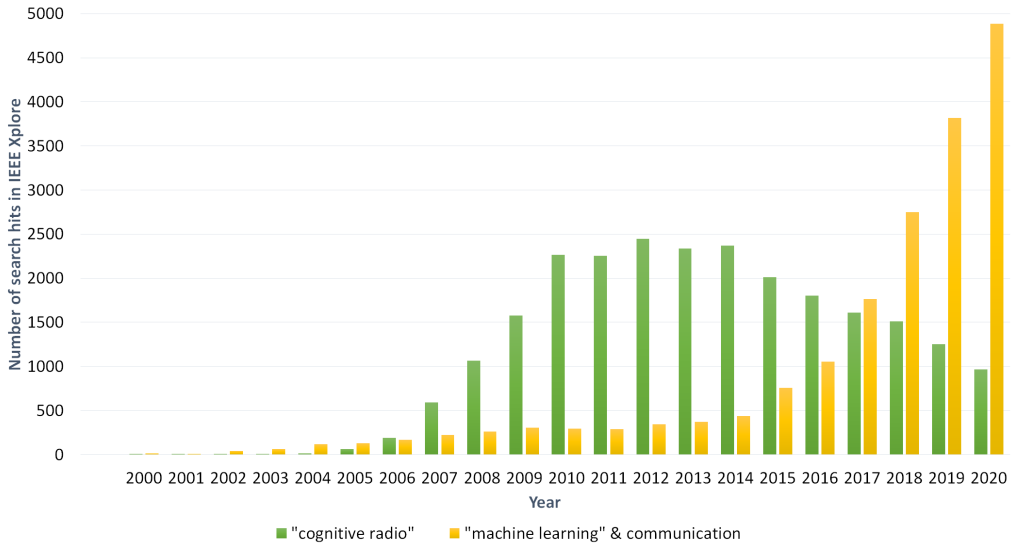


Figure 1.1: Number of publications in the IEEE Xplore database from the last 20 years for the search terms *cognitive radio* and the combination of *machine learning* and *communication*.

1.1 Purpose

The purpose of this report is to provide an overview of existing research literature about ML in wireless communications and to identify useful and probable applications of ML in future wireless communication systems.

1.2 Method

The literature review in this took the Research Library[1] of the IEEE Communications Society (ComSoc) Machine Learning for Communications (MLC) Emerging Technology Initiative (ETI) as a starting point. This library contains more than 800 articles and, as shown in Figure 1.1, the total amount of research literature in this area is impossible to study in detail. Therefore, a selected subset of publications from the IEEE ComSoc MLC ETI research library, and the references therein, have been studied. In addition, newer publications that cite the selected set of publications, and were deemed the most interesting, were also included in the study. The selected publications have been found mainly through IEEE Xplore and Google Scholar.

[1] https://mlc.committees.comsoc.org/research-library/

1.3 Reading Instructions and ML Basics

Readers of this report are expected to have knowledge of wireless communications and at least basic knowledge of ML and deep learning (DL). Some basic concepts of DL that appear repeatedly throughout the report are briefly mentioned in the following. Readers that are not familiar with these concepts are encouraged to read [1], [2] for a brief overview or [3] and [4] for more thorough walk-throughs of DL and traditional ML respectively.

Artificial Intelligence (AI) is a wide technology field where machine learning (ML) is one sub-area. ML with deep neural networks, deep learning (DL) [3], is one area within ML. DL has been developed over a long time, especially for applications in image processing, and is currently the dominant method in that area.

The recent boom in popularity for DL is its application to other areas than image processing. The company Deepmind created a lot of publicity with AlphaGo [5] and AlphaStar [6] by beating professional players in the board game Go and the computer game Starcraft 2. The development of translation and speech recognition in cellphones and speech controlled assistants has reached many homes. The car industry is continuing to drive development towards autonomous cars.

Deep neural networks are built up by multiple network layers and can be combined in many ways. There are a few commonly used structures which have shown to be particularly effective. Each network layer produces an output vector that is determined by the input via a deterministic function, which is characterized by a parameter, or weight, vector. The following common types of network layers are most relevant for applications in communications.

- Fully connected neural network (FCNN), where all inputs are connected to all outputs. Sometimes also called dense neural network.
- Convolutional neural network (CNN), which is based on convolutions of the input and is one of the biggest technological breakthroughs in image processing.
- Recurrent neural network (RNN) creates internal states from inputs and previous states, which are in turn used for future states and outputs.
- Graph neural network (GNN) are neural models that capture the dependence of graphs via message passing between the nodes of graphs [7].

These can be combined with each other and there are also many variants of these basic layer types. A neural network can be trained using one of the three basic strategies: supervised, unsupervised or reinforcement learning. For further details see [3], [4].

The following basic network architectures are commonly used in various applications, also for communications.

- Autoencoder and variational autoencoder (VAE) contains a coder-decoder network structure to compress and recreate a dataset. The goal function for an autoencoder is to minimize the root mean square error (RMSE). The VAE is similar to the autoencoder, but also different because it is a generative method that tries to find the correct distribution of the data instead of minimizing RMSE.

- Generative adversarial network (GAN) uses two competing networks, a generative network that tries to create fake data and a discriminating network that tries to find the difference between fake and real data. GAN can be used to create realistic fakes (so called deep fakes).
- *Attention* is a term that refers to an effect that enhances the important parts of the input data and fades out the rest. This way more computing power is focused on the more important data for improved results.

1.4 Outline

The report is organized as follows. Chapter 2 briefly describes ongoing research activities and Chapter 3 presents a review of existing literature about ML for wireless communications. A discussion about the possibilities and challenges of using ML in different applications of a communication system is provided in Chapter 4, and Chapter 5 concludes the report.

2 Related International Research Activities

This chapter describes ongoing international research activities related to ML for wireless communications and robust ML.

2.1 IEEE ComSoc MLC ETI

The IEEE ComSoc MLC ETI was founded in November 2018. The aim of the MLC ETI is to foster research and innovation surrounding the use of ML for the physical and medium access control (MAC) layers for all types of communication systems, such as wireless, optical, satellite, and molecular. The ETI organizes conference workshops, tracks, sessions, industry symposia, tutorials, summer schools, data science competitions, as well as special issues in journals. The ETI aims to establish common data sets and related benchmarks and invite authors to open-source their code for reproducible research.

2.2 H2020

H2020 is the European Commission's framework for research and innovation. Plenty of research projects have been founded by this program. H2020 is the 8th research program in EU and started in 2014 and ended in 2020. A short overview of H2020 projects with focus on AI or ML for communication applications is described in this section.

The project *Artificial Intelligence Aided D-band Network for 5G Long Term Evolution* (ARIADNE) started in November 2019. ARIADNE aims to bring together a novel high frequency (beyond 100 GHz) radio architecture, an advanced wireless connectivity based on reconfigurable metasurfaces, and an enhanced network management supported by AI to establish a new type of intelligent communication system beyond 5G. The project aim to use ML for channel modelling, resource allocation and network deployment optimization [8].

Technologies beyond 5G are also studied in the project *Network intelligence for adaptive and self-Learning mobile networks* (DAEMON). DAEMON will set forth a pragmatic approach to network intelligence (NI) design. The project does systematic analysis of which NI tasks are appropriately solved with AI models, providing a set of guidelines for the use of ML in network functions. For those problems where AI is a suitable tool, DAEMON aims to design tailored AI models that respond to the specific needs of network functions, taking advantage of the most recent advances in ML [9].

Private networks is a part of 5G and beyond 5G communication networks. The project *Beyond 5G multi-tenant private networks integrating cellular, WiFi, and LiFi*[1]*, powered by artificial intelligence and intent based policy* (5G-CLARITY) focuses on beyond 5G systems for private networks. By supporting AI-driven management, 5G-CLARITY aims to enable effective provision of network slices, managing and optimizing their performance. This is intended to support faster fulfillment of user and business intents while enabling efficient resource sharing during the entire lifetime of slices. AI-driven management also drives network automation by greatly reducing the need for human intervention [10].

[1] Light fidelity (LiFi) is a bidirectional wireless system that transmits data via LED or infrared light.

2.3 DARPA RFMLS

The Defense Advanced Research Projects Agency (DARPA) issued a broad agency announcement for a new Radio Frequency Machine Learning Systems (RFMLS) program in the fall of 2017. The goal of the RFMLS program is to develop the foundations for applying modern data-driven ML to the RF Spectrum domain.

Under the RFMLS program, systems will seek to learn to perform four specific tasks. The four solutions can be combined and applied to address DoD operational needs in the RF Spectrum.

1. RF fingerprinting. The RFMLS will aim to learn to recognize a specific transmitter based on the unique RF fingerprint naturally imparted by hardware imperfections within that transmitter. This task focuses on learning RF features.
2. RF fingerprint enhancement. To further enhance wireless security, a communication system learns to modify it's transmit waveforms to enhance its natural fingerprint. This task focuses on learning to synthesize waveforms.
3. Spectrum awareness. Traditional systems which monitor the RF spectrum use narrow bandwidths and relatively simple strategies (such as the frequency of transmission) to identify the signals occupying the wireless spectrum. Availability of commodity analog-to-digital converters with wide bandwidths combined with proliferation of software defined radios, spectrum sharing, and general wireless technology growth, challenge these approaches. RFMLS will learn to understand the difference between important and unimportant signals present in large bandwidths in order to build more useful and accurate spectrum awareness.
4. Autonomous RF system configuration. To further enhance spectrum awareness performance, a RFMLS will seek to learn how to best tune and configure its hardware resources in order to maximize the number of important signals discovered in harsh RF environments.

3 Literature Review on ML for Communications

In this chapter, a literature review on ML for radio communication is presented. ML techniques are used either in the transmitter, in the receiver or in both the transmitter and receiver. Figure 3.1 gives an overview of different applications of ML in the transmitter-receiver chain. The different applications correspond to separate sections of the literature review and the use of ML for these applications is further described in those sections. A majority of the applications, found in the conducted literature review, of ML in the radio domain are at the receiver side, see Figure 3.1.

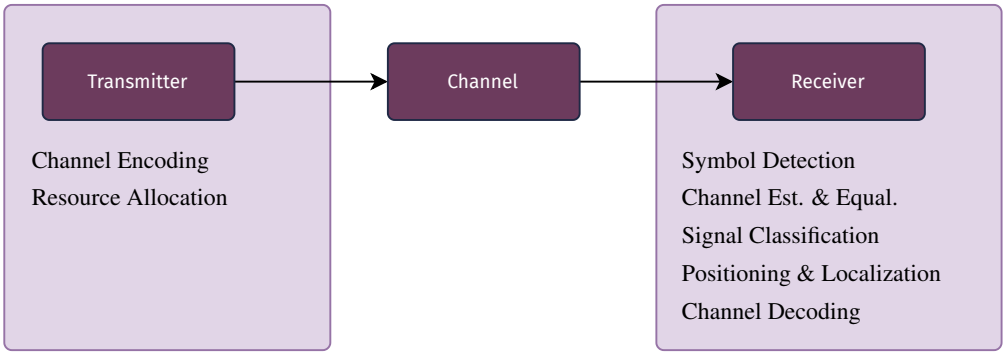


Figure 3.1: Overview of the use of ML techniques in the transmitter and the receiver.

A tutorial-like overview of ML for radio communications is given in [1]. In [1], the authors give examples of what ML is and when to use it. More examples of ML-applications in communications are discussed in [11]. In [2], a number of applications for the physical layer are suggested where ML might perform well. End-to-end reconstruction with autoencoders to jointly learn transmitter and receiver implementations is introduced. The need for a common dataset that can be used for benchmarking various ML models and algorithms for communication applications (similar to those available for image recognition) is identified in [2].

A block diagram of a digital communication system is shown in Figure 3.2. To the left in Figure 3.2 is the transmitter and to the right the receiver. Several references are highlighted in conjunction to a block or a set of blocks in Figure 3.2. This is to give an overview of the application and where the focus of existing research literature on ML in digital radio communication is.

3.1 Symbol Detection

Channel estimation and equalization reduces the impact of the channel on the received symbols at the cost of computational complexity. Using a DL method to directly demodulate the received symbols without any channel estimation and equalization is proposed in [12]. Three DL methods, deep belief networks (DBN), stacked autoencoder (SAE) [13] and a twice training network (TTN) are proposed and evaluated by simulations of a binary shift keying (BPSK)-modulated signal. A channel that introduces multi-path effects and inter-symbol interference is applied to the symbols before demodulation with the purposed methods takes place. The simulation results show that all three methods outperform a maximum likelihood demodulator without any prior channel equalization. This comparison is, however, not fair. A more relevant comparison would have been between a maximum likelihood demodulator after channel equalization. However, the bit

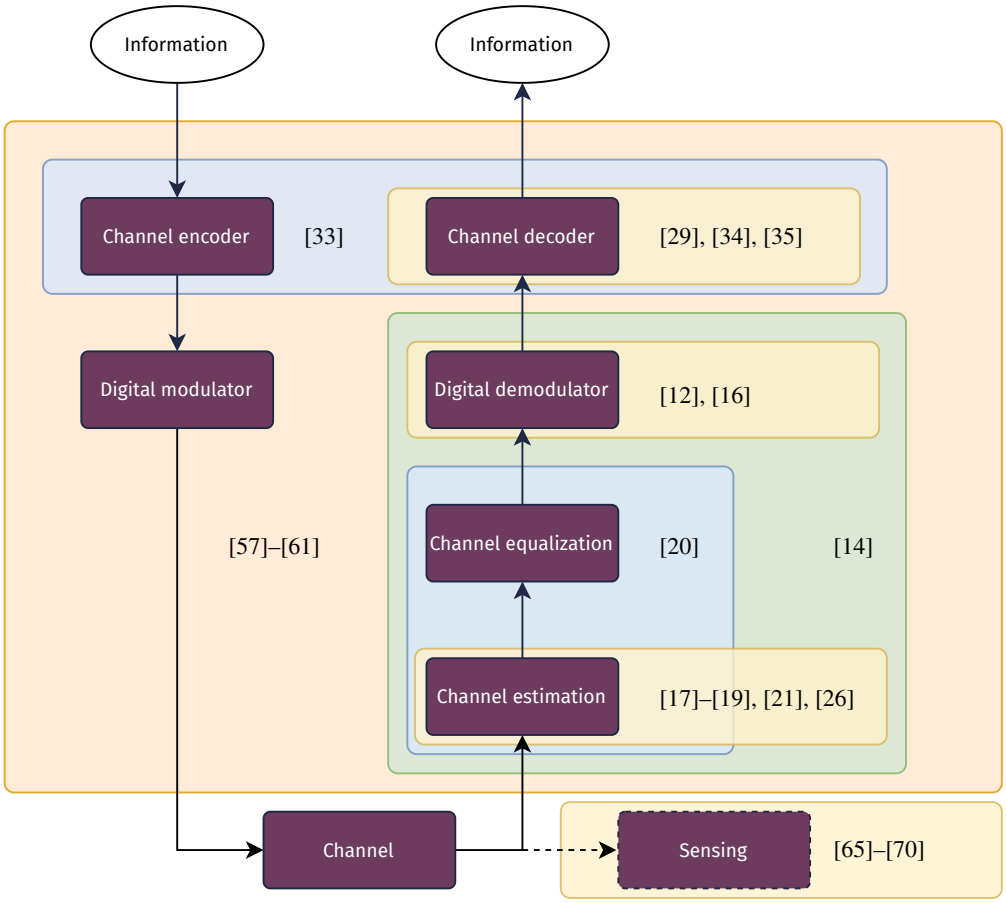


Figure 3.2: Block diagram of a digital communication system. References to research literature of ML for the different blocks are shown.

error rate (BER) for the ML methods is not as low as the theoretical bit error probability (BEP) for BPSK in an additive white Gaussian noise (AWGN) channel.

A similar approach is used in [14]. The algorithm proposed in [14] uses a deep autoencoder to estimate the received signal along with an additional layer for symbol detection, which are jointly trained and fine-tuned. The proposed extended autoencoder architecture is designed to remove the noise effects, mitigate frequency selective fading channel effects and extract signal features as well as frequency selective fading channel effects, hence arriving at correct symbol detection. Simulation results of BER performance under multipath fading (up to ten channel taps) demonstrate a signal-to-noise ratio (SNR) gain of approximately 6 dB on average compared to the theoretical result of flat fading. It is observed that training with a higher SNR can generalize well for low SNRs. Training the model with a range of SNRs, e.g. from low to high SNR, might cause the DL model to unlearn what have been learned when it sees a wide range of outliers.

Machine learning is introduced to the Viterbi algorithm in [15]. The log-likelihood computational part of the Viterbi algorithm was identified to require full knowledge of the channel input-output statistical relationship and an ML based architecture was proposed to replace the conventional log-likelihood computation part of the Viterbi algorithm. Numerical results show that the ML modified Viterbi algorithm outperforms the original

Viterbi algorithm when there are channel state information (CSI) uncertainties. For the case with perfect knowledge of the CSI, the ML modified Viterbi algorithm approaches the performance of the original Viterbi algorithm.

In [16], deep neural network (NN) in the context of multiple input multiple output (MIMO) detection is proposed. In that work, two cases are considered: constant and varying channel. In the first case, the MIMO channel is constant and the network learns a detector for a specific system. The second case is a more difficult case in which the parameters are known, yet changing, and a single detector must be learned for multiple varying channels. The numerical results show that deep networks can achieve state-of-the-art accuracy with significantly lower complexity while providing robustness against ill-conditioned channels and misspecified noise variance.

Table A.1 summarizes the reviewed research literature in the application of symbol detection. All of the reviewed works have used synthetically generated complex baseband signals for training and evaluation of their work.

3.2 Channel Estimation, Equalization and Modeling

Accurate CSI is crucial in wireless communication in order to approach Shannon's channel capacity limit. However, exactly estimating the CSI is extremely difficult. Traditionally, least square (LS) and minimum mean square error (MMSE) based algorithms are used to estimate the CSI with a reasonable computationally complexity.

A low-complexity channel estimator inspired by the MMSE channel estimator structure is suggested with a CNN as the core component in [17]. The computational complexity of the MMSE channel estimator for MIMO is very high, under certain Toeplitz and shift-invariance structures of the covariance matrix the computational complexity can be reduce to $\mathcal{O}(M \log M)$, where M is the channel dimension. Same computational complexity is achieved with the CNN-MMSE channel estimator. The CNN is trained with phase and path gains, first for a single path channel model, secondly for a channel model with three propagation paths. For the channel model with one path there is not much potential for the CNN methods, but for the more complicated and realistic channel model the CNN-MMSE channel estimator outperforms state-of-art approaches.

In [18] a DL-based channel estimator for a time-varying Rayleigh-fading channels is proposed. The channel estimator is able to dynamically track the channel status without any prior knowledge about the channel model and its statistical characteristics. The results show that the suggested estimator has better mean square error (MSE) performance compared to the traditional algorithms.

In mobile scenarios, the transmitted signals often undergo doubly selective fading (i.e., both frequency- and time-selective fading) due to multipath effects and Doppler spread. In [19], an online DL-based channel estimator using a deep neural network for doubly selective channels is proposed. The neural network is first trained with simulated data in an offline manner and then tracks the dynamic channel in an online manner. The numerical results show that the proposed method outperforms the linear MMSE estimator in terms of both efficiency and robustness, and also demonstrates the great potential of DL on channel estimation and tracking. Since the proposed DL-based estimator is data-driven and does not rely on knowledge of channel statistics, it could be a promising

candidate when the channel models are unknown or difficult to model analytically.

A learning-based scheme to realize channel estimation and equalization of an orthogonal frequency-division multiplexing (OFDM) system is proposed in [20]. At the receiver side, a single feed-forward network is introduced in between channel estimation (classic LS-based decision-directed) and demodulation. The introduced single feed-forward network equalizes channel effects. From the demodulation, the decision-directed estimation of pilot symbols are fed back to the channel estimator. Simulation results show approximately the same performance as that of a classic MMSE approach.

In [21], a DL-based approach for joint channel estimation and symbol detection for OFDM systems is proposed. The channel estimation network is designed to replace the conventional interpolation between time-frequency resources in the classic pilot aided channel estimation. This is done by exploiting the time and frequency correlation of wireless channel and capitalizing on the image super resolution technique. Simulation results show that the proposed method can significantly improve the channel estimation accuracy, and yield better BER performance compared to the conventional zero-forcing detector and regularized zero-forcing detector. The proposed DL-based algorithm exhibit good generalization ability and is quite robust to the variation of channel parameters such as the operating SNR.

In frequency division duplex mode of a massive MIMO system, the downlink CSI should be sent to the base station through feedback links so that the potential MIMO gains can be exhibited. However, the feedback quantities resulting from these approaches need to be scaled linearly with the number of transmit antennas and are prohibitive in a massive MIMO regime. The challenge of CSI feedback in massive MIMO systems has motivated numerous studies, and several works have focused mainly on reducing feedback overhead by using the spatial and temporal correlation of CSI. In [22], a DL-based method is studied to compress and recover the CSI at the base station. The studied method is an autoencoder where the encoder uses CNN in the user equipment (UE) to compress the CSI. In the base station, the CSI is recovered by use of several convolutional layers and identity shortcut connections. Experiments show that the proposed method performs well at low compression ratios and reduced time complexity. In [23], a long short-term memory (LSTM) architecture admits the autoencoder to benefit from exploiting temporal and frequency correlations of wireless channels.

The method proposed in [22] is designed for a single compression rate. In [24], a multiple-rate compressive sensing framework is proposed. The proposed method does not only improve the reconstruction accuracy but also bridges the gap between DL-based methods and practical deployment by reducing the needed storage space in the UE and providing multiple compression rates. Two different variable rate frameworks are proposed, which are based on the work of [22]. The proposed frameworks reduces the reconstruction errors and the parameter number is reduced by 38.0% and 46.7%, respectively, thereby greatly saving the storage space at the UE. The work in [24] is the first to address the variable compression rate issue in DL-based CSI feedback.

Compression of CSI in the MIMO application is also addressed in the work of [25]. The proposed CSI compression network consist of a deep CNN in conjunction with quantization and entropy coding blocks. The architecture is based solely on CNN layers and has no FCNN layers. This means that the approach in [25] is flexible and can be

used for different numbers of channels and antennas, while methods with fully connected layers as in [22] needs to be trained for each possible setting of numbers of channels and antennas. The results in [25] shows negligible performance degradation when the number of channels and antennas are different from the numbers used in the training.

Channel estimation is challenging for multicell massive MIMO cellular networks. This is a fundamental difficulty due to pilot contamination, which is the interference of pilot symbols utilized by the users in neighboring cells. In [26], a low complexity massive MIMO channel estimation technique that is robust to pilot contamination is proposed. The proposed channel estimator is composed of two stages. In the first stage, a less noisy signal is generated from the received signal through a specially designed FCNN architecture. The used FCNN is a modified deep image prior (DIP) network.

DIP is originally a technique that uses CNN to enhance images without any prior training of the network [27]. The DIP network is randomly initialized and used to solve inverse problems like noise reduction, inpainting and super resolution. The DIP design of the channel estimator of [26] eliminates the need to map labels to inputs as in supervised learning. DIP has observed efficient denoising and inpainting performance thanks to the specifically designed FCNN architecture, which has low impedance for natural images and high impedance against noise. That is, the model fits more to the structured signal and less to the unstructured noise [27].

In the second stage of channel estimation, the generated or filtered signal is multiplied by the Hermitian transpose of the pilot sequence for channel estimation [26]. Effectively, an LS-type of channel estimator is proposed, with the only difference being that the signal generated by the FCNN is used instead of the received signal. By doing so, the low complexity nature of the LS estimator is combined with the noise reduction capability of the FCNN so as to have a near MMSE estimation performance. The price paid for the proposed deep channel estimator is the need for fitting the parameters of the FCNN periodically for each OFDM grid, whose period is determined by the channel coherence time (or equivalently maximum Doppler spread).

All of the reviewed research work in the application of channel estimation, equalization and modeling have used synthetically generated data as input in their training and evaluation of their ML algorithms. For channel estimation, complex baseband signals are used as input to the algorithms, while estimated channel matrix is used for CSI compression and recovery. A summary of all reviewed literature in this section is found in Table A.2.

3.3 Channel Coding

Channel coding refers to the process of detecting and correcting bit errors in digital communication systems. Channel coding enables the receiver to amend errors occurring during transmission due to, for example, noise, interference and fading. Coding theory has been around since the late 1940's and optimization of coding algorithms has come a long way. It is hypothesized that neural network-based technology could offer a more powerful channel coding solution than conventional approaches in some aspects, including coding performance, power consumption, and processing latency, but mainly NNs aim to reduce the computational complexity of coding algorithms. However, a few obstacles are left to overcome before the technology reaches that point.

ML applications for encoding and decoding of error control codes suffer from the *curse of dimensionality* [28], where successful training of a neural network is inhibited by the large number of code words required to correctly create the necessary decision regions. A short code of, for example, length $N = 100$ and coding rate $r = 1/2$ allows for the existence of 2^{50} different code words, far too many to completely train any neural network. Successful design of the network would require a training set of equal extent, resulting in impractically large training time, storage space and number of neurons. A possible work-around for this problem is to consider a neural network with the ability to generalize and predict code words, i.e. learning a decoding algorithm which can infer the entirety of the code by training on a small fraction of code words. Such generalization of code words requires that the code itself possesses some kind of structure, such as convolutional or algebraic codes.

Trials of using deep learning for decoding of structured codes are abundant, (cf. [29]–[32]), with the main purpose of trying to match established, state-of-the-art techniques. In [30] a simple neural network is designed to decode polar coded short packets over a frequency-flat fading channel, motivating the choice of low code rates with applications in smart networks and messaging services requiring low latency. The paper finds that the neural network achieves the coding gain by increasing the learning epochs, but does not match the theoretical gain.

In [31], a different ML algorithm is implemented to help design code words as opposed to encoding or decoding. The network, an reinforcement learning (RL)-based approach, attempts to learn a polar code construction framework in order to reach performance comparable to classic coding approaches. The design framework consists of a code constructor and a code evaluator adapted for a suitable environment (channel conditions, decoding algorithm, etc.). The constructor learns a series of valid code structures by iteration using feedback from the evaluator, until the performance metric converges. The network is able to iteratively refine code construction without being taught explicit knowledge of coding theory. The performance does not achieve that of classic techniques, but the benefit of removing the necessity of coding knowledge for optimal coding is promising for future work.

Another RL-based approach is implemented in [32], in the context of factor-graph permutations for polar decoding, a type of graphical interpretation of the code. This can be formulated as a multi-armed bandit problem for RL, where an agent has to repeatedly make a choice among a number of different actions. After each action, the agent receives a reward drawn from a distribution that depends on the selected action. By letting the permutated factor graphs represent the possible actions, the decoding algorithm decides which action maximizes the reward during the course of decoding. The paper finds a performance gain of around 0.125 dB in comparison to similar decoding approaches, and shows no additional latency overhead for the method.

In [29] the purpose is to investigate if a NN is able to decode code words that it has never seen during training and to determine if structured codes are easier to learn than random codes. The paper evaluates the effect of typical design parameters, such as the number of training epochs, the learning and validation SNR, the effectiveness of two different loss functions (MSE and binary cross entropy (BCE)), as well as network size and number of neurons. The paper aims to optimize these parameters in order to generalize random channel coding and structured codes in the form of polar codes.

The results in [29] indicate that the neural network decoder (NND) is able to achieve the generalization for polar codes, but not for random codes. The network does not achieve maximum a posteriori (MAP)-performance without training on the entire set, but an NND that is able to generalize is interesting for future applications, as the implication is that it is possible to learn complete decoding algorithms. State-of-the-art decoding of polar codes suffers from high complexity, lack of possible parallelization and thus, critical coding latency. An NND inherently describes a highly parallelizable structure and emerges as a promising alternative channel decoding approach as it avoids sequential algorithms.

In [33], a joint source-channel coding RNN is proposed for a channel-encoder-decoder system. The network is trained to recognize words and to preserve semantic information of sentences. That is, generate a text on the transmission side, and receive and recognize a similar but not an exact copy that constitutes the same information, for example ”the car is coming” and ”the automobile is arriving”. The encoder is set up as a stacked bidirectional LSTM (BLSTM) network with a binarization step. The channel is designed as an erasure channel which determines if a packet is dropped or not with a tunable probability, described by a dropout layer in the RNN. The decoder is designed by a stack of LSTM's where the observed vector from the channel is used as input.

The paper utilizes a dataset containing transcriptions of proceedings of the European Parliament. Since the ML-algorithm is supposed to learn to distinguish semantic information, the dataset is pre-processed in the sense that the most common words from the set are extracted and assumed to be known by the network. Next, sentences of 4-30 words are selected where less than 20% of the words are unknown to the network. As the algorithm is supposed to identify similar sentences, the known words provide context where each estimated word in a sentence function as a key in order to form the most likely sequence of words, known and unknown, creating a sentence.

A disadvantage with the implementation in [33], as noted by the authors, is that a fixed number of bits is used for encoding of all sentences. The schemes inherently produce 'embeddings' (representations of the words) of different lengths. Encoding of a sentence might exceed the allocated bit budget, where upon any remaining words are excluded, resulting in word errors.

The NN is shown to perform well when fewer bits per sentence than necessary are allocated but loses its advantage when the bit allocation exceeds the number of bits required, whereas the compared standard Reed-Solomon codes compensate for all channel erasures. The paper [33] also investigates a varying drop rate and finds that the NND performs well for larger drop rates, when fewer bits per sentence than necessary are used to encode. A varying sentence length is also examined. When the sentences reach a certain length, the allocated bits per sentence are no longer enough to code the entire text. At that point the NN outperforms the standard methods due to a constant bit allocation for every sentence and can be improved by varying the embedding length based on the sentence length. The authors concludes that their jointly-designed NND outperforms separately coded source- and channel methods, particularly when fewer bits are allocated to encode each sentence.

An increasingly common decoding approach is to combine standard belief propagation (BP) decoding with a neural network. In [34], a neural belief propagation (NBP) decod-

ing approach is proposed. The idea is to consider NBP decoding over an over-complete parity check matrix and use the weights of the NBP as a measure of the importance of the check nodes associated with the NBP. Check nodes deemed unimportant are pruned. The results compared to a regular NBP (without pruning) are increased by 0.27-0.31 dB while reducing the number of check nodes evaluations by 97 %. The method also outperforms regular low-density parity-check (LDPC) decoding by 0.52 dB.

In [35], an iterative BP-CNN architecture for channel decoding is proposed. The paper suggests a concatenation of a BP decoder with a trained CNN under correlated noise. The BP decoder estimates the coded bits as it normally would, and the CNN is trained to remove the estimation errors arising from the BP decoder. Iterating between the BP and CNN gradually improves the performance. The BP-CNN decoder is found to be robust to SNR and correlation mismatches. For various levels of noise correlation, the proposed decoder manages similar or better performance than the equivalent BP-decoder. This approach is a pure feed-forward network, without memory or state information. The authors point out that another angle of this work would be to substitute the CNN with an RNN, as the recurrent network is effective in taking advantage of the sequential information and may exploit the correlation of the noise sequence.

All told, machine learning approaches for channel encoding and decoding does not yet reach the raw performance of traditional coding methods, having been near-optimized since the birth of information theory. Nevertheless, research continues to advance as the possible complexity gains of substituting classical encoding-decoding systems with a suitable neural network provide great incentive. The literature on channel coding is summarized in Table A.3.

3.4 Resource Allocation

Resource allocation problems, such as power control, channel selection as well as network resource allocation and routing, often leads to complex optimization problems that are computationally burdensome to solve. Therefore, many resource allocation problems can be solved using ML algorithms. This is stated as follows in an overview paper on DL-based wireless resource allocation:

”The traditional wisdom is to explicitly formulate resource allocation as an optimization problem and then exploit mathematical programming to solve the problem to a certain level of optimality. Nonetheless, as wireless networks become increasingly diverse and complex, [. . .] the current design methodologies face significant challenges and thus call for rethinking of the traditional design philosophy. Meanwhile, deep learning, with many success stories in various disciplines, represents a promising alternative due to its remarkable power to leverage data for problem solving.” [36]

Several examples of using ML for resource allocation in wireless communications are outlined in the following sections. The literature about resource allocation is summarized in Table A.4. In summary, much research about resource allocation use ML to either approximate existing optimization algorithms or to learn an approximately optimal solution. The aim is usually to find near-optimal solutions to optimization problems that are too computationally burdensome to solve with existing methods, as will be exemplified in the following sections.

3.4.1 Power Control

A learning approach is proposed in [37] to perform max-min and max-prod power allocation in the downlink of massive MIMO networks. A deep neural network is trained to learn the map between the positions of UEs and the optimal power allocation policy, and then use the map to predict the power allocation profile for new sets of user positions. The use of deep learning improves the complexity-performance trade-off of power allocation, compared to traditional optimization-oriented methods.

Optimization of the sum spectral efficiency in multi-cell massive MIMO systems with a varying number of active users is considered in [38]. This is formulated as a joint pilot and data power control problem. Since the problem is non-convex, an iterative algorithm is first designed that obtains a stationary point in polynomial time. To enable real-time implementation, a deep learning solution is also developed. The proposed neural network uses the large-scale fading information to predict both the pilot and data powers. A single neural network that can handle a dynamically varying number of active users is designed, and as such is simultaneously approximating many different power control functions with varying number of inputs and outputs.

Optimization algorithms often entail significant complexity, which creates a gap between theoretical analysis and real-time processing. The work of [39] provides a learning-based approach to approximate optimization algorithms, to address this issue. The key idea is to treat the input and output of an algorithm as an unknown nonlinear mapping and use a neural network to approximate it. If the nonlinear mapping can be learned with desired accuracy by a neural network of moderate size, the algorithm can be performed effectively since the neural network approximation only requires a small number of simple operations. In [39], a class of optimization algorithms is identified that can be accurately approximated by a fully connected neural network. The proposed approach is exemplified by approximating an interference management algorithm, which can be cast as a power control problem over an interference channel.

A framework for energy-efficient power control in wireless networks is developed in [40]. The proposed method is a branch-and-bound procedure based on problem-specific bounds for energy-efficiency maximization. This enables to find the global solution for the most common energy-efficient power control problems with reduced complexity, and allows its practical implementation through the use of deep neural networks. That is done by the development of a feed-forward neural network. The reduced complexity of the proposed branch-and-bound based algorithm enables offline generation of a large training data set, containing optimal power allocations. The data set is then used to train the neural network to learn the optimal map between the network channel realization and the corresponding optimal power control policy. By contrast, many other power control methods based on deep learning train the neural network based on suboptimal power allocations, due to the large complexity of generating large training sets with available global optimization methods.

A mathematical description of using neural networks for resource allocation optimization is shown in [41]. The paper shows examples of using neural network for power allocation strategies that aim to optimize the aggregated capacity in simple AWGN fading channels and in interference channels, with channel states as inputs.

The paper [42] considers optimal resource allocation across a set of transmitters and receivers in a cellular or ad-hoc wireless network. CNNs are used for parameterization, as their dimensionality is small and does not scale with network size, in contrast to FCNN. The paper use a random edge GNN, which performs convolutions over random graphs. The random graphs are formed by the fading interference patterns in the wireless network. Numerical examples are provided for power allocation in different types of networks.

In [43], a deep RL framework is proposed to provide model-free resource allocation in the downlink of a cellular wireless network. The goal of the proposed algorithm is to guarantee high end-to-end reliability and low end-to-end latency, under explicit data rate constraints, for each wireless user without any models of or assumptions on the users' traffic. The algorithm is based on GAN, which is used to pre-train the algorithm with a mix of real and synthetic data. The proposed algorithm is applied to a multi-user orthogonal frequency-division multiple access (OFDMA) resource allocation problem, posed as a power minimization problem under reliability, latency, and rate constraints.

In [44], power control in a multicast scheme for a wireless downlink is considered. Obtaining optimal power control is intractable because of a very large state space. Therefore, deep RL is applied by function approximation of the Q-function via a deep NN.

The paper [45] propose an algorithm based on a DL approach to perform sum-rate-max and max-min power allocation in the uplink of a cell-free massive MIMO network, with single antenna user terminals. A deep neural network is trained to learn the mapping between a set of input data and the optimal power allocation strategy. It is shown numerically that the learning based algorithm performs near optimally in some cases, but may be significantly degraded in other cases.

3.4.2 Channel Selection and Assignment

The paper [46] proposes a deep RL-based MAC protocol for the uplink of a heterogeneous wireless network. More specifically, the paper considers a number of heterogeneous networks, using different MAC protocols, trying to access the time slots of a common wireless channel. The proposed protocol is assumed to have no knowledge of how the other networks make decisions on when to transmit and when not to. The goal of the protocol is to be learn an optimal channel access strategy to achieve a certain global objective, e.g. maximizing the sum throughput.

The work of [47] deals with spectrum prediction collision avoidance and proposes an algorithm that can predict the behavior of other surrounding networks by using supervised deep learning. The proposed algorithm is constructed from a CNN that predicts the spectrum usage of the other neighboring networks. Power is measured to determine spectrum usage which is used as input to the CNN to predict usage of the channel in the next time slot.

3.4.3 Joint Resource Allocation

A deep reinforcement learning based joint channel assignment and power allocation scheme of a multi-carrier non-orthogonal multiple access (NOMA) system is proposed in [48]. The joint channel assignment and power allocation problem is first formulated

as an optimization problem with max sum-rate and max min-rate objectives. To resolve the optimization problem, a closed-form solution to the power allocation problem given channel assignment is first derived. Then a deep reinforcement learning framework, which utilizes an attention-based neural network, is proposed to address the channel assignment problem. The attention-based neural network exploits an encoder-decoder structure.

A distributed resource sharing scheme, based on multi-agent RL, where multiple vehicle-to-vehicle (V2V) links reuse the frequency spectrum of vehicle-to-infrastructure (V2I) links is developed in [49]. The resource sharing problem is solved using a fingerprint-based deep Q-network method that is compliant with a distributed implementation. The V2V links receive distinctive observations, but a common reward, and learn to improve spectrum and power allocation through updating Q-networks. The proposed method is divided into a centralized training stage and a distributed implementation stage, and is shown to improve system level performance although decision making is performed locally at each V2V transmitter.

Joint optimization of beamforming, power control, and interference coordination in a 5G wireless network, to enhance the communication performance, is considered in [50]. The problem is formulated as a non-convex optimization problem to maximize the signal-to-interference-plus-noise ratio (SINR). The problem does not have a closed-form solution, but requires exhaustive search. The problem is then solved using deep RL (deep Q-learning) to estimate future rewards of actions, using the reported coordinates and SINR of the user terminals.

Resource allocation in an intelligent reflecting surface (IRS)-aided multiple input single output (MISO) NOMA network is considered in [51]. An algorithm is proposed that aims to maximize the sum rate of all users by jointly optimizing the passive beamforming vector at the IRS, decoding order and power allocation, subject to the rate requirements of users. The proposed algorithm is based on three machine learning algorithms that are used to predict user mobility, partition users into clusters and design the phase shift matrix, respectively. More specifically, an LSTM-based algorithm is first adopted for predicting the mobility of users, a K-means based Gaussian mixture model algorithm is proposed for user clustering, and a deep Q-network based algorithm is invoked for jointly determining the phase shift matrix and power allocation policy.

3.4.4 Network Resource Allocation and Routing

Network modeling is critical for optimization of routing. One of the fundamental characteristics of optimization is that we can only optimize what we can model. There are complex relationships between topology, routing and input traffic. A tutorial overview of DL for wireless mobile network applications is presented in [54]. The following specific domains are identified, among others, where deep learning has made advances in wireless networking.

- Network-level mobile data analysis, which focuses on deep learning applications built on mobile big data collected within the network, including network prediction and traffic classification.
- App-level mobile data analysis, which deals with mobile data analytics on edge devices.

- User mobility analysis to understand the movement patterns of mobile users, both at group or individual levels.
- Network control, which includes the usage of DL on network optimization, routing, scheduling, resource allocation and radio control.
- Network security, which is divided into infrastructure, software, and privacy related security.
- Signal processing on the physical layer that benefit from deep learning.

A review of RL-based routing in networks is presented in [55]. One major concern in routing is the optimization of routes while considering dynamic topology changes. Many modern networks, for example vehicular and other ad hoc networks, are dynamic in nature which results in topology changes. RL can be an efficient alternative to address network conditions as they appear in real world, and the work of [55] provides a presentation of the main characteristics of RL-based routing protocols.

In [52], GNNs are used to model a network. The GNN approach show generalization capabilities in other topologies and routing schemes not seen during training. The GNN network is trained using simulated data using the open-source simulator software Omnet++[1]. The dataset contains pairwise source-destination traffic matrices in different topologies, routing schemes, and traffic patterns. The model takes as input topology, routing scheme, and traffic matrix (bandwidth between each pair of nodes in the network). Output is performance metrics of delays and jitter according to the current state. Following on their previous work, in [53] the optimization is done with a (RL) method instead of Markov decision process.

3.5 End-to-End learning

End-to-end (E2E) learning of communication systems is a field of study whose goal is to learn full transmitter and receiver path characteristics which are optimized for a specific performance and channel metric. E2E learning refers, in this context, to training a possibly complex system represented by a single DL model that represents the complete target system, bypassing the intermediate layers usually present in traditional pipeline designs. The idea of using a single model that can specialize to predict the outputs directly from the inputs allows, potentially, the development of otherwise extremely complex systems that can be considered state-of-the-art.

One of the more prominent designs of end-to-end DL of communication systems is describing the system as encoding and decoding functions with a layer in between that represents the channel. In this configuration, the system can be interpreted as an autoencoder [57]–[59], which does not compress but adds redundancy to increase reliability. The concept of the autoencoder provides a powerful method for performing unsupervised learning with strengths in dimensionality reduction and information retrieval, facilitating many possible applications.

In [57], the concept of the channel autoencoder is investigated in the form of deep NNs and CNNs. The channel autoencoder includes an encoder, a channel regularizer, and a decoder. Synthetic channel impairments such as additive Gaussian noise, unknown time

[1] https://omnetpp.org/

and rate of arrival, frequency and phase offset and delay spread are included as separate layers in the autoencoder to model different channel behavior. The autoencoder is evaluated by comparing different loss functions, namely the MSE and the BCE-function. A deep NN and a CNN are employed as network structures. The paper finds that the MSE provides the best performance of the loss functions, and coupled with the deep NN, slightly outperforms the CNN-setup. The conclusion of the paper is that the autoencoder architecture is viable for end-to-end design and implementation, maintaining generality and providing low complexity.

However, a few issues remain. One of the major drawbacks of current learning approaches, including the autoencoder approach, is that a sufficiently accurate channel model is needed for training of the underlying neural networks. This model also needs to be differentiable to enable back-propagation through the whole system to learn the optimal weights of the deep NN [59]. In real-world scenarios, such a channel model is hardly available and often many aspects of the channel response, e.g. hardware effects and noise distribution, are not even known at all. Some works, therefore, focus on a generative approach [60], or rely on reinforcement learning [61] to circumvent this problem.

In [60], a conditional GAN is used to represent the channel effects, where the encoded signal of the transmitter serves as the conditioning information. The purpose of the GAN is to develop a channel-agnostic E2E learning-based system, where different types of channel effects can be automatically learned without knowing the specific channel transfer function. The transmitter, receiver and channel generator are trained iteratively. The channel generator is trained alongside the discriminator, where the latter attempts to determine whether the input belongs to a real dataset or if it is fabricated by the generator, which in turns tries to maximize the output of the discriminator. By iteratively training these networks, it is found that the end-to-end loss can be optimized for a network of GANs without prior information of the channel, and obtains a similar performance as traditional methods basing the foundation on perfect knowledge of channel models.

In [61], RL is proposed as a way to circumvent the back-propagation problem by using a feedback link. In this approach, the transmitter can be interpreted as an agent which performs actions in an environment and receives a reward by way of the feedback link. The transmitter can be designed to minimize an arbitrary loss function connected to the actions. The algorithm in [61] iterates between two phases: supervised training of the receiver and RL-based training of the transmitter based on an estimated gradient of the loss.

After taking an action, the transmitter receives a per-example loss which it tries to minimize in the next action by finding a suitable policy. This alternating training algorithm shifts between receiver and transmitter, thus it first improves the receiver for fixed transmitter parameters, then improves the transmitter for fixed receiver parameters. This RL-based network is found to achieve a similar performance as a corresponding fully supervised approach, with the benefit of not requiring a mathematical model of the channel.

In [58], end-to-end learning for over-the-air (OTA) transmissions is demonstrated for communication over an actual wireless channel. Continuous OTA transmissions are subjected to several potential problems that can severely degrade performance, such

as inter-symbol and inter-message interference, additional random phase and carrier frequency offset, frame synchronization, Doppler and multipath effects. The system is described as an autoencoder, with trainable transmitter, channel and receiver parts. The autoencoder might be able to compensate for a few of these impairments in some capacity due to the choice of stochastic channel model, but more significant issues can be handled by implementing additional FCNN as extensions to the autoencoder. The final end-to-end system in [58] consists of an autoencoder with five different deep NNs, designed to cover specific channel effects.

The system is evaluated over simulated and real channels, and compared to a baseline system consisting of a differential quadrature phase shift keying (DQPSK) transceiver. The performance of the autoencoder is about 1 dB worse than the DQPSK baseline over the entire SNR-range. For OTA transmissions, the performance gap is about 2 dB as the imperfections are more apparent in real systems.

To summarize this chapter, end-to-end applications are implemented in order to solve a specific challenge or improve performance as a whole of communication systems. In the papers mentioned above, reduced complexity achieved in the system usually emerges as the main contribution. Comparisons are mainly towards equivalent, separate-module ML-designs and not specifically traditional non-ML techniques. Any end-to-end approach require suitable, non-trivial models for all modules of the network in order to accurately portray a real system, where either great quantities of data, perfect capture of channel- or hardware effects and proper optimization algorithms are essential in order to achieve comparable performance results vis-a-vis state-of-the-art techniques. Papers discussed in this section are summarized in Table A.5.

3.6 Spectrum Sensing

Spectrum sensing is a key feature of cognitive radio and usually refers to the ability to detect the presence of or classify a user in a licensed spectrum [62]. If no user is present, the spectrum can be used by an unlicensed user, and the spectrum is used more efficiently. Signal classification can be expressed as an M-ary hypothesis test as follows:

$$\begin{aligned} &H_0 : \ y(n) = w(n), \\ &H_1 : \ y(n) = h_1 s_1(n) + w(n), \\ &\vdots \\ &H_M : \ y(n) = h_M s_M(n) + w(n), \end{aligned} \tag{3.1}$$

where $y(n)$ is the received signal, $s_m(n)$ is the transmitted signal, $w(n)$ is noise, and h_m represents the channel. Making spectrum sensing a classification problem invites the use of machine learning. Conventional model-based signal detection methods often need an underlying model to be defined, for example, a model of the signal-noise probability, the background noise, or the activity pattern of the users. Model-based methods are limited by the accuracy of the assumed model. Machine learning has the advantage of being a data-driven method where the data pattern is learned from a training process. In this way, the common problem of making the wrong assumptions when creating a model can be avoided.

3.6.1 Signal Detection

Spectrum sensing, in the interpretation of detecting the presence of a signal, can be expressed as a binary hypothesis test as follows:

$$\begin{aligned} H_0 &: \ y(n) = w(n), \\ H_1 &: \ y(n) = hs(n) + w(n). \end{aligned} \tag{3.2}$$

The problem shown in (3.2) can be viewed as a special case of the classification problem (3.1) with only two categories: no signal present (H_0) and signal present (H_1).

The papers [63] and [64], both present DL-based methods to solve the signal detection problem. The authors of [63] propose a method using two parallel CNNs, one CNN to model the current activity and one CNN to model the historical activity. The output from the two parallel CNNs is at the end connected and flattened into two elements, one element for hypothesis H_0 and the other for hypothesis H_1. The paper assumes a frame structure and that the unlicensed user has adopted the slotted scheme. The input to the first CNN is the sample covariance matrix (SCM) of sensing data at the current frame. SCM is chosen as input because of its ability to show both energy and correlation. Input to the second CNN is a large historical SCM created by adding multiple historical SCMs together. The SCMs are treated as pictures, and the CNNs' well-known ability to learn the spatial features from images can be used. Simulation results show that the proposed method performs better than conventional hidden Markov model-based detection and estimator-correlator detection.

The authors of [64], propose a convolutional LSTM network for signal detection. While [63] use historical SCMs to learn the sequential pattern, in [64] the LSTM is used for this purpose. The paper [64] is inspired by modulation recognition, mainly the work of [65], and wants the network to learn the structure of a modulated signal to use for signal detection. The dataset used in the paper is also from [65]. Input to the network in [64] is the complex baseband signal. The number of input samples tested is 64, 128, 256, 512 and 1024, and results show that the more samples, the better performance. The NN of [64] is built on two CNN layers, two LSTM layers, and three FCNN layers. In the paper [64], the network is compared to energy detection, a simple and popular sensing method that detects user activity based on the energy of the received signal. The proposed network and the energy detector were tested for different SNR values, showing that the proposed network is 5 dB better than the energy detector regardless of modulation type. The detection performance was evaluated for a network trained on one modulation scheme and tested on signals with different modulation schemes. The results show some generalization capability if the modulation type is of the same type, for example, 16-QAM and 64-QAM, or BPSK and quadrature phase shift keying (QPSK).

3.6.2 Signal Classification

Signal classification with autonomous accurate labeling of the signal type is a key enabling factor for spectrum interference monitoring, signal intelligence, dynamic spectrum access etc. Traditionally this is done with hand-crafted feature extraction and by deriving decision bounds for different signal types.

The idea of using ML and especially CNN and FCNN for modulation classification is introduced in [65]. In that work, a CNN is trained and evaluated with complex

baseband signals that are synthetically generated for 11 modulation types, eight digital and three analog. For benchmark comparison, 32 expert features of the signals are used as input to various classic classifiers such as nearest neighbor, Gaussian Naive Bayes, and an RBF[2]-SVM[3]. Additionally, a 3-layer deep neural network is trained with the expert features. The obtained classification accuracy results are promising. The overall classification accuracy of the CNN is 87.4 %. The classical feature based reference methods are outperformed by 2.5-5 dB of SNR for low SNR, while the classification accuracy is similar above 5 dB SNR.

In [66] a more extensive dataset, with 35 modulation types (analog and digital) both synthetically generated and received over-the-air, is used. The dataset is divided into a normal and a difficult part. In the paper, two different types of networks, CNN and residual neural network (ResNet), are compared with a baseline method. The baseline method used is an XGBoost expert feature classifier. The CNN and the ResNet are trained and evaluated with a series of 1024 complex baseband samples of the signal. In the authors' previous work [65], 128 samples were used. For the difficult part of the dataset and an AWGN channel, the best classification accuracy of 99.8 % is achieved with the ResNet, while the CNN achieves 98.3 % and the baseline method 94.6 %. When different channel impairments are introduced, the classification performance decreases. However, with unsynchronized local oscillator (LO)s and Doppler frequency offset there is nearly a 6 dB performance advantage of the ResNet approach compared with the baseline, and a 20 % accuracy increase at high SNR. For the OTA dataset, 80 % of the dataset is used for training and the remaining part used for evaluation. The SNR is approximately 10 dB and results show a classification accuracy of 95.6 % for the ResNet.

Transferring a model trained with synthetic data to be used with OTA data can be useful in some cases. For example where OTA training data is a scarce resource but synthetic data can be generated. In [66] this concept is evaluated. By evaluating the performance on OTA data without any changes to the trained model, the performance decreases from 96 % to 80 %. By fine tuning the trained model with some OTA training data, approximately 10 % additional accuracy can be recovered.

Complex baseband signals are used as input data in the different ML networks in both papers by O'Shea et al. [65], [66]. However, in [67] a slightly different approach is taken, such that the amplitude and phase (polar coordinates) are used instead of rectangular coordinates which are commonly used for complex baseband signals. Rajendran [67] use oversampled versions of the dataset used in [65] and compares different LSTM models with the CNN used in [65]. The LSTM model gave poor results when fed with complex baseband samples in rectangular coordinates while it gave accuracies close to 90 % for high SNR when provided with amplitude and phase data. The CNN model, however, did not achieve better performance with amplitude and phase representation compared to rectangular I/Q representation. This result indicates that the choice of input data may affect how well a NN can fit the training data. Although the LSTM models perform very well at high SNR conditions, CNN models seem to provide an additional 5-10 % accuracy on the low SNR conditions (SNRs below -2 dB)

After O'Sheas two papers [65], [66] in 2016 and 2018 respectively, the number of papers

[2]Radial Basis Function
[3]Support Vector Machine

have increased on the application of modulation and signal classification with the use of ML.

Training a model can take a substantial time depending on the type of model that is going to be trained, the size of the training dataset and the used hardware. In [68] algorithms to reduce the training time by minimizing the size of the training dataset, while incurring a minimal loss in classification accuracy, is studied. The authors of [68], demonstrates the performance of Principal Component Analysis (PCA) [4] in significantly reducing the training time, while maintaining good classification performance at low SNR. Training with only two SNR values, one at high and one at low SNR, leads to high classification accuracy over a wide portion of that SNR range and thus reducing training time.

Machine learning can be used for automated monitoring of the electromagnetic spectrum over frequency, time and space. In [69] an adversarial autoencoder architecture is trained to detect spectrum anomalies from power spectrum density data. The model achieves an average anomaly detection accuracy above 80 % at a constant false alarm rate of 1 % along with anomaly localization (in time and frequency) in an unsupervised setting. Finally, the model is tested on data from one of the distributed ElectroSense[4] sensors over a long term of 500 hours showing its anomaly detection capabilities.

In [70], four types of neural networks are trained to classify different transmissions that are commonly used in high frequency (HF) communication. The synthetically generated training data consist of 18 different transmission modes, including analog and digital modulation schemes, with various baud rates for the digital schemes. The detection performance is evaluated for the SNR range from -10 to 25 dB. Best performance is shown by the deep CNN and the residual net with an accuracy of approximately 94 $\%$.

3.7 Positioning and Localization

Determining the location of an emitter has many practical uses. In cellular networks, knowing the location of a mobile phone can be used to improve communication performance, but also enables services such as location of mobile phones during emergency calls. Identifying and locating ad-hoc network emitters can be useful for optimizing use of the communication spectrum.

The definition used in this report for positioning is to produce data that can be used to estimate coordinates, while localization is to reach a conclusion of the place of the emitter on a map. These two terms are however often used with very similar meaning in the literature.

This section is divided into two parts to divide the two cases defined by the prerequisite of requiring or having an infrastructure in place (such as cellular networks) or not (such as ad-hoc networks). The first part is covered only briefly since it is closely related to the off-topic area of surveillance.

[4]The ElectroSense network is a crowd-sourcing initiative to collect and analyze spectrum data. https://electrosense.org/

3.7.1 Localization without infrastructure

The expression *without infrastructure* refers to systems not connected to the core network of a communication network. An example of a device not connected to the communication infrastructure is a stand-alone receiver used to monitor the radio spectrum.

One family of localization methods is direction-of-arrival (DoA) estimation. There are several methods to estimate direction-of-arrival. A common method is phase interferometry, and requires an antenna array. This method assumes that there is only one signal, and to separate and locate several signals simultaneously an algorithm like multiple signal classification (MUSIC) can be used [71].

The MUSIC algorithm estimates the noise subspace and projects a set of angles on the noise subspace. Any direction with a signal will be perpendicular to the noise subspace and thus give a value close to zero. Although MUSIC generally has high performance, the computational complexity of searching the parameter space (searching over angles) is high.

Another problem with high-precision direction-of-arrival methods like MUSIC is array imperfections. In [72], a deep neural network is trained to correct array imperfections and locate two emitters. Performance is improved compared to using MUSIC without correcting for array imperfections. The solution network consists of two sub-networks. First an autoencoder is used to correct for the array imperfections (acting like a group of spatial filters). Then a set of classifier networks are used to find emitters in a smaller sector.

In [73] the method is improved by using CNN and an evaluation of computational performance is done. The method shows high performance and low computational complexity.

3.7.2 Localization with infrastructure

Communication infrastructure refers to techniques and systems that are needed to support communication networks such as the Internet, cellular networks, and local area networks. Examples of communication infrastructure are base stations for cellular networks, and optical fiber cables connecting users to the Internet.

Fingerprint positioning

Fingerprint positioning is based on measuring signal characteristics at different positions and storing the information in a database [74]. The method consists of two phases, the offline training phase and the online positioning phase. In the offline training phase, a device transmits packets from known positions to a base station or access point and the base station or access point stores the received signal characteristics as the fingerprint of that position. In the online phase, the position of an unknown device is then estimated by comparing the device's signal characteristics to the fingerprints in the database and giving the device the closest position. The signal characteristics used as fingerprints vary depending on the application, two commonly used are receive signal strength (RSS) and CSI. Fingerprint-based methods are often seen as a solution for positioning in indoor, urban, or other complex environments where non-line-of-sight (NLOS) scenarios usually occur. In contrast to traditional positioning algorithms, in fingerprint positioning, mul-

tipath propagation does not aggravate the positioning. Instead, multipath propagation can contribute to making the fingerprints more distinguishable. However, this advantage only applies as long as the environment is static.

Fingerprint methods suffer from high complexity. In the worst case, the characteristics of an unknown device must be compared with each fingerprint in the database. Methods to lower the complexity of finding a corresponding fingerprint are, therefore, a significant focus. Especially with massive MIMO's emergence, where measurements from multiple antennas are available, more measurements contribute to higher complexity. The paper [74] highlights the advantage of using CNN for fingerprint positioning. The most computationally heavy aspect of a CNN is the training process. This is due to the large size of training data. Once the training is finalized, positioning can easily be achieved due to the feed-forward structure of the network. One position estimate has a complexity order proportional to the product of the number of Kernels, the number of CNN-layers, the number of antennas at the base station, the number of channels and the kernel sizes. The fact that the estimation complexity does not depend on the training data size is an advantage of using CNN for positioning.

The paper [74] also presents a method for fingerprint positioning. The method uses CSI values as input data. The CSI values are transformed into channel snapshots. The channel snapshot can be represented as a two-dimensional image with delay on one axis and direction of arrival on one axis, often called delay angular spectrum. The images are generated using the channel model COST 2100, a geometry-based stochastic channel model. In COST 2100, multipath propagation is represented by clusters of multipath components (MPC). In the delay angular spectrum, these multipath clusters can be distinguishable. In the paper, the CNN is trained to output a user's location given a channel snapshot. The training positions are spaced at $\lambda/2$. Multiple kernels are used to capture both clusters of many MPC and few MPC. The method presented in the paper can achieve an accuracy of a fraction of a wavelength if a representative enough data set is available for training.

In [75], an autoencoder is used to reduce the complexity of the fingerprint positioning. The setup used in the paper is a signal of 30 carriers and three receive antennas for diversity. Input to the network is CSI amplitude responses (not phase), giving 90 CSI for each fingerprint location. The network lowers the dimension of the input data to 50. After the network is trained, the weights in the network are stored as fingerprints to facilitate localization in the online test. The proposed method was tested in a living room environment and a computer laboratory environment. In the test environment, multiple mobile users were present. The experimental results show that the network performs well in the tested environment. An average localization error of 0.9 meters is obtained. The group of authors has published multiple works extending their methods.

The main disadvantage of fingerprint positioning is the need for extensive measurement campaigns to create the fingerprint database. The fingerprint database is accurate for as long as the environment stays unchanged. If, for example, a piece of furniture is removed or a new building built, the fingerprint may change and the measurement campaign has to be redone. An alternative to bypassing the measurement campaigns is using simulation methods to generate the fingerprints database. Such simulations rely on a good approximation of the signal propagation phenomenon and a good geometrical environment description. In [76], the fingerprint database is created by using ray-tracing

together with 5G system simulation. For ray-tracing, the software WinProp was used, which gave a resolution of 1 m. The system simulated was the 5G New Radio air interface. The region considered in the article is Lincoln Park in Chicago, which have eight serving cells. For each location, a reference signal received power (RSRP) was simulated and used as training data to a FCNN. In the paper, different variations of FCNNs were evaluated on both network-level and cell-specific. Network-level refers to when the FCNN is trained on data from all eight cells together. The feature vector then includes both the RSRP value and the cell-ID. The authors of [76] claim that cell-specific training performs better than network-level training but with the disclaimer that the difference can be due to a lack of training data. Cell-specific FCNN of two layers provided a mean positioning error of 1.4 m. The paper [76] emphasizes that the use of ray-tracing and system simulators serves as a helpful tool to evaluate future methods before actual deployment.

LOS and NLOS separation

Many localization techniques assume line-of-sight (LOS) between transmitter and receiver or a much stronger LOS signal component than NLOS components. For indoor and urban localization this is often not the case. To avoid large errors from using a localization method that assumes LOS where there is no direct LOS, i.e. NLOS, other methods are needed.

To estimate self localization, this problem is approached in [77] with two methods based on NNs. The algorithms use channel impulse response (CIR) information from ultra wide-band (UWB) radios. The first method uses a CNN classifier to remove the signals received from a transmitter with only NLOS components. The second method uses the CNN to weigh the result instead of removing it. The CNN operates on raw CIR and successfully solves the classification problem without feature extraction. Best results are achieved with the weights obtained from the CNN used with a weighted least squares (WLS) algorithm and improves localization accuracy over LS-based methods. Finally, different hardware solutions are evaluated which shows that edge device CPUs are sufficient to run the algorithm for an application on UWB.

3.8 Adversarial Attacks

An adversarial example, or adversarial attack, is a crafted sample of input data to an ML algorithm which has been modified in a way that is intended to cause the ML algorithm to provide erroneous output [78]. In what follows, this is the meaning of the term *attack*, no matter what purpose is served by the subjected ML algorithm or the attack itself. Adversarial attacks can be cast into either black-box or white-box attacks depending on what the adversary knows about the victim's NN model [79]. In white-box attacks, the adversary has complete knowledge of the NN, including training data, model architecture and parameters. In black-box attacks the adversary has no or limited knowledge of the NN model. Adversarial attacks can also be divided into digital and physical attacks based on what access the adversary has to the input of the victim system [80]. In digital attacks, the adversary has direct access to the input of the ML model, whereas in physical attacks the adversary indirectly accesses the input to the model for example via OTA transmission.

There is apparent interest in adversarial ML from a defense perspective. Fraunhofer

Institute for Technological Trend Analysis INT has conducted an overview report on that topic on behalf of the Swedish Defence Materiel Administration (Försvarets materielverk, FMV) [81]. This report also outlines international actors that perform research on adversarial ML. None of these, however, focus on wireless communications. Even so, there exist scientific publications that deal with adversarial ML for wireless communications, mostly from the academia. Some of these are briefly summarized in the following.

The robustness against adversarial attacks on the DL signal classification algorithms of [65] is evaluated in [80]. The aim of such attacks is to cause the DL classifyer to make misclassifications. In [80] three different types of adversarial attacks are described, each with varying assumed knowledge of the signal that is to be classified. The results in [80] show that the DL signal classification is extremely vulnerable against attacks. Significantly less transmit power is required by the attacker in order to cause misclassification, as compared to conventional jamming.

The work of [82] also evaluates the impact of adversarial evasion attacks on the modulation classification principle originally proposed in [65]. Tha aim of the attack, in this work, is to prevent an adversary to eavesdrop, or more specifically to classify the modulation type that is used between a communicating transmitter and receiver. The work evaluates digital and physical threats that have, respectively, direct access to classifier input or are synchronously transmitted OTA. Performance of the attack is evaluated in terms of classification accuracy at the eavesdropping DL classifyer, but also in terms of BER experienced at the communication receiver. The attack is generated using the fast gradient sign method (FGSM) [83]. It is demonstrated in [82] that adversarial machine learning is extremely effective when an adversary has direct access to the classifier input, but these vulnerabilities are greatly mitigated when the adversary transmits the perturbation OTA. It is also shown that frequency offset and time shifts impact the adversarial performance negatively. Moreover, although signal classification can be evaded with adversarial attacks, it requires sacrificing spectral efficiency to achieve similar BER.

Another attack designed to fool a DL signal classifier, but with the use of multiple antennas by the adversary, is proposed in [84]. Adding more antennas and transmitting perturbations with equal power from each antenna does not always improve the success rate of the attack. When channel diversity is exploited and transmit power is allocated differently among the antennas the effectiveness of the adversarial attack increases.

A DL-based MIMO CSI feedback network is proposed in [22]. In [85], a white-box adversarial attack is launched against the network of [22]. The results show that an adversarial attack causes a devastating impact on the CSI feedback compared with a jamming attack. Training the work of [22] in an AWGN channel decreases the impact of both the adversarial and jamming attack.

Adversarial examples against the power allocation strategy [37], in a multi-cell massive MIMO network, are studied in [86]. Different ways to craft the adversarial examples are proposed, using gradient-based methods. It is shown in [86] that adversarial attacks can break the DL-based power allocation with a small perturbation in the input of the NN.

It is shown in [87] that end-to-end learning of communication systems through deep

neural network autoencoders can be extremely vulnerable to physical adversarial attacks. The paper demonstrates how an attacker can craft effective physical black-box adversarial attacks and show that the attacks are more destructive than conventional jamming attacks. It is also shown that classical coding schemes are more robust than the autoencoder schemes against both adversarial and conventional jamming attacks.

An attack where the adversary manipulates the training data is proposed in [88]. The so called Trojan attack manipulates the classifier by inserting triggers during training. A DL modulation classifier is considered, using raw (I/Q) samples in the publicly available dataset of [65]. The poisoned training data is used to train the DL classifier. In test (inference) time, an adversary transmits signals with the same phase shift that was added as a trigger during training. It is shown that the receiver cannot reliably classify poisoned signals, while it can accurately classify clean signals without triggers. The paper [88] also shows that the attack can be detected based on outlier detection with clustering techniques.

The works of [89]–[91] consider a spectrum data poisoning attack against a cognitive radio that senses the spectrum and transmits on idle channels determined by a machine learning algorithm. The cognitive radio uses a DL model that predicts idle channels with minimum sensing error for data transmissions, based on spectrum sensing data. The adversary uses another NN model to predict when the transmitter will have a successful transmission. The adversary then performs an OTA spectrum data poisoning attack. In [89], [90] the attack aims to change the channel occupancy status from idle to busy when the transmitter is sensing so that the transmitter is fooled into making incorrect transmit decisions. In [91] the jamming is aimed at the cognitive radio's transmission. It is shown in [89] that the attack causes a significant throughput reduction for the victim cognitive radio. Poisoning during both the training phase and the classification phase are analyzed in [90]. In addition, a defense strategy is designed for the cognitive radio that deliberately makes a small number of incorrect transmissions to manipulate the adversary's training data.

The paper [92] proposes an ML-based spread-spectrum scheme that generates featureless, non-repetitive noise-like spread signals. The proposed spreading scheme provides several benefits over standard direct-sequence spread-spectrum, including the ability to generate signals with low probabilities of intercept and of detection (LPI/LPD) as well as additional processing gain.

An anti-jamming algorithm is proposed in [93] by learning the attacker's jamming strategy and then adapting the rate or exploiting the jamming signals to transmit information (backscattering modulated information on the jamming signals). The algorithm is based on a deep reinforcement learning model. The proposed algorithm allows the transmitter to effectively learn about the jammer and attain effective countermeasures, such as adapting the transmission rate, backscattering, harvesting energy or staying idle. It is demonstrated that the anti-jamming strategy can improve the average throughput and reduce the packet loss under smart and reactive jamming attacks. However, the practicability of such techniques remains to be shown.

4 Possibilities and Challenges of ML for Communications

The literature review presented in Chapter 3, as well as Figure1.1, reveals that extensive research was conducted about ML for communication applications during the last five years.

The extensive research does not necessarily imply that ML is the best choice for all applications, or that it even provides any benefits compared to traditional methods. ML can provide many new opportunities, but may not be a suitable choice for all applications in wireless communications. There are certain requirements or properties of the problem at hand where using ML for communication applications is technically sound. The following list of requirements for when ML is a useful tool is given in [1].

- The traditional engineering flow is not applicable or is undesirable due to a model deficit or to an algorithm deficit, e.g. no physics-based mathematical models exist for the problem or a well-established mathematical model is available but existing algorithms are too complex to be implemented.
- A sufficiently large training data set exists or can be created.
- The task does not require the application of logic, common sense, or explicit reasoning based on background knowledge.
- The task does not require detailed explanations for how the decision was made.
- The phenomenon or function being learned is stationary for a sufficiently long period of time, in order to enable data collection and learning.
- The task has either loose requirement constraints, or, in the case of an algorithm deficit, the required performance guarantees can be provided via numerical simulations.

Two general applications of ML in communications are given in [94]:

- Approximation of a computationally complex algorithm (e.g. non-convex optimization).
- Accurate model does not exist (e.g. traffic pattern).

The use of ML for applications that does not fall within any of the above, and where ML is used just because it can, is probably not sound engineering practice other than for educational or academic purposes.

Several applications that was demonstrated in Chapter 3 use ML to reduce computational complexity compared to existing traditional (non-ML) algorithms. Many publications show that ML-based algorithms achieve near-optimal performance with significantly reduced computational complexity. This is also supported, and often motivated by, the fundamental property of a FCNN to approximate any function with arbitrary accuracy depending on the number of layers and parameters. On the one hand, it is not clear, how well these optimization algorithms perform when generalized to include additional non-idealities in the model. On the other hand, computing an optimal solution with classical methods, with added non-idealities, may also be highly complex and compu-

tationally burdensome. Adding new training data to an ML model, that capture these non-idealities, may be sufficient to extend the algorithm with retained accuracy.

Another trend that may contribute to increased use of ML is the development of dedicated hardware to perform NN computations. Execution of DL models can be implemented in FPGAs, where a large gain in efficiency is from quantization, i.e. calculating with integer numbers instead of floating numbers. To minimize errors from heavy quantization, this error is simultaneously optimized during training. Specialized hardware for trained deep networks is available, but was covered in this report.

Other applications that were presented in Chapter 3 exploit the ability of ML to perform model-free learning. That is, algorithms can be developed and adapted to specific cases and behaviors that are completely or partially unknown. A clear example of this is resource allocation at different levels of complex communication networks, also including traffic modelling. This may involve huge amounts of data and relations between nodes in the communication network that cannot be properly modelled. Another example is signal detection or classification of unknown signals that may exhibit features that are unknown or difficult to model with sufficient accuracy. Yet another example is end-to-end learning of channels that are unknown or difficult to model, potentially spanning multiple traditional algorithms (e.g. multiple blocks of Figure 3.2). In addition, ML-based algorithms may be easier than traditional algorithms to extend to new circumstances. Traditional methods are usually based on a theoretical model. Adding new non-idealites to the model may require significant changes and development to the algorithm. By contrast, an ML-based algorithm can be adapted by re-learning based on newly added training data. That way, the algorithms can be made more robust towards non-ideal effects that appear in real systems. That is, ML may provide an opportunity to construct more robust algorithms also in communication applications.

Another crucial aspect for the development and use of ML in any application is the access to a sufficiently large set of training data with good enough quality. The creation of such data sets may be quite burdensome and costly. For some applications, such as algorithm approximations, it may be sufficient to use synthetic training data. However, to make benefit from the ability to handle unknown models, the training data has to be measured from real systems that capture the specifics of the system. This could be a major hurdle in the development of certain algorithms. Creating good quality data under realistic circumstances is complex, time consuming and costly. There has been some attempts to create publicly available data sets for communication applications, such as [65], but these do not cover all applications and are therefore not sufficient. For some applications, such as in the mobile telephony industry, operators may be able to collect data in their networks to further develop and improve ML-based algorithms. For other applications, such as military mobile ad-hoc networks, it may be a more challenging task to collect such data. It is of uttermost importance to solve this difficulty, to be able to use ML-based algorithms in military communication application.

ML algorithms are learnt based on training data, and not necessarily based on any knowledge of underlying physical properties. That means that many of these algorithms are like a black box that provide output signals with unknown relations to the input signals. For example, deviations in the input signal, that has not been included in the training data, can be handled by the algorithm in an unknown and uncontrollable way. Therefore, it is of significant importance to increase knowledge of the exploitable vulnerabilities

and consequences of using ML algorithms in wireless communications. For security and defense applications, it is essential to be aware of and limit any vulnerabilities or security risks. This report has touched only briefly on the vulnerabilities of using ML for wireless communications. Future research need put more focus on exploitable vulnerabilities and consequences of these on communication performance.

5 Conclusions

This report has focused on existing research literature about exploiting ML techniques for commmunication system applications. An extensive literature review has been conducted, covering many areas of communications.

The extensive amount of research literature demonstrates that there are various applications where ML can be exploited. However, it is important to keep in mind that this does not necessarily imply that ML is the best choice, or that it even offers any benefits compared to traditional methods, for all types of applications. Applications where ML may be beneficial are, for example, such that require reduced computational complexity with near-optimal performance, model-free learning that can be adapted to behaviors that are completely or partially unknown. Examples of such algorithms include resource allocation at different levels of complex communication networks, traffic modelling, signal detection or classification of unknown signals, and end-to-end learning of channels that are unknown or difficult to model.

Access to a sufficiently large set of training data with good enough quality is critical for the development and use of ML in any application. The creation of such data sets may be quite burdensome and costly, and it is of uttermost importance to solve this difficulty to be able to use ML-based algorithms in communication applications.

This report has only briefly touched upon the vulnerabilities of using ML techniques for commmunication applications. Future studies need to put more emphasis on exploitable vulnerabilities and the consequences of these on communication performance. For example, the overall knowledge is limited of adversarial attacks on this type of algorithms and defense mechanisms against such attacks.

Adversarial attacks may also be exploited for purposes that are benefitial in defense and security applications, such as LPI/LPD[1] communications. Such applications should be further studied.

[1] low probability of intercept and low probability of detection

A ML Applications Summary Tables

The literature review in Chapter 3 are summarized in tables in the following. The literature is summarized in one table for each section of Chapter 3 as Section 3.1 Symbol Detection in Table A.1, Section 3.2 Channel Estimation, Equalization and Modeling in Table A.2, Section 3.3 Channel Coding in Table A.3, Section 3.4 Resource Allocation in Table A.4, Section 3.5 End-to-End learning in Table A.5, Section 3.6 Spectrum Sensing in Table A.6 and Section 3.7 Positioning and Localization in Table A.7.

Table A.1: Overview of machine learning for symbol detection.

Ref.	Application	Data type	Input data	ML algorithm(s)	Pros	Cons
[12]	Demodulation without prior channel estimation and equalization	Synthetic	Features from complex baseband signal	DBN, SAE, TTN	Channel equalization not needed	-
[14]	Demodulation under low SNR without explicit equalization, matched filtering	Synthetic	Complex baseband signal	Autoencoder	No need of explicit matched filtering, equalization, continuous estimation of CSI, etc. as in a conventional receiver	Extensive channel model library and pre-trained
[15]	ML Integrated into the Viterbi algorithm	Synthetic	Complex baseband signal	Deep NN	Capable of tracking time-varying channels without needing instantaneous CSI or additional training data, robust to CSI uncertainty	Suboptimal
[16]	MIMO detection	Synthetic	Complex baseband signal, CSI	Deep NN	Computationally inexpensive	Assume perfect CSI

Table A.2: Overview of machine learning for channel estimation, equalization and modeling.

Ref.	Application	Data type	Input data	ML algorithm(s)	Pros	Cons
[17]	Channel estimation	Synthetic	Phase and gain from channel models	CNN	Computational cost	Suboptimal
[18]	Channel estimation	Synthetic	Complex baseband signal and pilot blocks	Sliding Bidirectional Gated Recurrent Unit	Tracks time varying fading channel	-
[19]	Channel estimation	Synthetic	Complex baseband signal, pilot symbols, LS estimates of channel, previous estimated channel	deep NN	Computational cost, no prior knowledge about the channel statistics needed	-
[20]	Channel estimation and equalization for OFDM	Synthetic	Received pilots and transmitted pilots	single hidden layer feed forward network	Computational cost	-
[21]	Joint channel estimation and symbol detection	Synthetic	Complex baseband signal and LS estimates of channel	second-order attention network, GAN	Improved channel estimation accuracy, better BER performance compared to zero-forcing detector	-
[22]	MIMO CSI compression and recovery	Synthetic	Channel matrix	CNN, Auto encoder	Reduced overhead	Reconstruction losses
[23]	MIMO CSI compression and recovery	Synthetic	Channel matrix	LSTM, CNN, Auto encoder	Reduced overhead, exploit time-varying features	Reconstruction losses, fixed numbers of antennas
[24]	MIMO CSI multiple-rate compression and recovery	Synthetic	Channel matrix	CNN, Auto encoder	Reduced overhead, multiple-rate	Reconstruction losses, fixed numbers of antennas
[25]	MIMO CSI compression	Synthetic	Channel matrix	deep CNN, auto encoder	Reduced overhead, flexible in terms of number of antennas	Reconstruction losses
[26]	Channel estimation	Synthetic	Complex baseband signal	deep NN, DIP	Training not required, robust to pilot contamination, outperforms MMSE estimator	-

Table A.3: Overview of machine learning for channel coding.

Ref.	Application	Data type	Input data	ML algorithm(s)	Pros	Cons
[29]	Channel decoding of structured and random codes	Synthetic	Complex baseband signals (LLR and direct output)	deep NN	Computational cost, parallelization	Suboptimal
[33]	Joint source and channel coding	Real	Embedded data (encoder), complex baseband signals (decoder)	RNN, LSTM	Outperforms separated source and channel coding	Fixed bit length in encoding, severe bit restrictions per sentence
[34]	Channel decoding	Synthetic	Complex baseband signals	Pruning-based (PB)-NBP	Outperforms conventional NBP (without pruning) for short codewords, reduced complexity	Subpar performance in other cases
[30]	Channel decoding	Synthetic	Complex baseband signals	deep NN	Achieves coding gain for short codewords	Suboptimal
[31]	Code word construction	Synthetic	Complex baseband signals	RL	Removes necessity of optimal coding knowledge	Subpar performance except for specific cases
[32]	Channel decoding	Synthetic	Complex baseband signals	RL	No additional overhead, outperforms similar methods in a few use cases	Performance heavily dependent on suitable RL-reward algorithm
[35]	Iterative channel decoding framework	Synthetic	Complex baseband signals	BP-CNN	Manages correlated channel noise, outperforms standard BP-decoder, allows parallel computing	Feed-forward network (no memory or state information), suboptimal

Table A.4: Overview of machine learning for resource allocation.

Ref.	Application	Data type	Input data	ML algorithm(s)	Pros	Cons
[37]	Power allocation in massive MIMO	Synthetic	User positions	FCNN	Computational complexity	Suboptimal
[38]	Power allocation in massive MIMO	Synthetic	Large-scale fading coeff.	CNN	Computational complexity	Suboptimal
[39]	Power control in interference channel	Synthetic and real	Channel gains	FCNN	Computational complexity	Suboptimal
[40]	Power control	Synthetic	Channel gains	FCNN	Computational complexity	Suboptimal
[41]	Resource allocation (ex. power control)	Synthetic	Channel gains or functions thereof	FCNN	Computational complexity, model-free learning	Suboptimal (optimal for arbitrary NN size)
[42]	Resource allocation (ex. power control)	Synthetic	Channel gains and number of states that arrive in the time slot	Random edge GNN	Allows extension to larger networks, outperforms model-free heuristics	Suboptimal
[43]	Resource block and power allocation in OFDMA systems	Mix of real and synthetic data	Channel gains and user rate, reliability and latency constraints	RL with GAN	Model-free learning	-
[44]	Power control in a multicast scheme for a wireless downlink	Synthetic	Channel gain and multicast queue	RL by function approximation via deep NN	Computational complexity, learning with unknown system statistics	Suboptimal (approximation of optimal function)
[45]	Power allocation in the uplink of a cell-free massive MIMO network	Synthetic	User positions or shadow fading coefficients	FCNN	Pilot contamination does not significantly affect the learning capability	Suboptimal, shadowing effects cause worse performance
[46]	MAC protocol for heterogeneous wireless networks	State of action	Synthetic	RL, deep Q-network	Faster convergence and better robustness against non-optimal parameter tuning, compared to traditional RL	Suboptimal
[47]	Spectrum collision avoidance by usage prediction	Real	Spectrum usage based on power measure	CNN	Reduced collisions	Short time predictions
[48]	Joint channel and power allocation of a multi-carrier NOMA system	Synthetic	Channel gains and noise powers	RL, NN with encoder-decoder structure	Computational complexity	Suboptimal
[49]	Distributed spectrum and power allocation where multiple V2V links reuse the frequency spectrum of V2I links	Synthetic	Local channel gains	multi-agent RL, fingerprint-based deep Q-network	Improved sum capacity of V2I links and payload delivery rate of V2V links	-
[50]	Joint optimization of beamforming, power control, and interference coordination in a 5G wireless network	Synthetic	Reported coordinates and SINR of UEs	RL, deep Q-learning	Computational complexity, outperform state-of-the-art algorithms	Suboptimal, requires coordinates and SINR every millisecond
[51]	Joint optimization of the passive beamforming vector, decoding order and power allocation in an IRS-aided MISO NOMA network	Synthetic	Channel gains	Three steps with LSTM-based algorithm, K-means based Gaussian mixture model and deep Q-network	Increased sum rate	-

Table A.5: Overview of machine learning for end-to-end applications.

Ref.	Application	Data type	Input data	ML algorithm(s)	Pros	Cons
[57]	Concept of the channel autoencoder	Synthetic	Encoded data	deep NN and CNN, Autoencoder	Computational complexity, reduced power requirements	No analytic channel gradients, error feedback bandwidth and latency constraints
[58]	Deep learning for over-the-air communication	Synthetic, Real	Complex baseband signals, embedded signals	deep NN, Autoencoder	Computational complexity gain by substituting the entire physical layer to NNs	Subpar performance compared to traditional methods
[59]	Channel coding via neural mutual information estimation	Synthetic	Donsker-Varadhan mutual information estimates	deep NN	Implemented only on transmitter side, manages similar performance to state-of-the-art E2E-approaches	Limited comparison in channel model and sample size bounds
[60]	Wireless and channel agnostic E2E-system	Synthetic	Encoded signal and pilot data, CSI	deep NNs, GAN	Does not rely on prior channel information. Can be applied to realistic time-varying channels	Suboptimal
[61]	E2E-learning without a channel model	Synthetic	Encoded data	deep NNs, RL	No reliance on channel model	Requires additional reliable feedback-channel during training of receiver

Table A.6: Overview of machine learning for signal detection and classification.

Ref.	Application	Data type	Input data	ML algorithm(s)	Pros	Cons
[63]	Signal detection	Synthetic	SCM of complex baseband signal	CNN	Require no prior knowledge of background noise or the primary user's activity pattern.	-
[64]	Signal detection	Synthetic, real	Complex baseband signal	CNN, LSTM	Require no prior knowledge about CSI or background noise. Some generalization capability over different modulation types	-
[65]	Modulation classification	Synthetic	Complex baseband signal	CNN	First use of complex baseband signal as input to CNN for modulation classification	Lack of good training dataset
[66]	Modulation classification	Synthetic, real	Complex baseband signal	CNN	Automated signal identification for short time observations, Synthetically trained networks can be transferred to real datatset	Large annotated dataset with good channel models needed
[67]	Modulation classification	Synthetic, real	Amplitude and phase (polar coordinates) or averaged magnitude fast fourier transform (FFT) of the complex baseband signal	LSTM	Variable symbol rate is allowed	-
[68]	Modulation classification	Synthetic	Complex baseband signal	CNN, LSTM, ResNet	Lowering training time with subsampling	
[69]	Anomaly detection and localization in frequency spectrum	Synthetic, real	Complex baseband signal and Power spectrum density	Adversarial autoencoder, CNN, LSTM	Automated labeling of anomalies	Weak definition of spectrum anomaly
[70]	Classify different HF transmission modes	Synthetic	Complex baseband signal	CNN, Residual Net	Computational complexity	-

Table A.7: Overview of machine learning for positioning.

Ref.	Application	Data type	Input data	ML alg.	Pros	Cons
[72]	Positioning without infrastructure	Synthetic	Covariance matrix	Two stages with autoencoder and classifier NN	Compensates for array imperfections, better performance than MUSIC without corrections	Requires a large amount of labeled data.
[73]	Positioning without infrastructure	Synthetic	Covariance matrix	CNN	Improved accuracy and more efficient network compared to [72]	Requires a large amount of labeled data.
[74]	Fingerprint positioning with infrastructure	Synthetic	CSI	CNN	Computational complexity, exploits un-modeled features	Requires large amounts of traning data.
[75]	Fingerprint positioning with infrastructure	Real	CSI	Autoencoder	Computational complexity, exploits un-modeled features	Requires large amounts of traning data. Difficult to collect valid data
[76]	Fingerprint positioning with infrastructure	Synthetic	RSRP	FCNN	Computational complexity, exploits un-modeled features	Requires large amounts of traning data.

References

[1] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems", *IEEE Trans. on Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.

[2] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer", *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Verlag, New York, 2007.

[5] D. Silver, A. Huang, C. Maddison, *et al.*, "Mastering the game of Go with deep neural networks and tree search", *Nature*, vol. 529, pp. 484–489, Jan. 2016.

[6] O. Vinyals, I. Babuschkin, J. Chung, *et al.*, *AlphaStar: Mastering the Real-Time Strategy Game StarCraft II*, `https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/`, 2019.

[7] J. Zhou, G. Cui, S. Hu, *et al.*, "Graph neural networks: A review of methods and applications", *AI Open*, vol. 1, pp. 57–81, 2020.

[8] *ARIADNE*, Online, `https://www.ict-ariadne.eu/`, (Accessed 2021-12-02), 2019.

[9] *DAEMON*, Online, `https://h2020daemon.eu`, (Accessed 2021-12-02), 2021.

[10] *5G-CLARITY*, Online, `https://www.5gclarity.com/`, (Accessed 2021-12-02), 2021.

[11] M. Ibnkahla, "Applications of Neural Networks to Digital Communications - A Survey", *Signal Processing*, vol. 80, no. 7, pp. 1185–1215, 2000.

[12] L. Fang and L. Wu, "Deep learning Detection Method for Signal Demodulation in Short Range Multipath Channel", in *Proc. IEEE Int. Conf. on Opto-Electronic Information Processing (ICOIP)*, 2017, pp. 16–20.

[13] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets", *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[14] A. Al-Baidhani and H. H. Fan, "Learning for Detection: A Deep Learning Wireless Communication Receiver Over Rayleigh Fading Channels", in *Proc. Int. Conf. on Computing, Networking and Communications (ICNC)*, 2019, pp. 6–10.

[15] N. Shlezinger, Y. C. Eldar, N. Farsad, and A. J. Goldsmith, "ViterbiNet: Symbol Detection Using a Deep Learning Based Viterbi Algorithm", in *Proc. IEEE Workshop on Signal Processing Adv. in Wireless Commun. (SPAWC)*, 2019, pp. 1–5.

[16] N. Samuel, T. Diskin, and A. Wiesel, “Deep MIMO Detection”, in *Proc. IEEE Workshop on Signal Processing Adv. in Wireless Commun. (SPAWC)*, 2017, pp. 1–5.

[17] D. Neumann, T. Wiese, and W. Utschick, “Learning the MMSE Channel Estimator”, *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905–2917, 2018.

[18] Q. Bai, J. Wang, Y. Zhang, and J. Song, “Deep Learning-Based Channel Estimation Algorithm Over Time Selective Fading Channels”, *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 125–134, 2020.

[19] Y. Yang, F. Gao, X. Ma, and S. Zhang, “Deep Learning-Based Channel Estimation for Doubly Selective Fading Channels”, *IEEE Access*, vol. 7, pp. 36 579–36 589, 2019.

[20] J. Liu, K. Mei, X. Zhang, D. Ma, and J. Wei, “Online Extreme Learning Machine-Based Channel Estimation and Equalization for OFDM Systems”, *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1276–1279, 2019.

[21] X. Yi and C. Zhong, “Deep Learning for Joint Channel Estimation and Signal Detection in OFDM Systems”, *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2780–2784, 2020.

[22] C.-K. Wen, W.-T. Shih, and S. Jin, “Deep Learning for Massive MIMO CSI Feedback”, *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, 2018.

[23] C. Lu, W. Xu, H. Shen, J. Zhu, and K. Wang, “MIMO Channel Information Feedback Using Deep Recurrent Network”, *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 188–191, 2019.

[24] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, “Convolutional Neural Network-Based Multiple-Rate Compressive Sensing for Massive MIMO CSI Feedback: Design, Simulation, and Analysis”, *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2827–2840, 2020.

[25] Q. Yang, M. B. Mashhadi, and D. Gündüz, “Deep Convolutional Compression For Massive MIMO CSI Feedback”, in *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1–6.

[26] E. Balevi, A. Doshi, and J. G. Andrews, “Massive MIMO Channel Estimation With an Untrained Deep Neural Network”, *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2079–2090, 2020.

[27] V. Lempitsky, A. Vedaldi, and D. Ulyanov, “Deep Image Prior”, in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[28] X.-A. Wang and S. Wicker, “An Artificial Neural Net Viterbi Decoder”, *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 165–171, 1996.

[29] T. Gruber, S. Cammerer, J. Hoydis, and S. t. Brink, “On Deep Learning-Based Channel Decoding”, in *Proc. Conf. on Information Sciences and Systems (CISS)*, 2017, pp. 1–6.

[30] A. Irawan, G. Witjaksono, and W. K. Wibowo, “Deep Learning for Polar Codes over Flat Fading Channels”, in *Proc. Int. Conf. on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019, pp. 488–491.

[31] L. Huang, H. Zhang, R. Li, Y. Ge, and J. Wang, “AI Coding: Learning to Construct Error Correction Codes”, *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 26–39, 2020.

[32] N. Doan, S. A. Hashemi, and W. J. Gross, “Decoding Polar Codes with Reinforcement Learning”, in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[33] N. Farsad, M. Rao, and A. Goldsmith, “Deep Learning for Joint Source-Channel Coding of Text”, in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process. (ICASSP)*, 2018, pp. 2326–2330.

[34] A. Buchberger, C. Häger, H. D. Pfister, L. Schmalen, and A. Graell i Amat, “Pruning and Quantizing Neural Belief Propagation Decoders”, *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1957–1966, 2021.

[35] F. Liang, C. Shen, and F. Wu, “An Iterative BP-CNN Architecture for Channel Decoding”, *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 144–159, 2018.

[36] L. Liang, H. Ye, G. Yu, and G. Y. Li, “Deep-Learning-Based Wireless Resource Allocation With Application to Vehicular Networks”, *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, 2020.

[37] L. Sanguinetti, A. Zappone, and M. Debbah, “Deep Learning Power Allocation in Massive MIMO”, in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, 2018, pp. 1257–1261.

[38] T. Van Chien, T. Nguyen Canh, E. Björnson, and E. G. Larsson, “Power Control in Cellular Massive MIMO With Varying User Activity: A Deep Learning Solution”, *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5732–5748, 2020.

[39] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, “Learning to Optimize: Training Deep Neural Networks for Interference Management”, *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, 2018.

[40] B. Matthiesen, A. Zappone, K.-L. Besser, E. A. Jorswieck, and M. Debbah, “A Globally Optimal Energy-Efficient Power Control Framework and Its Efficient Implementation in Wireless Interference Networks”, *IEEE Trans. Signal Process.*, vol. 68, pp. 3887–3902, 2020.

[41] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, “Learning Optimal Resource Allocations in Wireless Systems”, *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2775–2790, 2019.

[42] M. Eisen and A. Ribeiro, “Optimal Wireless Resource Allocation With Random Edge Graph Neural Networks”, *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, 2020.

[43] A. T. Z. Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, “Experienced Deep Reinforcement Learning With Generative Adversarial Networks (GANs) for Model-

Free Ultra Reliable Low Latency Communication", *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 884–899, 2021.

[44] R. Raghu, P. Upadhyaya, M. Panju, V. Agarwal, and V. Sharma, "Deep Reinforcement Learning Based Power Control for Wireless Multicast Systems", in *Proc. Annual Allerton Conf. on Commun., Cont., and Comp.*, 2019, pp. 1168–1175.

[45] C. D'Andrea, A. Zappone, S. Buzzi, and M. Debbah, "Uplink Power Control in Cell-Free Massive MIMO via Deep Learning", in *Proc. IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Proces. (CAMSAP)*, 2019, pp. 554–558.

[46] Y. Yu, T. Wang, and S. C. Liew, "Deep-Reinforcement Learning Multiple Access for Heterogeneous Wireless Networks", *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, 2019.

[47] R. Mennes, M. Claeys, F. A. P. De Figueiredo, I. Jabandžić, I. Moerman, and S. Latré, "Deep Learning-Based Spectrum Prediction Collision Avoidance for Hybrid Wireless Environments", *IEEE Access*, vol. 7, pp. 45 818–45 830, 2019.

[48] C. He, Y. Hu, Y. Chen, and B. Zeng, "Joint Power Allocation and Channel Assignment for NOMA With Deep Reinforcement Learning", *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2200–2210, 2019.

[49] L. Liang, H. Ye, and G. Y. Li, "Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning", *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, 2019.

[50] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination", *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1581–1592, 2020.

[51] X. Gao, Y. Liu, X. Liu, and Z. Qin, "Resource Allocation in IRSs Aided MISO-NOMA Networks: A Machine Learning Approach", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[52] K. Rusek, J. Suárez-Varela, A. Mestres, P. Barlet-Ros, and A. Cabellos-Aparicio, "Unveiling the Potential of Graph Neural Networks for Network Modeling and Optimization in SDN", in *Proc. ACM Symposium on SDN Research*, San Jose, CA, USA, 2019, pp. 140–151.

[53] P. Almasan, J. Suárez-Varela, A. Badia-Sampera, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, "Deep Reinforcement Learning Meets Graph Neural Networks: An Optical Network Routing Use Case", 2019. [Online]. Available: `http://arxiv.org/abs/1910.07421`.

[54] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey", *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 2019.

[55] Z. Mammeri, "Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches", *IEEE Access*, vol. 7, pp. 55 916–55 950, 2019.

[56] M. Johnston, C. Danilov, and K. Larson, “A Reinforcement Learning Approach to Adaptive Redundancy for Routing in Tactical Networks”, in *Proc. IEEE Mil. Commun. Conf. MILCOM*, 2018, pp. 267–272.

[57] T. J. O’Shea, K. Karra, and T. C. Clancy, “Learning to Communicate: Channel Auto-Encoders, Domain Specific Regularizers, and Attention”, in *Proc. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2016, pp. 223–228.

[58] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, “Deep Learning Based Communication Over the Air”, *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, 2018.

[59] R. Fritschek, R. F. Schaefer, and G. Wunder, “Deep Learning for Channel Coding via Neural Mutual Information Estimation”, in *Proc. IEEE Workshop on Signal Processing Adv. in Wireless Commun. (SPAWC)*, 2019, pp. 1–5.

[60] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, “Channel Agnostic End-to-End Learning Based Communication Systems with Conditional GAN”, in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–5.

[61] F. A. Aoudia and J. Hoydis, “End-to-End Learning of Communications Systems Without a Channel Model”, in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, 2018, pp. 298–303.

[62] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, “Spectrum Sensing for Cognitive Radio: State-of-the-Art and Recent Advances”, *IEEE Signal Process. Mag.*, vol. 29, no. 3, pp. 101–116, May 2012.

[63] J. Xie, C. Liu, Y.-C. Liang, and J. Fang, “Activity Pattern Aware Spectrum Sensing: A CNN-Based Deep Learning Approach”, *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1025–1028, 2019.

[64] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, “Deep Learning for Spectrum Sensing”, *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1727–1730, 2019.

[65] T. J. O’Shea, J. Corgan, and T. C. Clancy, “Convolutional Radio Modulation Recognition Networks”, in *Proc. Int. Conf. Engineering Applications of Neural Networks*, 2016.

[66] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-Air Deep Learning Based Radio Signal Classification”, *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, 2018.

[67] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, “Deep Learning Models for Wireless Signal Classification With Distributed Low-Cost Spectrum Sensors”, *IEEE Trans. on Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, 2018.

[68] S. Ramjee, S. Ju, D. Yang, X. Liu, A. E. Gamal, and Y. C. Eldar, “Fast Deep Learning for Automatic Modulation Classification”, 2019. [Online]. Available: `https://arxiv.org/abs/1901.05850`.

[69] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, "Unsupervised Wireless Spectrum Anomaly Detection With Interpretable Features", *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 637–647, 2019.

[70] S. Scholl, "Classification of Radio Signals and HF Transmission Modes with Deep Learning", 2019. [Online]. Available: `https://arxiv.org/abs/1906.04459`.

[71] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation", *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, 1986.

[72] Z.-M. Liu, C. Zhang, and P. S. Yu, "Direction-of-Arrival Estimation Based on Deep Neural Networks With Robustness to Array Imperfections", *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7315–7327, 2018.

[73] L. Wu, Z.-M. Liu, and Z.-T. Huang, "Deep Convolution Network for Direction of Arrival Estimation With Sparse Prior", *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1688–1692, 2019.

[74] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep Convolutional Neural Networks for Massive MIMO Fingerprint-Based Positioning", in *Proc. IEEE Int. Symp. on Personal, Indoor, Mobile Radio Commun. (PIMRC)*, IEEE, 2017, pp. 1–6.

[75] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach", *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, 2016.

[76] M. M. Butt, A. Rao, and D. Yoon, "RF Fingerprinting and Deep Learning Assisted UE Positioning in 5G", in *Proc. IEEE Vehicular Technology Conf. (VTC2020-Spring)*, 2020, pp. 1–7.

[77] K. Bregar and M. Mohorčič, "Improving Indoor Localization Using Convolutional Neural Networks on Computationally Restricted Devices", *IEEE Access*, vol. 6, pp. 17 429–17 441, 2018.

[78] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Examples in the Physical World", 2016. [Online]. Available: `https://arxiv.org/abs/1607.02533`.

[79] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning", *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, 2019.

[80] M. Sadeghi and E. G. Larsson, "Adversarial Attacks on Deep-Learning Based Radio Signal Classification", *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 213–216, 2019.

[81] K. Ruhlig and D. Thorleuchter, "Adversarial Machine Learning", *Fraunhofer Institute for Technological Trend Analysis INT*, 2020.

[82] B. Flowers, R. M. Buehrer, and W. C. Headley, “Evaluating Adversarial Evasion Attacks in the Context of Wireless Communications”, *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1102–1113, 2020.

[83] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples”, in *Proc. Int. Conf. on Learning Representations (ICLR)*, San Diego, CA, USA, 2015, pp. 1–11.

[84] B. Kim, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, and S. Ulukus, “Adversarial Attacks with Multiple Antennas Against Deep Learning-Based Modulation Classifiers”, in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6.

[85] Q. Liu, J. Guo, C.-K. Wen, and S. Jin, “Adversarial Attack on DL-Based Massive MIMO CSI Feedback”, *Journal of Communications and Networks*, vol. 22, no. 3, pp. 230–235, 2020.

[86] B. R. Manoj, M. Sadeghi, and E. G. Larsson, “Adversarial Attacks on Deep Learning Based Power Allocation in a Massive MIMO Network”, in *Proc. IEEE Int. Conf. on Commun. (ICC)*, 2021.

[87] M. Sadeghi and E. G. Larsson, “Physical Adversarial Attacks Against End-to-End Autoencoder Communication Systems”, *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 847–850, 2019.

[88] K. Davaslioglu and Y. E. Sagduyu, “Trojan Attacks on Wireless Signal Classification with Adversarial Machine Learning”, in *Proc. IEEE Int. Dynamic Spectrum Access Networks (DySPAN) symposium*, 2019, pp. 1–6.

[89] Y. Shi, T. Erpek, Y. E. Sagduyu, and J. H. Li, “Spectrum Data Poisoning with Adversarial Deep Learning”, in *Proc. IEEE Mil. Commun. Conf. MILCOM*, 2018, pp. 407–412.

[90] Y. E. Sagduyu, Y. Shi, and T. Erpek, “Adversarial Deep Learning for Over-the-Air Spectrum Poisoning Attacks”, *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 306–319, 2021.

[91] T. Erpek, Y. E. Sagduyu, and Y. Shi, “Deep Learning for Launching and Mitigating Wireless Jamming Attacks”, *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, no. 1, pp. 2–14, 2019.

[92] I. Shakeel, “Machine Learning Based Featureless Signalling”, in *Proc. IEEE Mil. Commun. Conf. MILCOM*, 2018, pp. 1–9.

[93] N. Van Huynh, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, “”Jam Me If You Can:” Defeating Jammer With Deep Dueling Neural Network Architecture and Ambient Backscattering Augmented Communications”, *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2603–2620, 2019.

[94] E. Björnson and P. Giselsson, “Two Applications of Deep Learning in the Physical Layer of Communication Systems [Lecture Notes]”, *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 134–140, 2020.

FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1000 personnel of whom about 800 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

FOI
Defence Research Agency
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se