Call: H2020-ICT-2020-2

Project reference: 101015956

Project Name:

A flagship for B5G/6G vision and intelligent fabric of technology enablers connecting human, physical, and digital worlds

Hexa-X

Deliverable D4.2

AI-driven communication & computation co-design: initial solutions

Date of delivery:	30/06/2022	Version:	1.0
Start date of project:	01/01/2021	Duration:	30 months

Document properties:

Document Number:	D4.2
Document Title:	AI-driven communication & computation co-design: initial solutions
Editor(s):	Nandana Rajatheva (OUL), Ricardo Marco Alaez (ATO), Andras Benczur (SZT), Tamas Borsos (EHU), Pietro Ducange (UPI), Miltiadis Filippou (INT), Dani Korpi (NOF), Luc Le Magoarou (BCO), Mattia Merluzzi (CEA), Jafar Mohammadi (NOG), Pietro Piscione (NXW), Alessandro Renda (UPI), Elif Ustundag Soykan (EBY)
Authors:	Nandana Rajatheva (OUL), Ricardo Marco Alaez (ATO), Andras Benczur (SZT), Tamas Borsos (EHU), Pietro Ducange (UPI), Miltiadis Filippou (INT), Dani Korpi (NOF), Luc Le Magoarou (BCO), Mattia Merluzzi (CEA), Jafar Mohammadi (NOG), Pietro Piscione (NXW), Alessandro Renda (UPI), Elif Ustundag Soykan (EBY), Leonardo Gomes Baltar (INT), Sokratis Barmpounakis (WIN), Alessio Bechini (UPI), Giacomo Bernini (NXW), Elena Bucchianeri (NXW), Ioannis Chondroulis (WIN), Dilin Dampahalage (OUL), Panagiotis Demestichas (WIN), Hamed Fahadi (EAB), Johan Haraldson (EAB), Ismath Mohamed Insaf (OUL), Leyli Karaçay (EBY), Quentin Lampin (ORA), Vasiliki Lamprousi (WIN), Giada Landi (NXW), Guillaume Larue (ORA), Heunchul Lee (EAB), Dileepa Marasinghe (OUL), Francesco Marcelloni (UPI), Markus Mueck (INT), Christos Ntogkas (WIN), Nuwanthika Rajapaksha (OUL), Vismika Ranasinghe (OUL), Erin Seder (NXW), Adrián Gallego Sánchez (ATO), Emilio Calvanese Strinati (CEA)
Contractual Date of Delivery:	30/06/2022
Dissemination level:	PU[1]
Status:	Final
Version:	1.0
File Name:	Hexa-X_D4.2_v1.0

Revision History

Revision	Date	Issued by	Description
D4.2 ToC	18/11/2021	Hexa-X WP4	ToC
v0.1	13/01/2022	Hexa-X WP4	Version addressing first WP4 internal review
v0.22	11/02/2022	Hexa-X WP4	Version addressing first WP4 internal review
v0.41	17/03/2022	Hexa-X WP4	Version addressing first external review

[1] PU = Public

v0.42	20/04/2022	Hexa-X WP4	Version addressing second external review
v0.45	27/05/2022	Hexa-X WP4	Version addressing PMT review
v1.0	28/06/2022	Hexa-X WP4	Final version for publication

Abstract

This report will describe AI/ML approaches developed in WP4 and their evaluation against the identified KPIs. It will also describe AI/ML solutions for communications to be validated in the project.

Keywords

6G, services, Artificial Intelligence, Machine Learning, Connecting Intelligence

This project has received funding from the European Union's Horizon 2020 research and innovation programmed under grant agreement No 101015956.

Executive Summary

This report is the second deliverable of project Hexa-X work package four (WP4) "AI-driven communication and computation co-design", building on D4.1 providing a comprehensive picture on the intermediate solutions provided by the technical tasks in the work package, i.e., T4.2 and T4.3 giving also an update on the demonstration activity.

The network performance enhancement using AI/ML in 6G is discussed in detail going through various technical enablers mostly coming under T4.2. The main broad topic covered is radio access network performance improvements over classical design methods, This, therefore includes radio access network with the familiar KPIs in terms of bit error rate, spectral efficiency, as well as beamforming and resource allocation improvements. This also provides a discussion on the improvements in E2E network operations and management.

6G as an efficient AI platform is investigated taking technical enablers under T4.3 under the themes; seamless and pervasive in-network AI operation, scalable solutions for distributed AI applications and communication & compute resource allocation. These discuss common technical enablers under each.

AI/ML as an enabler for 6G network sustainability is analysed under both T4.2 and T4.3. The lower complexity is a key aspect here. The results are reported under sustainability by complexity reduction, frugal ML and learning semantics.

Privacy, security & trust in AI-enabled 6G provides work in T4.3 in the areas of federated learning and privacy, explainable AI and design and implementation of Fed-XAI algorithms.

The status of demonstration activities - federated eXplainable AI (FED-XAI) is also discussed.

The progress of the work carried out in WP4 till now is given with respect to different project targets considering both T4.2 and T4.3. The down selected KPIs/KVIs along with various technical enablers addressing those are also given.

Table of Contents

Executive Summary **4**

List of Figures **7**

List of Tables **10**

List of Acronyms and Abbreviations **11**

1 Introduction **16**
1.1 Objective of the document 16
1.2 Structure of the document 16

2 Network performance enhancement using AI/ML in 6G **18**
2.1 Radio access network performance improvements over classical design methods 18
2.1.1 Communication reliability improvements 18
2.1.2 Bit-rate & spectral efficiency improvements 23
2.1.3 Designs accounting for nonlinear distortion 32
2.2 Improvements in E2E network operation & management 35
2.2.1 AI/ML-based predictive orchestration 35
2.2.2 Distributed AI for automated UPF scaling in low-latency network slices 36

3 6G network as an efficient AI platform **39**
3.1 Seamless and pervasive in-network AI operation 39
3.1.1 AI-as-a-Service - enabling a UE to seamlessly exploit network knowledge 39
3.1.2 Network impairment resilience of autonomous agents 41
3.1.3 Distributed low-complexity model learning 43
3.2 Scalable solutions for distributed AI applications 45
3.2.1 Federated ML model load balancing at the edge 46
3.2.2 Scalable and resilient deployment of distributed AI 47
3.2.3 Multi-agent ML for multi-cell multi-user MIMO 50
3.3 Communication & compute resource allocation 51
3.3.1 Flexible computing workload assignment (CaaS) 52
3.3.2 AI workload placement for energy, knowledge sharing and trust optimisation 53
3.3.3 Joint allocation of communication and computation resources for inference at the edge with low energy devices 56

4 AI/ML as an enabler for 6G network sustainability **59**
4.1 Sustainability by complexity reduction 59
4.1.1 Low complexity radio resource allocation in cell-free massive MIMO 59
4.1.2 Supervised learning based sparse channel estimation for RIS aided communications 61
4.2 Frugal ML 64
4.2.1 Over-the-air model learning for data-frugal and resource-efficient network AI operation 64
4.2.2 Low complexity channel estimation using NNs mimicking MMSE 66
4.2.3 Deep unfolding for online unsupervised correction of physical models used in channel estimation 67
4.2.4 Efficient channel charting 69
4.3 Learning semantics: An Opportunity for Effective 6G Communications 71

5 Privacy, security & trust in AI-enabled 6G **75**
5.1 Federated Learning and Privacy 75

5.1.1 Privacy preserving clustering: Federated fuzzy c-means 75
5.1.2 Differentially Private Federated Learning 76
5.1.3 Security mechanism friendly privacy solutions for federated learning 78
5.2 Explainable AI 79
5.2.1 XAI models for QoE prediction 80
5.2.2 XAI for radio network control 81
5.3 Design and implementation of Fed-XAI algorithms 84
5.4 Detection and classification of cybersecurity anomalies in 5G network 85

6 Demonstration activities - Federated eXplainable AI (FED-XAI) demo 87
6.1 UE application with Fed-XAI capability 87
6.2 Functional Requirements 88

7 Conclusions 90

8 References 103

Annex A: Ongoing standardisation activities with relevance to in-network AI/ML 115

List of Figures

Figure 2-1: Three-stage system for detection and tracking, trajectory prediction and blockage detection. 19

Figure 2-2: Evaluation metric score (accuracy, precision, recall and F1 score – all metrics in range [0,1]) vs. prediction window size (no.of sampling time instances) for the simulated data. 20

Figure 2-3: Graph NN architecture for AP selection..... 20

Figure 2-4: Precision and recall of the GNN based AP selection on a cell-free system with 100 [(a)] and 50 [(b)] APs..... 21

Figure 2-5: CS-based scanning reduces the number of measurements M significantly compared to scanning N physical beams. 22

Figure 2-6: The ratio of the predicted best beam being in Top3 best beams with different number of measurements required (128 beams to scan sequentially)..... 23

Figure 2-7: E2E learning for constellation shaping and emission reduction..... 24

Figure 2-8: (a) An example of a learned constellation shape, and (b) the corresponding BER under a nonlinear PA. 25

Figure 2-9: Proposed approach, modelling the transmission chain as an auto-encoder. 26

Figure 2-10: Initial results: promising coding schemes. 28

Figure 2-11: Proposed NN structure. 29

Figure 2-12: Comparison between a classical LBB method (left) and the proposed one (right) in terms of correlation between the precoder and the channel (the higher the better)..... 29

Figure 2 13: Comparison in terms of CDF of the correlation between the proposed method (RFF), classical LBB approaches and with a deep learning approach without RFF (DL)..... 30

Figure 2-14: An example system model containing active “primary” APs (PAPs) and dormant SAPs. 31

Figure 2-15: Modelled physical environment considered for the simulation studies..... 31

Figure 2-16: Regret incurred by the contextual bandit model during training. 32

Figure 2-17: (a) BER of the different schemes with respect to the PA input backoff when the SNR is fixed at 24 dB, and (b) ACLR of the different schemes with respect to the PA input backoff. Note that here the PA backoff is defined with respect to unit variance, i.e., it represents the power of the PA input signal (not to be confused with PA backoff with respect to 1dB compression point). 33

Figure 2-18: Received equalised Discrete Fourier Transform (DFT)-s-OFDM modulated symbols subject to different levels of PA nonlinearities: (a) ideal PA (b) nonlinear PA with 2 dB back-off (c) nonlinear PA with 0.8 dB back-off. 34

Figure 2-19: Schematic diagram of a radio transceiver with ML empowered receiver to compensate for PA nonlinearities. 34

Figure 2-20: The bit error rate (BER) of links with legacy demapper and NN- demapper in the presence of linear PA and nonlinear PA with 4 dB back-off..... 34

Figure 2-21: Continuum Device-Edge-cloud management orchestration. 35

Figure 2-22: Architectural diagram of NWDAF data collection and aggregation. 37

Figure 2-23: Closed-loop pre-emptive auto-scaling of UPF within Edge Compute Node. 38

Figure 3-1: Signalling flow for requesting/delivering of new relevant ML models from a multitude of nodes per some filtering criteria. 41

Figure 3-2: Experimental results for predicting dropped calls, using features for signal strength and other smartphone sensors. 43

Figure 3-3: Overview: channel estimation with transfer learning. 44

Figure 3-4: Delegation of the training task to cloud-based computation resources. 45

Figure 3-5: Model exploitation & retraining policy. 45

Figure 3-6: Left: a FL hot spot with too much data (top) and an insufficient data with one type (camera) missing (bottom). Right: a reconnection decision causes state migration between the bottom AI agents. 46

Figure 3-7: High level architecture of the investigated SNN system. 48

Figure 3-8: Inference accuracy (left) and total neuron activity (right) in case of different spike prioritisation scenarios. 49

Figure 3-9: Schematic of Multiagent Deep Deterministic Policy Gradient in Multiple Input Single Output Inferference Channel. 51

Figure 3-10: Abstracted architecture of a Radio Virtual Machine (RVM) [ETS17]. 52

Figure 3-11: The "follow me computer" CaaS approach in a multi-operator 6G network. 53

Figure 3-12: Overview of AI workload placement algorithm. 55

Figure 3-13: Scenario for dynamic edge inference. 56

Figure 3-14: Energy delay-reliability trade-off in edge AI. 58

Figure 4-1: Sum rate performance comparison between proposed method (PowerNet_Ext) and baseline (optimisation-based) with and without hardware impairments in the transceivers. Dashed lines: Ideal transmitters and receivers without hardware impairments. Solid lines: With hardware impairments in the transmitters and receivers. 61

Figure 4-2: Recorded CPU timing for the CVX solver and the PowerNet_Ext to produce outputs for 100 channel realisations for different network configurations. 61

Figure 4-3: NN architecture for AoA calculation. 63

Figure 4-4: Comparison of performance of proposed methods with LS estimation for both direct and reflected channels, in the off-grid case. 63

Figure 4-5: Data significance pre- and post-evaluation driving resource allocation in a factory setting. 65

Figure 4-6: Turbo AI method compared with lower complexity method developed based on MMSE formulation in terms of NNs. This type of design paradigm allows for producing a solution for wide range of computational complexity limits. 66

Figure 4-7: The Turbo-AI idea repeats a small NN that is inspired by MMSE to handle the correlation in different directions of the channel tensor, namely: frequency, horizontal spatial, vertical spatial, and time. 67

Figure 4-8: One layer of mpNet. 68

Figure 4-9: Comparison of mpNet to several baselines. On the left, 10% of the antennas (chosen uniformly at random) are broken after 100k channels are estimated. On the right, 30% of the antennas are broken. 69

Figure 4-10: Comparison of the proposed approach to several baselines on channels generated with the Quadriga channel simulator. Continuity (CT) and trustworthiness (TW) measures are given (the higher the better) as a function of the size of the considered neighbourhood $\boldsymbol{K}$. 71

Figure 4-11: Transformer-based semantic communication system architecture [SC21]. 72

Figure 4-12: Impact of the SNR and HM(M) on the accuracy. Here we use n = 6 symbols/word over AWGN channel [SC21]. 73

Figure 4-13: 1-gran BLEU Score vs. SNR in the context of AWGN channel French-to-(French/English) translation [SC21]. 74

Figure 5-1: Results on xclara dataset. Average values (shaded region indicates the standard deviation). (left) Federated FCM: Frobenius norm of the difference in the cluster centres between consecutive rounds; (right) $\boldsymbol{Vfed\gamma - Vsum}$ over $\boldsymbol{\gamma}$. 76

Figure 5 2: Illustration of the idea of differential privacy 76

Figure 5-3: Left: Training evaluation loss for different clip-scale values (c) when noise multiplier, z=1 and batch size, B=100. Right: Training evaluation loss for different noise multiplier values when B=100 and c=0.5. 77

Figure 5-4: Security-friendly privacy preserving federated learning scheme. 79

Figure 5-5: (left) Example of fuzzy multiway decision tree. (right) example of strong triangular fuzzy partition on attribute A_f. 80

Figure 5-6: Causal structure of the different feature groups in our use case. 83

Figure 5-7: Illustration of Federated Learning of XAI models 84

Figure 5-8: Heartbeat messages format. 85

Figure 6-1: Overview of AI stacks at user and network side considered for the FED-XAI demo. 87

List of Tables

Table 5-1: Experimental Results: performance comparison between different tree-based models on the QoS-QoE dataset. Average values. 81

Table 5-2: “If-then” rules extracted from decision trees MFDT-4 and BDT-6. 81

Table 5-3: Average of the absolute value of feature attribution made by different methods on EHU 4G mobile radio network data. 84

Table 7-1: Technical enablers and related 6G KPIs/ KVIs to be addressed..... 90

Table 7-2: KPIs considered for AI-driven air interface design. 98

Table 7-3: KPIs considered for sustainable and trustworthy in-network learning. 100

List of Acronyms and Abbreviations

3GPP	3rd Generation Partnership Project
5G-PPP	5G Infrastructure Public Private Partnership
AABB	Axis Aligned Bounding Box
AI	Artificial Intelligence
AIaaS	AI-as-a-Service
AIS	Artificial Intelligence Service
AoA	Angle-of-Arrival
AP	Access Point
APE	Abstract Processing Element
API	Application Programming Interface
AUC	Area Under the Curve
AWGN	Additive White Gaussian Noise
B5G	Beyond 5G
BCH	Bose-Chaudhuri-Hocquenghem
BDT	Binary Decision Tree
BER	Bit Error Rate
BLEU	Bilingual Evaluation Understudy
BP	Belief Propagation
BPSK	Binary Phased Shift Keying
BS	Base Station
CaaS	Compute-as-a-Service
CDF	Cumulative Distribution Function
CNN	Convolutional Neural Network
COTS	Commercial off-the-shelf
CP	Cyclic Prefix
CPU	Central Processing Unit
CS	Compressed Sensing
CSI	Channel State Information
DCB	Deep Contextual Bandit
DFT	Discrete Fourier Transform
D-MIMO	Distributed massive Multiple Input Multiple Output
DMRS	Demodulation Reference Signal

DNN	Deep Neural Network
DO	Data Object
DP	Differential Privacy
DS-OMP	Double-Structured Orthogonal Matching Pursuit
DT	Decision Tree
DW-SNN	Distributed Wireless Spiking Neural Network
E2E	End-to-End
ENI	Experiential Network Intelligence
ETSI	European Telecommunications Standards Institute
FDT	Fuzzy Decision Tree
FEC	Forward Error Correction
FedAvg	Federated Averaging
Fed-XAI	Federated and Explainable Artificial Intelligence
FL	Federated Learning
FPGA	Field Programmable Gate Array
GNBP	Gated Neural Belief Propagation
GNN	Graph Neural Network
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HE	Homomorphic Encryption
IFC	Interference Channel
IFFT	Inverse Fast Fourier Transform
i.i.d.	Independent and Identically Distributed
I/O	Input/ Output
IoT	Internet of Things
IP	Internet Protocol
IRTF	Internet Research Task Force
ISG	Industry Specification Group
ITS	Intelligent Transportation System
ITU	International Telecommunications Union
KPI	Key Performance Indicator
KVI	Key Value Indicator
LBB	Location-Based Beamforming
LDPC	Low-Density Parity-Check code

LiDAR	Light Detection And Ranging
LLR	Log Likelihood Ratio
LMMSE	Linear Minimum Mean Square Error
log-MAP	log-Maximum A-Posteriori
LoS	Line of Sight
LS	Least Squares
LSTM	Long Short Term Memory
M&O	Management and Orchestration
MA-DDPG	Multiagent Deep Deterministic Policy Gradient
MAPE	Mean Absolute Percentage Error
MEC	Multi-access Edge Computing
MFDT	Multi-way Fuzzy Decision Tree
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
ML	Machine Learning
MLP	Multilayer Perceptron
MMSE	Minimum Mean Squared Error
mmWave	Millimeter Wave
MSE	Mean Squared Error
NBP	Neural Belief Propagation
NF	Network Function
NLoS	Non Line of Sight
NLP	Natural Language Processing
NN	Neural Network
NPU	Neural Processing Unit
NWDAF	Network Data Analytics Function
OAM	Operation, Administration and Management
OBB	Oriented Bounding Box
OFDM	Orthogonal Frequency Division Multiplexing
OMP	Orthogonal Matching Pursuit
PA	Power Amplifier
PAP	Primary Access Point
PDU	Protocol Data Unit
PHY	Physical Layer
QAM	Quadrature Amplitude Modulation

QoE	Quality-of-Experience
QoS	Quality-of-Service
RAM	Random Access Memory
RAN	Radio Access Network
RAT	Radio Access Technology
RBS	Rule-Based System
RE	Resource Element
ReLU	Rectified Linear Unit
RF	Random Forest
RFF	Random Fourier Feature
RIS	Reconfigurable Intelligence Surface
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RRM	Radio Resource Management
RSRP	Reference Signal Received Power
RVM	Radio Virtual Machine
RX	Receiver
SAI	Securing AI
SAP	Sleeping Access Point
SDO	Standards Development Organisation
SGD	Stochastic Gradient Descent
SINR	Signal to Interference plus Noise Ratio
SMPC	Secure Multi-Party Computation
SNN	Spiking Neural Network
SNR	Signal to Noise Ratio
SP	Service Provider
SSL	Sockets Layer
TA	Timing Advance
TCP	Transmission Control Protocol
TFF	Tensorflow Federated
TLS	Transport Layer Security
TTL	Time To Live
TX	Transmitter
UAV	Unmanned Aerial Vehicle
UE	User Equipment

ULA	Uniform Linear Array
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency Communications
V2X	Vehicle-to-Everything
VM	Virtual Machine
VNF	Virtualised Network Function
XAI	Explainable AI
ZSM	Zero touch network & Service Management

1 Introduction

Hexa-X is one of the 5G-PPP projects under the EU Horizon 2020 framework. It is a flagship project which develops a Beyond 5G (B5G)/ 6G vision and an intelligent fabric of technology enablers connecting human, physical, and digital worlds. Relevant information has been documented in D4.1 "AI-driven communication & computation co-design: Gap analysis and blueprint" [D4.1]. This report is the second deliverable of work package four (WP4) "AI-driven communication and computation co-design", led by tasks T4.2 - "AI-driven air interface design" and task T4.3 - "Methods and algorithms for sustainable and secure distributed AI" based on scenarios KPI/KVI identified in D4.1 and further refined. It focuses on the technical areas related to applying AI/ML to networking and concentrates on associated technical enablers to be investigated within WP4, as well as on the current results obtained.

1.1 Objective of the document

The purpose of this document is to discuss concrete details of the application of Artificial Intelligence (AI), including, but not limited to, Machine Learning (ML) mechanisms in 6G systems and provide intermediate solutions. The report, therefore, delves into a detailed investigation of relevant technical enablers and their performance and validations, which can be considered as a midpoint achievement of WP4.

Input is yielded from deliverables: D1.1 "6G Vision, use cases and key societal values", D1.2 "Expanded 6G vision, use cases and societal values –including aspects of sustainability, security and spectrum", D2.1 "Towards Tbps Communications in 6G: Use Cases and Gap Analysis", D6.1 "Gaps, features and enablers for B5G/6G service management and orchestration" and D7.1 "Gap analysis and technical work plan for special-purpose functionality".

The resulting guidelines provided by these deliverables are intended to direct the work in the following tasks of WP4, namely, task 4.2 (T4.2) "AI-driven air interface design" and task 4.3 (T4.3) "Methods and algorithms for sustainable and secure distributed AI". The outcomes of the work presented in this deliverable is consolidated in Chapter 7.

1.2 Structure of the document

The document is structured in the following way:

Chapter 2 explores network performance enhancement using AI/ML in 6G focusing on two main research design targets, i.e., i) radio access network performance improvements over classical design methods and ii) improvements in End-to-End (E2E) network operation and management.

Chapter 3 presents 6G network as an efficient AI platform, where the following topics are covered: i) seamless and pervasive in-network AI operation; ii) scalable solutions for distributed AI applications and iii) communication and compute resource allocation.

Chapter 4 discusses AI/ML as an enabler for 6G network sustainability featuring the topics of: i) sustainability by complexity reduction; ii) frugal ML and iii) learning semantics for effective 6G communications.

Chapter 5 investigates data privacy, security & trust in AI-enabled 6G, elaborating on the topics of: i) security and privacy for Federated Learning (FL), ii) explainable AI (XAI), iii) design and implementation of federated XAI (Fed-XAI) algorithms and iv) detection and classification of cybersecurity anomalies in 5G network.

Chapter 6 gives an outline about demonstration activities to be carried out in WP4 (FED-XAI demo).

The document concludes and provides overall guidelines in Chapter 7.

2 Network performance enhancement using AI/ML in 6G

In this section, the emphasis is on how AI/ML-based solutions enhance the network performance in a quantifiable way. The presented solutions will contribute to the expected performance improvements of 6G to effectively manage the network communications at radio and architectural resources with the help of deep learning. The first part of the section focuses on link-level enhancements, where the performance gains are quantified in terms of Bit Error Rate (BER)/ Block Error Rate (BLER) improvement, channel estimation error reduction, complexity gain, bit rate or spectral efficiency improvements, flexibility improvement, and improved mobility support. In addition, there are also proposed solutions for building resilience against nonlinear distortion, with the help of ML. In the latter part of the section, the focus shifts to the network-level aspects. In particular, proposals solutions to improve E2E operation and management via AI/ML-based predictive orchestration and automated User Plane Function (UPF) scaling using distributed AI. The performance gains achieved with these solutions can be quantified in terms of latency, network energy efficiency, and inferencing accuracy.

2.1 Radio access network performance improvements over classical design methods

This subsection presents the ML-based link-level enhancements, the considered scenarios ranging from individual links to distributed MIMO systems. The primary focus areas are solutions for reliability improvements, bit rate improvements, and dealing with nonlinear distortion.

2.1.1 Communication reliability improvements

2.1.1.1 LiDAR aided human blockage prediction for 6G

With the use of higher frequencies and more antennas in the transceivers, use of very narrow beams is feasible in 6G. At these frequencies, the link ispredominantly depending on the line-of-sight beam. Human blockage has been identified as a significant degrading factor to such links [MRR17], particularly in indoor scenarios, where large and dense crowds are present. Such blockages dynamically disrupt the link, because of movements of people. If such movements can be monitored, it is possible to predict the blockage before it happens and avoid transmission during blockage period. The target of such prediction is to improve reliability of the link, reduce link failures while supporting mobility and adapting to dynamics of the environment. Such monitoring capability can be enabled by exploiting vision-like sensors, which can capture the environment dynamics in a detailed manner. Light Detection And Ranging (LiDAR) sensors appear as an interesting sensor candidate, which is able to capture rich 3D environment data, while preserving the user privacy as such data cannot be used in facial recognition opposed to camera images. Therefore, this method utilises infrastructure mounted LiDARs to monitor indoor activity and use currently measured sensor data to predict dynamic human blockages. We propose an E2E solution, which comprises detection of humans, tracking of humans, then predicting their trajectories and, finally, evaluating the future blockage possibility using simple ray-casting techniques. Figure 2-1 gives an overview of the system.

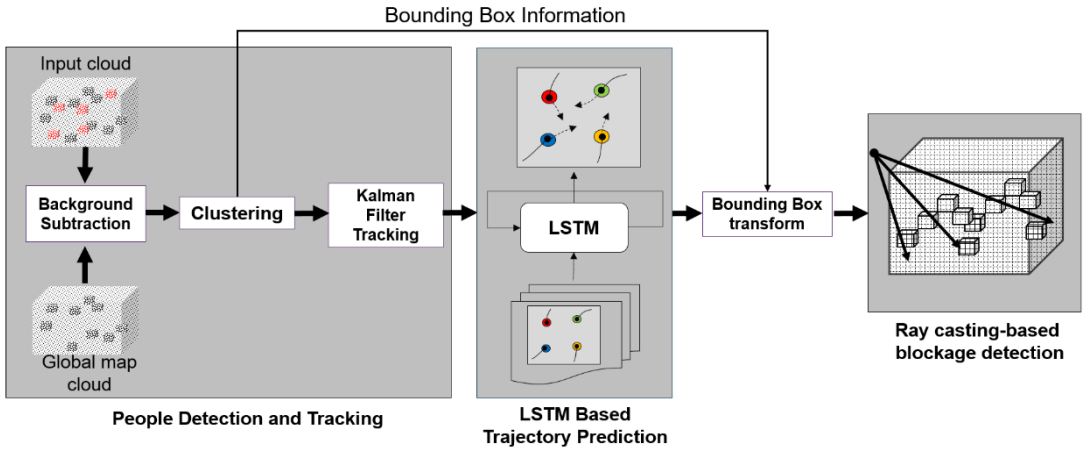


Figure 2-1: Three-stage system for detection and tracking, trajectory prediction and blockage detection.

The proposed method uses point cloud processing to extract dynamic points from the LiDAR data, then cluster them to identify humans and track them in between the LiDAR frames based on the method proposed in [KMM19]. Then the trajectory history of each human is used to predict the future path using a long short-term memory (LSTM) based network [KKA21]. Finally, the predicted human positions and cluster bounding box information from point cloud processing are used to evaluate the future blockage probability in a future prediction window using a ray-box intersection technique used in computer graphics [WBK+05]. We used two types of bounding boxes, namely the axis-aligned bounding boxes (AABB) which bounds the object with max and min values in each axis and oriented bounding boxes (OBB) which also includes the rotation of the object in each axis. The system was evaluated (Figure 2-2) with synthetic LiDAR data for 500 scenes with 10 humans, where trajectories are generated such that the humans start randomly from a point in the circumference of a 12.5 m radius circle and walk towards the antipodal position on the circle avoiding collisions with each other. For this blockage prediction problem, the accuracy means correctly identified LoS windows and NLoS windows out of all the prediction windows considered. Further, precision and recall were considered as the classifier output quality measures. Precision is defined as the number of true positives over the number of true positives plus the number of false positives and recall is defined as the number of true positives over the number of true positives plus the number of false negatives. Here, the precision indicates the fraction of blockage windows correctly predicted from all the predicted blockage windows while recall indicates the fraction of blockage windows correctly predicted from the true blocked windows. The F1 score gives the harmonic mean between precision and recall and a high F1 score means a better predictor. The results reflected that the method can predict future blockage events with an accuracy of 87% with a precision of 78% and a recall of 79% for a prediction window of 300 ms, when the sensors are operating at 10 Hz.Further details are available in the associated paper [MRL21]. This implies that the method can identify incoming blockage events well in advance

and notify to the communication system to act by either beam change or avoid transmission during the blockage period.

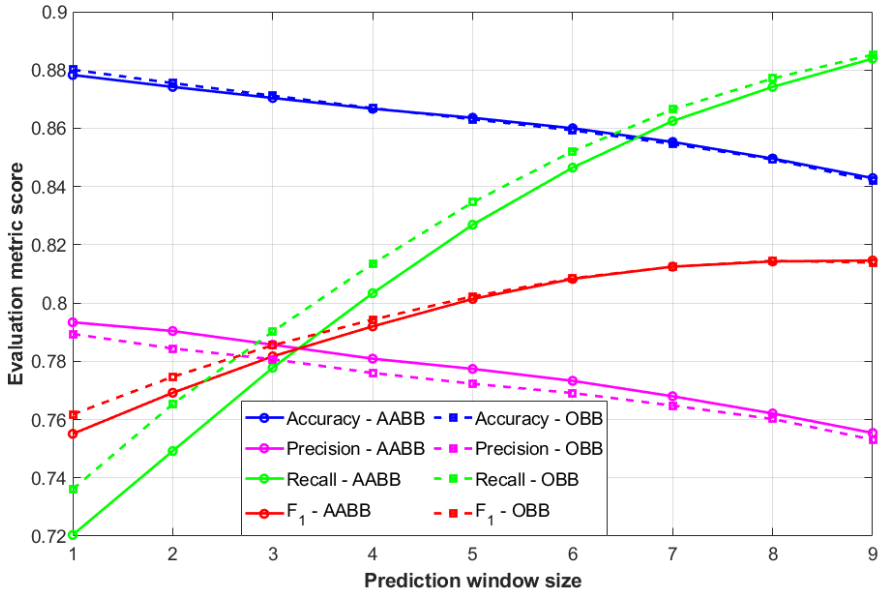


Figure 2-2: Evaluation metric score (accuracy, precision, recall and F1 score – all metrics in range [0,1]) vs. prediction window size (no.of sampling time instances) for the simulated data.

2.1.1.2 Graph neural network-based access point selection in cell-free massive MIMO systems

In a cell-free system, multiple APs serve a single user in nonorthogonal time frequency resources to achieve diversity gain and improve spectral efficiency. Hence, the AP selection in initial access and mobility management plays a crucial role in improving the latency when achieving the mobility support KPI targets. Compared to the large scale fading based AP selection algorithms used in [NTD+18], [DK20], which requires measurements of the signal strengths of all identified APs, AI/ML can be leveraged to predict the candidate APs based on the very limited number of measurements performed by the user equipment (UE). The idea is that the AI/ML method employed being able to learn the shadowing

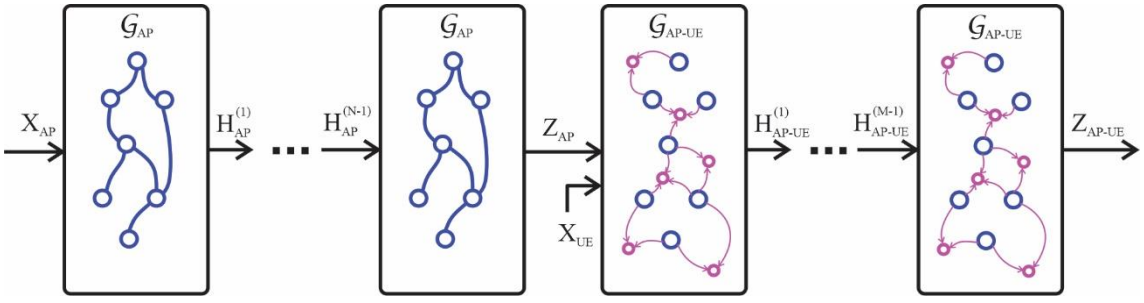


Figure 2-3: Graph NN architecture for AP selection.

due to static features of the environment. This enables more accurate AP selection compared to the proximity-based AP selection, i.e., selecting the APs neighbouring to the strongest AP [BS20]. By representing a cell-free network as a heterogenous graph, a graph neural network (GNN) can be used to predict the candidate APs, given there are a limited number of signal strength measurements from known identified APs. Once the UE is connected to the AP with maximum

signal strength (master AP) it has identified, master AP can request signal strength measurements to a known set of neighbouring APs via radio reconfiguration signalling. Based on the received measurements, a GNN trained to predict the candidate APs with comparable signal strength to the signal strength of the master AP can be used for AP selection. Figure 2-3 illustrates the architecture of the proposed GNN which is based on GraphSAGE, an inductive graph representation learning platform presented in [HYL17]. To make the AP embeddings independent from input features of the UE, graph convolutions are performed on two different graphs, where one includes only the AP nodes ($\mathcal{G}_{AP}$) , while the other graph includes both AP and UE nodes ($\mathcal{G}_{AP-UE}$). After obtaining the AP input feature vectors (X_{AP}) based on the measured signal strength between APs, N graph convolutions are performed on $\mathcal{G}_{AP}$ to obtain final embeddings of AP nodes (Z_{AP}). For the Mgraph convolutions on $\mathcal{G}_{AP-UE}$, the input feature vectors of the UE nodes (X_{UE}) along with Z_{AP} forms the input feature set for both UE and AP nodes, respectively. Here, to obtain X_{UE}, along with the signal strength of the master AP, two additional signal strengths of the APs closest to master AP are used. Finally, the candidate AP prediction is formulated as a link prediction task on $\mathcal{G}_{AP-UE}$ based on the obtained final embeddings of the APs and UEs (Z_{AP-UE}).

Figure 2-4 illustrates the precision and recall of the link prediction task for cell-free systems with 100 APs and 50 APs. Furthermore, two scenarios are considered depending on the number of neighbouring APs considered for the link prediction task i.e., c_{UE}. In this simulation, the signal strength between the master AP and the candidate AP is being less than 10dB, this is used as the criterion for the AP selection. From Figure 2-4, it can be observed that when $c_{UE} = 10$ the proposed method achieves a precision of 80% and a recall of 69% for both cell-free systems with 100 and 50 APs. Achieving a high precision in predicting the candidate APs, including the APs of which the signal strength has not being measured, using a limited number of measurements is illustrative of the potential of an AI/ML based predictive AP selection algorithm to reduce latency in initial access and mobility management in a cell-free network. More details about the proposed algorithm and the implementation are available in [RRL21].

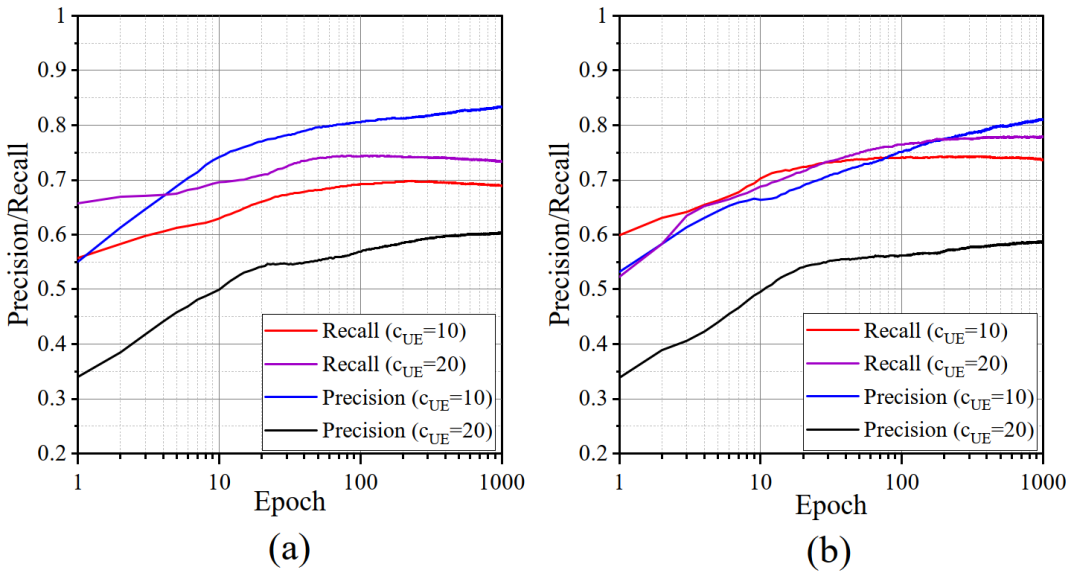


Figure 2-4: Precision and recall of the GNN based AP selection on a cell-free system with 100 [(a)] and 50 [(b)] APs.

2.1.1.3 AI based compressed sensing for beam selection in D-MIMO

6G wireless communication systems are expected to be able to operate at high frequencies in the mmWave range, where beamforming is a necessary technique to improve reliable coverage. Base Stations (BSs) with multiple antenna elements can focus antenna characteristics to specific physical directions towards individual UEs to increase received power in both directions. Finding the best beam in all situations is a cornerstone of the beamforming problem, which appears both at initial beam search and at beam tracking/update. At initial beam search when the UE has no connectivity to the network, it must scan all possible beam directions to find the best one. The problem can become even more severe in Distributed massive MIMO (D-MIMO) settings where there are a large number of APs (typically more than UEs) each with a couple of transmit antennas and possible beam directions.

Independent of beamforming, the compressed sensing or compressive sensing (CS) theory is a relatively newly explored scheme, which has found its application in different areas more recently, e.g., imaging applications or in radar signal processing. CS is a signal processing technique which says that if a signal is sparse in some domain, then the signal can be fully reconstructed from fewer number of measurement samples, than what would be required by sampling theory. Being sparse means that the signal contains a lot of zero elements when it is expressed in a certain domain or basis.

In the case of (distributed) massive MIMO mmWave systems the radio propagation on multipath channels is known to be very sparse and only several beams may be able to reach a UE out of hundreds. In the application of CS in beam selection, the APs can transmit the same reference signal in all directions using the same time frequency resource and varying the transmit power allocations in a few combinations. The UE needs to measure only the composite received signal from all directions, it does not have to identify beams individually. The best beam direction(s) and corresponding channel gains can be determined using CS computation done either at the UE or at the network.

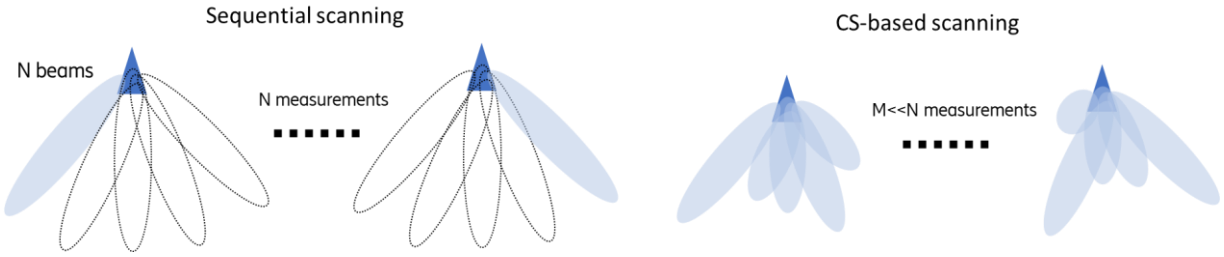


Figure 2-5: CS-based scanning reduces the number of measurements M significantly compared to scanning N physical beams.

Explaining CS in an algebraic way, we can formulate the CS problem as solving an under determined set of linear equations. Let $\{\boldsymbol{d_i}\}$ represent measurement vectors of length N, which maps to beam power allocation patterns, $\boldsymbol{x}$ is a vector of length N mapping to channel power loss (see Figure 2-5) due to large-scale propagation and y_i is one measurement representing received power aggregated from all beams. One measurement is formulated as $y_i = \boldsymbol{d}_i^T \boldsymbol{x}$, while all M<<N measurements yield $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}$, where $\boldsymbol{D}$ is the dictionary matrix of size $(M \times N)$.

In this case, sparse decoding is typically formulated as

$$argmin_{\hat{\boldsymbol{x}}} \|\hat{\boldsymbol{x}}\|_p \quad s.t. \quad \|\boldsymbol{D}\hat{\boldsymbol{x}} - \boldsymbol{y}\|_2 < \varepsilon, \quad 0 \le p < 2 \qquad \text{(Eq 2-1)}$$

Where $\|.\|_p$ is the p-norm. There are multiple algorithms for finding a good sparse solution, e.g., using LASSO l_1 regularisation:

$$min_{\hat{\boldsymbol{x}}} \|\boldsymbol{D}\hat{\boldsymbol{x}} - \boldsymbol{y}\|_2^2 \quad + \gamma \|\hat{\boldsymbol{x}}\|_1 \qquad \text{(Eq 2-2)}$$

CS theory formulates requirements on dictionary **D**, however these are often asymptotical properties. On the other hand, there are many good choices, e.g., random matrices with i.i.d. Gaussian or Bernoulli entries. D-MIMO deployments are expected to operate with very high number of beams from different APs, where finding the best dictionary **D** for beam selection efficiency is even more important.

In these scenarios, beam pattern distributions are not uniform and local environment statistics can be utilised. In particular, it is possible to optimise the dictionary D for a given environment to reduce the number of required measurements. This study aims to propose ML/AI techniques to learn the dictionary and evaluate the potential gains in terms of required measurements and, consequently, energy efficiency.

Initial analysis has been performed based on the open DeepMIMO dataset, with 4 access points and 32 horizontal beams on each of them. That means 128 beams altogether with 128 measurements if sequential scanning is used. Figure 2-6 below shows results for 3 solutions. One with random dictionary and iterative sparse decoding using LASSO. The other two scenarios show the results with a dictionary trained for the local environment and sparse decoding performed with either the iterative optimisation or the fully NN-based sparse decoder (e.g., LISTA). Application of these techniques enable smaller beam scanning times, better resolution for beam predictions, which leads to improved mobility reliability KPIs.

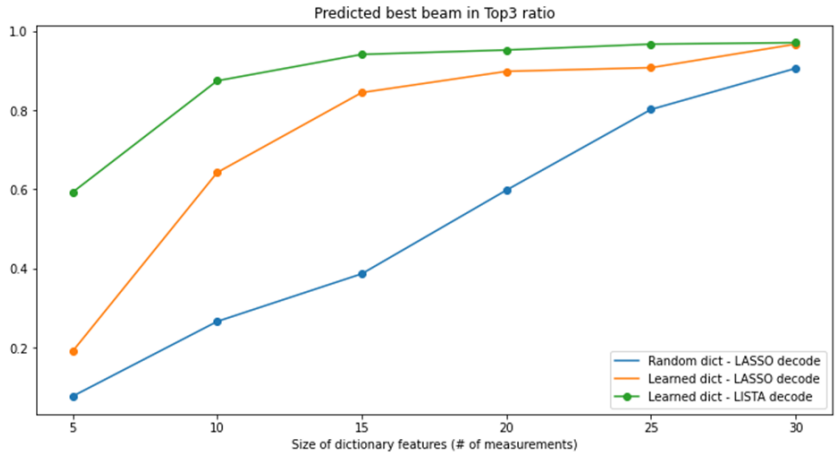


Figure 2-6: The ratio of the predicted best beam being in Top3 best beams with different number of measurements required (128 beams to scan sequentially).

2.1.2 Bit-rate & spectral efficiency improvements

2.1.2.1 Constellation shaping

The primary target of each new wireless communications system generation should be a higher bit rate for the end user, and one key method for achieving this is to enhance the spectral efficiency. This is also the case for 6G, and its AI-native air interface. There are various ways in which ML-based solutions can aid in achieving this goal, both in the physical layer, as well as on the network level.

One such method is removing the need for transmitting pilot symbols within the waveform [KHH+22]. Namely, as opposed to 5G and earlier network generations, where the receiver must be provided so-called Demodulation Reference Signals (DMRSs) for channel estimation and consecutive equalisation, an AI-native air interface can be trained to operate without such pilots.

This naturally removes the overhead they incur as all the Resource Elements (REs) can be used to carry useful / payload data.

The mechanism or technology for engaging in pilotless transmissions is based on constellation shaping, where the geometric shape of the complex-valued constellation is learned. Figure 2-7 illustrates this on a conceptual level. In this example, an, otherwise, conventional Orthogonal Frequency-Division Multiplexing (OFDM) transmitter is assumed, with the exception that the bits are mapped to symbols using the learned constellation. In addition, a small convolutional neural network (CNN) is added to the transmitter inverse fast Fourier transform (IFFT) output in order to reduce the out-of-band emissions, due to a nonlinear power amplifier (PA). It is crucial that this CNN operates in time-domain, as PA-induced nonlinear distortion is better modelled as time-domain phenomenon (for further discussion regarding the suppression of out-of-band emissions, please refer to Section 2.1.3.1).

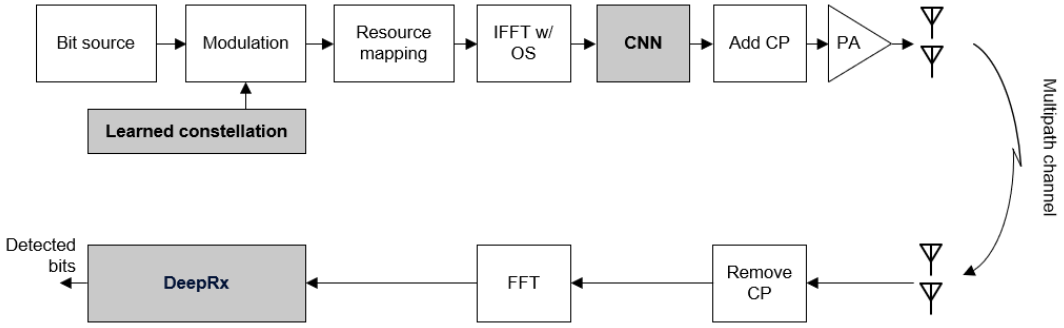


Figure 2-7: E2E learning for constellation shaping and emission reduction.

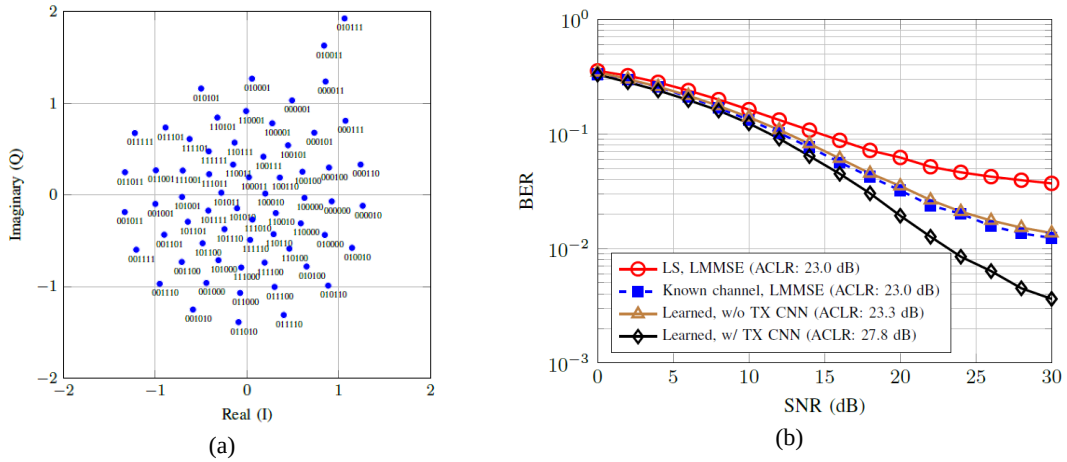
The complete transmission link is trained E2E, which can be done with supervised learning. This requires a differentiable model between the bit source and the receiver output, including the transmitter, multipath channel, and the receiver. The training is done based on the binary cross entropy loss between the transmitted bits and the detected bit estimates. With this, the transmitter and receiver are incorporated into a single E2E model and can thereby learn jointly the constellation shape and the required receiver-side processing for detecting the modulated bits. Note that such E2E learning is feasible only offline, and any post-deployment training requires a somewhat different training procedure, such as utilising reinforcement learning (RL) for the transmitter side, and supervised learning for the receiver side [AH18]. This will introduce a certain amount of intermittent overhead, the exact amount of which is dependent on the relationship between the training and deployment environments and hence very difficult to quantify.

Figure 2-8 shows an example of a learned constellation for a multipath scenario, as well as the achieved uncoded BER. The training is carried out with the TensorFlow library, where the full link is implemented as an E2E model. This also includes the channel model, which is based on precomputed channel coefficients generated under an urban micro (UMi) simulation scenario. In these results, the transmitter PA is operating near saturation. The proposed ML-based solution is compared to two baselines:

• A practical receiver, which estimates the channel from pilots using least squares, and interpolates it linearly over the whole slot (extrapolation of the channel estimate beyond the last pilot symbols is done with the nearest neighbour rule). The symbol estimates are obtained with linear minimum mean square error (LMMSE) equalisation, after which the soft bits are calculated using the log Maximum A-Posteriori (log-MAP) rule.

• A genie-aided receiver, which has perfect channel knowledge, and performs LMMSE equalisation followed by log-MAP demapping.

Both benchmarks utilise a conventional Quadrature Amplitude Modulation (QAM)-OFDM waveform with 2 OFDM symbols per slot (14 OFDM symbols) dedicated to pilot transmissions. It is evident from Figure 2-8(b) that the learned pilotless scheme is superior to the benchmark schemes in terms of the achievable spectral efficiency, as it achieves a lower BER with less overhead. The pilotless operation is made possible by the learned asymmetric constellation shape, shown in Figure 2-8(a), which the receiver can use for blind detection of transmitted symbols. The lower BER of the learned scheme is mostly due to the nonlinear distortion within the RX signal, which the baseline schemes are not capable of suppressing. In a linear communication system, the throughput gains stem purely from the reduction in pilot overhead [KHH+22, AH21]. Altogether, these findings, therefore, demonstrate the potential of an AI-native air interface in reducing BER, improving the spectral efficiency and reducing the signaling overhead over the current systems. Hence, this solution directly addresses the target KPIs such as BER gain, more efficient resource utilisation, and rate gain.



(a) (b)

Figure 2-8: (a) An example of a learned constellation shape, and (b) the corresponding BER under a nonlinear PA.

2.1.2.2 Neural network and machine learning aided channel (de)coding for constrained devices

Context

The availability of efficient Forward Error Correction (FEC) mechanisms is a key enabler for the development of Ultra-Reliable Low Latency Communications (URLLC) and/or Internet of Things (IoT) scenarios. These use cases typically impose strict constraints, e.g., in terms of energy consumption, available computing power, latency, hardware cost, which tend to favour the use of short packets (datagrams of a few tens of bits to a few hundreds of bits) and require the definition of low-complexity coding/decoding algorithms.

AI/ML techniques are expected to play an important role in future generations of communication networks and have naturally been applied to channel coding and notably to the short block length regime, e.g. [NML+18], [YLL19], [GCH+17]. However, AI/ML-aided linear block coding approaches usually face two main categories of challenges: the curse of dimensionality and the differentiability (or lack thereof) of the model (more generally, all the properties of the model with regard to the training e.g., convexity, smoothness, separability, etc.):

- The curse of dimensionality is a well-known phenomenon in ML, related to the fact that the feature space of a problem increases exponentially with its dimensionality, making the training data sparse and therefore non-statistically significant and/or the required amount of training material prohibitive. A similar phenomenon, called the "curse of codeword dimensionality", occurs in the context of the physical layer (PHY) when one tries to train a model with larger blocks of information. Indeed, when learning a communication scheme with blocks of dimension k, the number of possible input words is 2^k, which grows exponentially with k and can therefore quickly become a prohibitively large number. In other words, it become difficult to learn large code as the code-words space increase exponentially with the block length [OH17], [ASR+20]. Practically, the solution space of codes longer than a few tens of bits cannot be evaluated exhaustively.

- NN models in the domain of PHY layer can potentially present several non-differentiability issues, owing to the propagation channel, the digital modulation, the extensive use of finite field (e.g., GF2), discrete variables (parity-check and generator matrices) and related modulo arithmetic (e.g., XOR operator). This poses the question of the compatibility of all the model's layers with regards to ML techniques, such as gradient descent that requires, among other things, the differentiability of the loss function with regards to all the trainable model parameters.

Proposed approach:

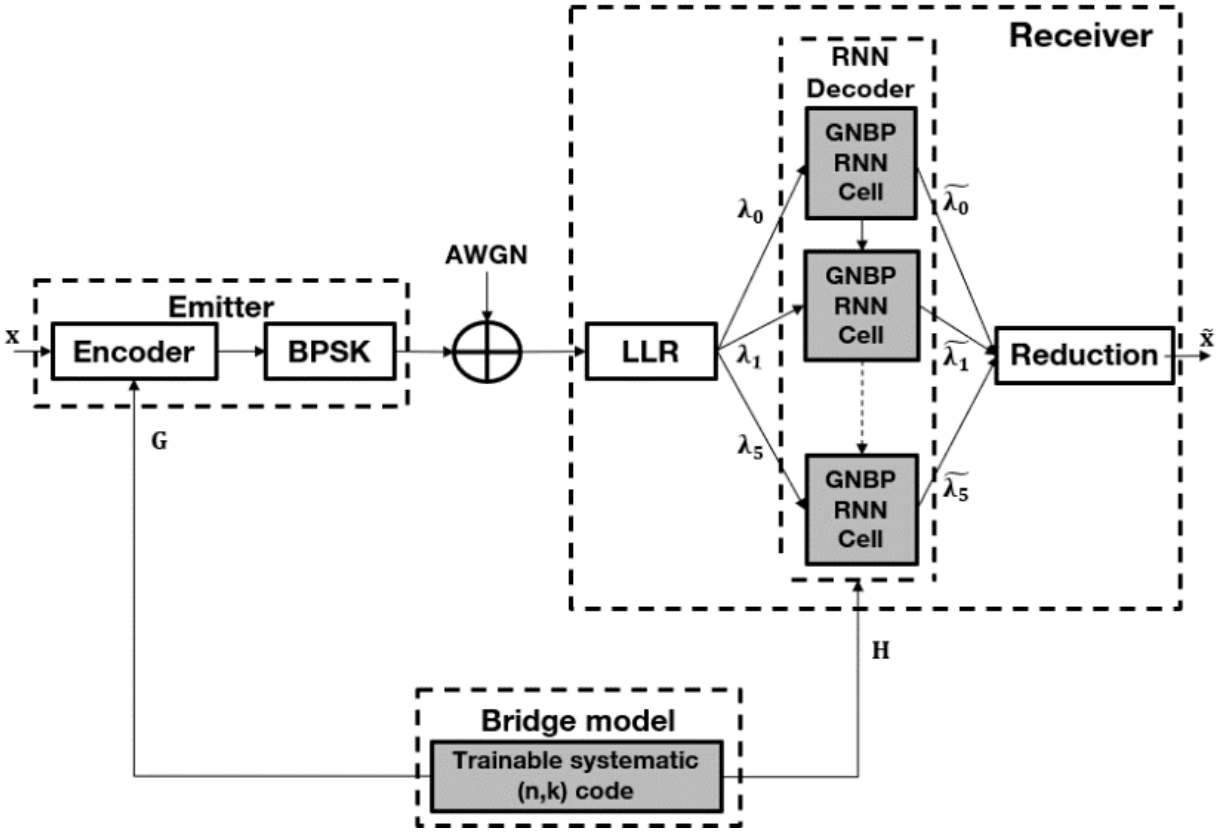


Figure 2-9: Proposed approach, modelling the transmission chain as an auto-encoder.

We propose to design and optimise linear block codes and decoders modelled jointly in an auto-encoder model. Specific structures are employed, as it will be detailed, to circumvent aforementioned challenges of curse of dimensionality and differentiability and allow for the training of relatively important size of code (around a hundred bits). A simplified view of the proposed auto-encoder model is described in Figure 2-9.

The encoder is rigorously defined as an explainable model of a linear block encoder. To ensure operation differentiability with regards to the code's generator matrix parameters, the XOR operation is represented as a product of bipolar inputs (0 mapped to +1 and 1 mapped to -1). The inputs represent binary data words and the generator matrix is discretised using a differentiable approximation of the step function. The encoder output is modulated according to a Binary Phased Shift Keying (BPSK) constellation mapping.

A simulated Additive White Gaussian Noise (AWGN) channel model is adopted, defined as an identity function and an addition with a vector of noise. This model has a single non-trainable parameter, the noise variance, and is thus differentiable with respect to the encoder parameters.

The demodulator output is fed directly from the channel and outputs Log Likelihood Ratios (LLRs) to the decoder according to the standard LLR formula for BPSK modulation [BM14].

The decoder is derived from the Belief Propagation (BP) algorithm [PS08]. This is an iterative decoding algorithm, based on message passing between variables nodes and parity check nodes of a Tanner Graph. The variable nodes correspond to LLR values of the individual bits of codes, while the control nodes correspond to parity check equations of the code. The proposed implementation, named Gated Neural BP (GNBP), is based on a Recurrent NN (RNN) model with a cell that executes a weighted sum product algorithm. Such weighting mechanism has shown to improve the decoding performance of short to medium sized codes when compared to standard BP decoders.

Parity-check and generator matrices of the code (H, G) used in this model are derived from a single set of real parameters W_g hosted by a model entity named bridge model. At each update of W_g, H and G are recomputed following a deterministic procedure. W_g is first quantified using a differentiable step function mapping negative parameters to 0s and positive parameters to 1s. The resulting parameter set is shaped into a k by $(n-k)$ matrix W. G is constructed by the concatenation of a k by k identity matrix with W and H by the concatenation of the transpose of W with a $(n-k)$ by $(n-k)$ identity matrix. Using this so-called standard form, we ensure that encoder and decoder are always matched so that the training procedure can "focus" on finding performing coding and decoding scheme (not trying to match a decoder to an always changing coding scheme).

Training and dataset:

As described above, without making assumptions on the model, training a coder-decoder system requires datasets that spans all possible input messages. For a $(63,36)$ code, i.e., 36 bits messages coded as 63 bits codewords, this represents roughly 70 billions different messages. Obviously, this is completely impractical. To train the proposed model, we exploit the symmetry and linearity properties of the model which allow training on a drastically reduced dataset: the base of messages, i.e., 36 linearly independent messages, e.g., one-hot vectors.

To the best of our knowledge, this is the first solution proposed to learn linear block codes without facing the curse of dimensionality. Additionally, the product operator used in both encoding and decoding phases is shown to become both strictly convex and linearly separable when considering this reduced dataset, which is a key requirement for training such model. Thus, the proposed model is both scalable and fully differentiable allowing for the joint training of a code and associated decoding scheme.

Performance:

As shown in Figure 2-10, the proposed model is shown to provide significant gain when compared to code from the literature such as Bose-Chaudhuri-Hocquenghem (BCH) codes associated with BP or Neural BP (NBP) decoders. The performance gain is expected to be coming from both the choice of a performing code and the use of jointly trained NBP-like decoder reducing the impact of potential cycles in the Tanner Graph of the code.

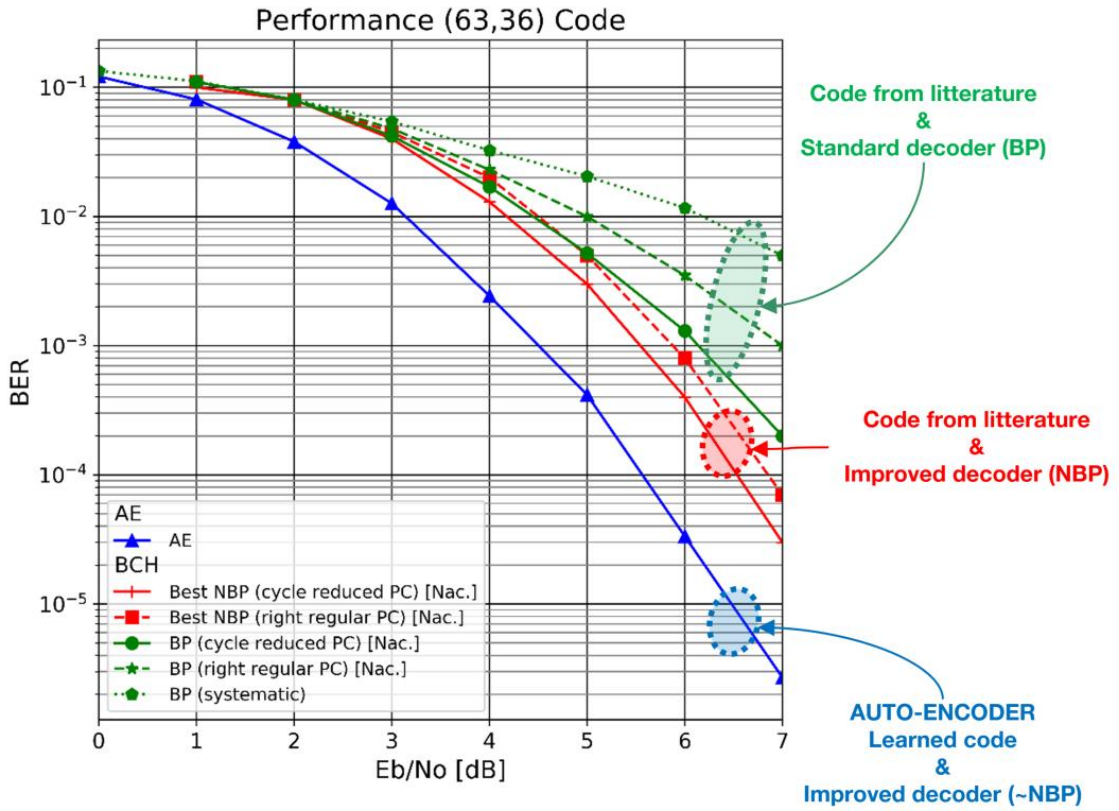


Figure 2-10: Initial results: promising coding schemes.

2.1.2.3 Deep learning for location-based beamforming

6G BSs are likely to comprise a great number of antennas, continuing the transition to actual massive MIMO systems that started with 5G. The choice of the used precoders is of paramount importance for massive MIMO systems, since it allows to focus energy on the right users in order to maximise their Signal to Interference plus Noise Ratio (SINR).

Determining appropriate precoders can be done in several ways. The channel can be estimated by means of the devices sending pilots, the estimate being then used by the BS to choose the precoder. Alternatively, the BS can test sequentially several beams and choose the one resulting in the highest received power at the user. Such a beam search strategy has been preferred for 5G. However, beam search can be very time consuming when using a large number of antennas at the BS, resulting in a large number of beams to test. Knowing a priori where users are located could greatly reduce the computational burden, since the device location is tightly linked to the optimal beam to use. Indeed, one could choose the precoder using only the location information, without requiring estimating the channel nor testing a high number of beams. This approach is known as Location-Based Beamforming (LBB). However, existing LBB methods [MDE10, YM15, KCT+16, AME18] assume the existence of a LoS path. In the NLoS case, the proposed methods do not perform well.

In order to overcome this limitation, we propose to train a deep NN to directly learn the location/precoder mapping. However, classical NNs are not well suited to such a task, because of their spectral bias [JGH18] which prevents them from learning functions containing high frequencies (whose output vary rapidly with respect to their input). This is explained by the fact that the dynamics of gradient descent leads to an error that is exponentially decreasing at a rate inversely proportional to frequency [TSM+20]. However, it is possible to use a special kind of

layer called Random Fourier Features (RFFs) in order to avoid this issue [TSM+20]. The proposed NN is, thus, structured in a particular way in order to be able to learn functions of high spatial frequency (see Figure 2-11), i.e., functions whose output (the precoder here) varies rapidly with respect to their input (the location here). A classical Multilayer Perceptron (MLP) using Rectified Linear Unit (ReLU) nonlinearities is used after an RFF layer. The input to the network is the location $\mathbf{l}$ of the user and its output $\mathbf{w}$ is the predicted precoder. The network is trained using a database comprising user locations and the corresponding channels: $\{\mathbf{l}_i, \mathbf{h}_i\}_{i=1}^{N_{train}}$, and a cost function expressed as $1 - \frac{|\mathbf{w}^H\mathbf{h}|^2}{\|\mathbf{h}\|^2}$ which measures the complement of the correlation between the precoder and the channel.

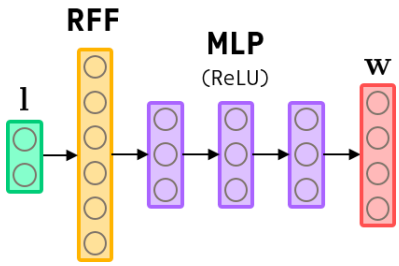


Figure 2-11: Proposed NN structure.

The proposed network has been assessed empirically on synthetic channels taken from the DeepMIMO database [A19]. A set of 10,000 channels are used for training data and obtained results (on test data not used for training) as shown on Figure 2-12, where it is compared in terms of correlation with classical LBB methods [MDE10, KCT+16]. As seen in Figure 2-12, the proposed method allows to obtain appropriate precoders for users everywhere in the considered area, whereas classical LBB methods allow to get good precoder only for users in LoS. Moreover, the proposed method is also compared (in terms of Cumulative Distribution Function - CDF of the correlation) with a classical deep learning approach without RFF (MLP). Results are shown on Figure 2-13, where it is seen that using an RFF layer allows to greatly improve the correlation performance, since much more locations exhibit a high correlation between channel and precoder (yellow areas on the figure), even in NLoS. These results are encouraging and show the potential benefit of using deep learning for LBB. More detailed explanations and a more thorough analysis of the obtained results is available in the associated paper [LYP+22].

In the future, it would be interesting to assess the proposed method on real measured channels, as well as in conjunction with a location estimation method (here the location has been assumed perfectly known). Moreover, the structure of the NN could be optimised considering the physics of signal propagation, in order to enhance the frugality of the proposed NN. The distribution of the RFFs as well as their form are potentially impacted.

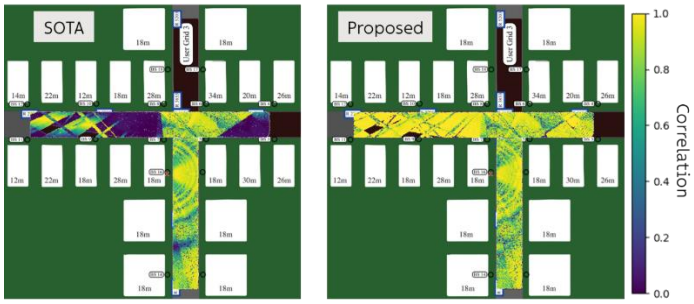


Figure 2-12: Comparison between a classical LBB method (left) and the proposed one (right) in terms of correlation between the precoder and the channel (the higher the better).

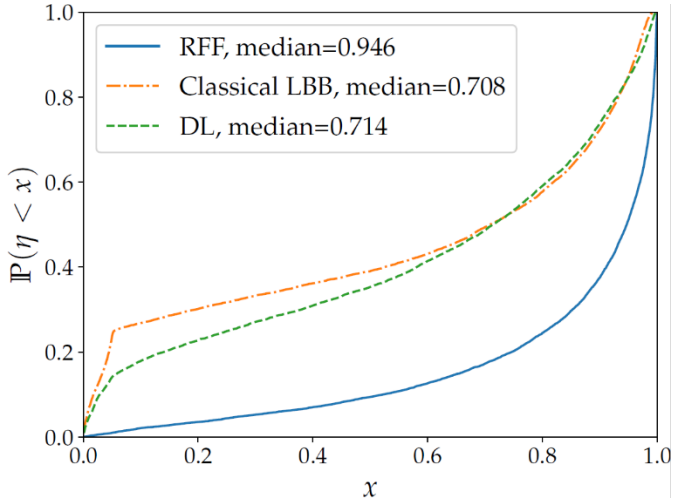


Figure 2-13: Comparison in terms of CDF of the correlation between the proposed method (RFF), classical LBB approaches and with a deep learning approach without RFF (DL).

2.1.2.4 Machine learning aided beam management

To compensate for the coverage issues in using mmWave and above bands, dense deployment of APs is suggested. However, it is not always necessary to maintain all the APs in a fully functional state to provide the required Quality of Service (QoS). Setting redundant APs to a sleep mode with low power consumption helps to improve the overall power consumption of the network. In coordinated multi-point serving scenarios, the involved APs are required to perform initial access procedures like the beam sweep using a common time-frequency resource set. Otherwise, beam search of an AP would interfere with the data transmissions of other APs. The turned on "sleep-state APs" would have to wait for the next shared beam-search opportunity to perform the beam sweep.

In such scenarios when these "sleeping-APs" are reanimated, they have to wait for the beginning of a common initial access period and this causes an additional delay component. However, users may require additional radio resources instantly to maintain reliability in high-throughput low-latency applications. The sleep-state APs should be able to contribute to the network as soon as they are "woken", i.e., when normal functionality is restored.

A Deep Contextual Bandit (DCB) based novel approach is proposed to perform instantaneous beam selection for a recently restored AP using some information from its neighbouring APs. With this approach, a newly woken AP can instantaneously start serving the users. The DCB model consists in each AP learning to characterise its environment through the beam choices made by the neighbouring (active) APs, and therefore, learning to map a given set of beam choices made by its neighbours to one of its beams. This beam information is shared among APs via backhaul links though a local processing and controlling hub called Central Processing Unit (CPU). For the example, as depicted in Figure 2-14, the model in the Sleeping AP (SAP) would predict the beam by taking b1 … b5 as the input.

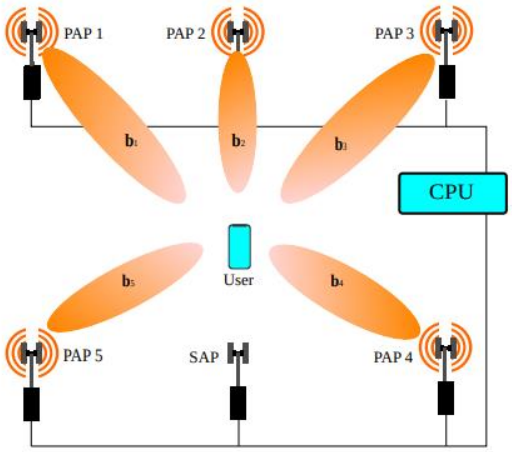


Figure 2-14: An example system model containing active "primary" APs (PAPs) and dormant SAPs.

The proposed approach was simulated in an indoor office environment. The simulation environment shown in Figure 2-15 is modelled using appropriate material properties in a commercial ray-tracing software. The green dots represent possible user locations and the red cubes denote the locations of six APs. The channel data used in the ML problem is generated using ray-tracing.

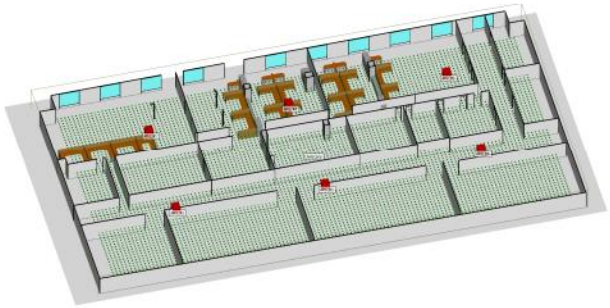


Figure 2-15: Modelled physical environment considered for the simulation studies.

The performance of the DCB model can be gauged by analysing regret which compares the action taken by the DCB agent against the optimal action in terms of resulted SNR. Therefore, a lower regret would imply better performance compared to a higher regret. The DCB model in a single SAP is trained in two scenarios which contains 4 and 5 PAPs, respectively. Figure 2-16 shows the evolution of the regret incurred by a DCB agent over the training episodes. The DCB model in the 5 PAP case performs better compared to the 4 PAP case with a regret measure around 4% compared to the 4 PAP case due to the more beam information available in the 5 PAP case.

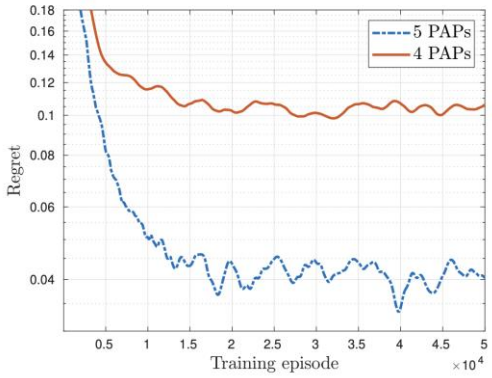


Figure 2-16: Regret incurred by the contextual bandit model during training.

2.1.3 Designs accounting for nonlinear distortion

NNs are well-suited for modeling nonlinear input-output relations, meaning that they can be readily applied for mitigating many types of hardware impairments. Therefore, ML-based solutions can be used to build 6G radio networks that are natively resilient against such impairments. Instead of designing the system to avoid hardware impairments, one can train the devices to operate under the distortions, which can result in lower device costs and higher energy efficiency. One prominent source of hardware impairments is the transmitter PA. This section describes two solutions for learning to operate under PA-induced nonlinear distortion.

2.1.3.1 TX-side CNN for reducing PA-induced out-of-band emissions

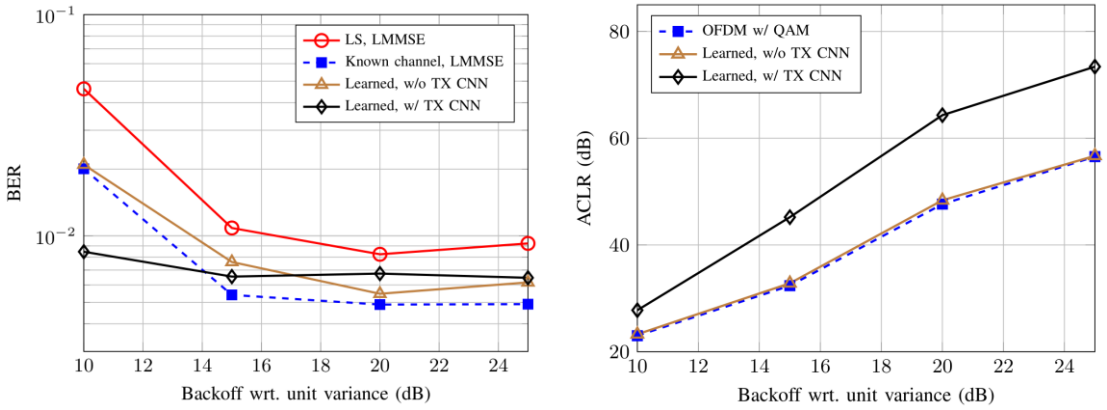
ML-based solutions are also well-suited for introducing resilience against hardware impairments. In a fully AI-native air interface, the transceivers can be trained to deal with hardware impairments, instead of avoiding them entirely. This represents a paradigm shift from the conventional thinking and can improve the power efficiency as well as reduce device manufacturing costs.

The E2E trained link in Figure 2-7 is a potential way in which ML can be used to add resilience against PA-induced nonlinear distortion. As already mentioned in Section 2.1.2.1, this is achieved by introducing a small CNN to the transmitter IFFT output, where the crucial aspect is that this CNN is operating on time-domain signals since this corresponds to the physical phenomena it is modelling [KHH+22]. Note that a relatively small CNN is considered here as it is expected that it is utilised primarily on the mobile terminal side. Altogether, the CNN has two layers and 32 parameters in total.

The weights of the CNN should be learned jointly with the other parts of the transmitter, as well as the receiver, by training the complete system in E2E fashion. In order to ensure that a generic scheme is learned, the training is done using several randomised PA responses, which prevents the transmitter from simply learning to pre-distort the waveform for an individual PA. Moreover, in addition to the typical approach of using the binary cross entropy as the loss function, now the out-of-band emissions at the PA output should also be incorporated to the loss term. This results in a transmission scheme which also minimises such emissions, in addition to the cross entropy.

To illustrate the benefits of such a learned transmitter, Figure 2-17(a) shows firstly the BER of the different schemes under a fixed Signal-to-Noise Ratio (SNR). It is evident that the learned system with the TX CNN achieves the lowest BER when the backoff is small (this is the most nonlinear operating point of the PA). However, when the nonlinearities are less severe, the benefit of the CNN diminishes, as expected. Figure 2-17(b) shows the Adjacent Channel Leakage Ratio

(ACLR) of the learned waveform, compared to a conventional QAM OFDM waveform, measured at the PA output. Here, ACLR is defined as the ratio of the power of the in-band desired signal to the power of the out-of-band emissions. The effect of the TX CNN is again clear, as it achieves clearly superior ACLR compared to the other schemes. Altogether, these findings show the benefits of ML-based processing when dealing with hardware impairments, contributing to the project target KPIs of BER gain and more efficient resource utilisation via facilitating more accurate detection of distorted signals and reducing the out-of-band emissions.

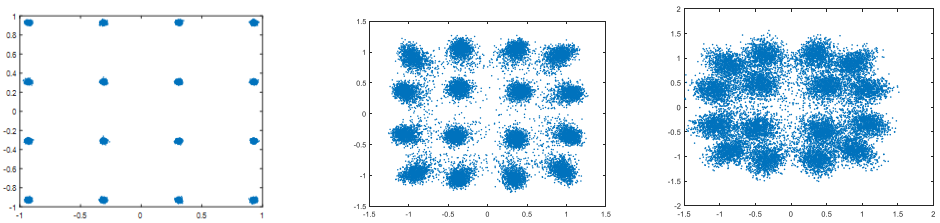


(a) (b)

Figure 2-17: (a) BER of the different schemes with respect to the PA input backoff when the SNR is fixed at 24 dB, and (b) ACLR of the different schemes with respect to the PA input backoff. Note that here the PA backoff is defined with respect to unit variance, i.e., it represents the power of the PA input signal (not to be confused with PA backoff with respect to 1dB compression point).

2.1.3.2 ML/AI empowered receiver for PA non-linearity compensation

With 6G the usage of higher carrier frequencies is envisaged, compared to the existing cellular technologies. Transmission at higher carrier frequencies provides new opportunities for larger spectrum allocations. However, transmission at higher frequencies is more challenging, due to the impact of hardware impairments, such as oscillator phase noise and PA nonlinearities, on the performance of communication systems, see Figure 2-18. In [FS20], it has been shown that an ML empowered receiver can compensate the impact of oscillator's phase noise. The design of PAs at these frequencies is challenging since, on the one hand, the energy efficiency of the PAs decreases as the carrier frequency increases. On the other hand, the output power of the PAs decays at high frequencies. Using ML techniques to model and compensate for the PA nonlinearities at the receiver, higher distortions can be tolerated, hence the PA output backoff can be reduced, leading to higher output power and more energy efficient operation of the transmitters.



(a) (b) (c)

Figure 2-18: Received equalised Discrete Fourier Transform (DFT)-s-OFDM modulated symbols subject to different levels of PA nonlinearities: (a) ideal PA (b) nonlinear PA with 2 dB back-off (c) nonlinear PA with 0.8 dB back-off.

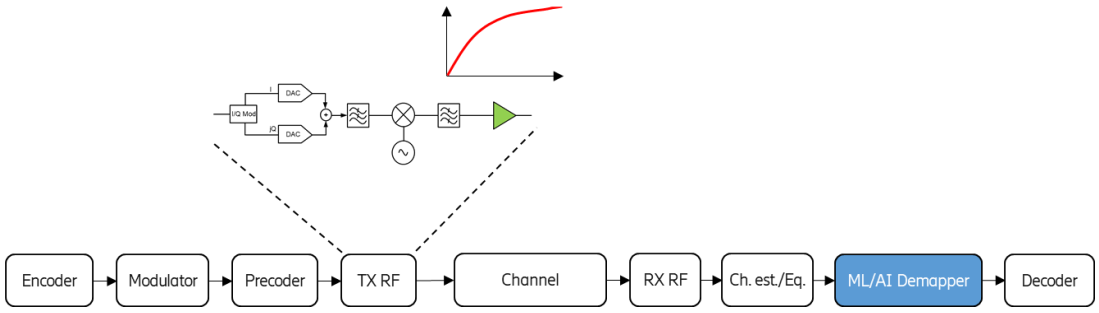


Figure 2-19: Schematic diagram of a radio transceiver with ML empowered receiver to compensate for PA nonlinearities.

A neural network(NN)-empowered receiver algorithm is developed for compensating transmitter PA nonlinearities using an ML demapper as outlined in Figure 2-19, where NN models are trained based on logged data from a link simulator. The ML demapper computes the soft bits to the decoder based on the equalised received symbols. The designed ML demapper is composed of a multi-layer fully connected NN, where the inputs are the real and imaginary parts of the equalised symbols and the estimated SNR, and the outputs are soft bits corresponding to each bit of a received symbol. The performance evaluations of the legacy demapper and the ML demapper subject to PA nonlinearities using BER performance metric based on link simulators is presented in Figure 2-20. The performance of the legacy demapper in the presence of linear PA is shown as an upperbound on the link performance.

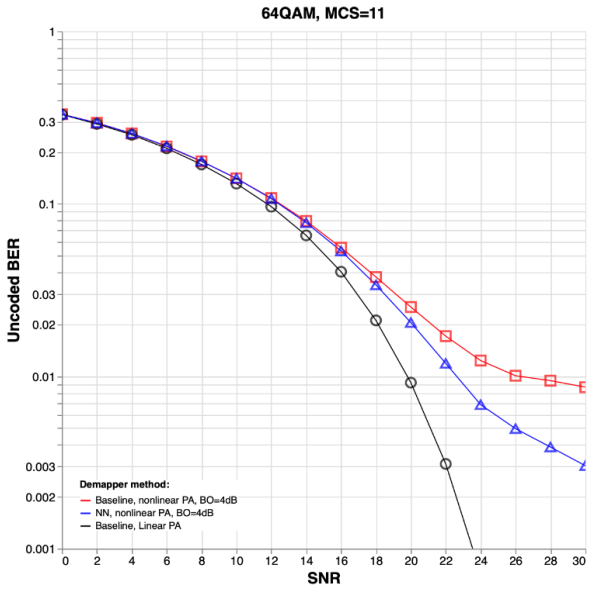


Figure 2-20: The bit error rate (BER) of links with legacy demapper and NN- demapper in the presence of linear PA and nonlinear PA with 4 dB back-off.

The NN-empowered receiver can enable more nonlinear PA operations, and hence can improve energy efficiency of the transmitters. This would help to operate 6G transmitters in more energy efficient operating point.

2.2 Improvements in E2E network operation & management

This subsection presents the AI/ML as a way forward to enhancing the network communication in 6G systems. It is presented predictive solutions regarding improvements on orchestration aspects and decentralised solution to improve the scaling of the architectural components.

2.2.1 AI/ML-based predictive orchestration

The last decade has seen a great rise of AI/ML, from Deepmind [YSH+20], virtual assistants (e.g., Alexa or Siri), self-driving vehicles (Tesla Autopilot [DC17] or comma's AI [SH16), to Sophia [HMG+12] or neural networks as YOLO (You Only Look Once) [RDG+15], or frameworks as Tensorflow [TF22] among many others.

The adoption of these AI/ML techniques is also impacting on the telecommunications scope [AIML21][OWY+21], and with the advent of 6G, the worthiness of AI/ML is expected to increase even more, evolving from connected things to connected intelligence [TRS22]. One of the areas on which AI/ML is expected to impact is the services Management and Orchestration (M&O), and more specifically, providing predictive capabilities that can be implemented by relying on AI/ML functions, as envisaged in the Hexa-X M&O architectural design [D6.2].

The usage of AI/ML functions can help to deal with the increased complexity of 6G networks [SHH22]; e.g., AI/ML techniques can help when, due the large number of involved variables, designing orchestration algorithms in the traditional way becomes extremely difficult. So, to support the regular M&O capabilities (e.g., fulfilment, assurance and artifacts management), management functions can make use of AI/ML techniques for managing, optimising and controlling the services deployed on the network, as well as to take decisions regarding actions to be performed at the infrastructure layer, network layer and service layer (e.g., scaling, placement, or services migration).

An example of this increase of complexity in 6G networks is the integration of the extreme-edge domain [PML+19][F220]. In this case the M&O system is envisaged to provide *continuum orchestration* [RCV+22][TS22][CZK21], i.e., the orchestration function will cover from the end devices domain (which can be massive in number) through the edge, until the central cloud, with all the related cloud nodes in between (see Figure 2-21).

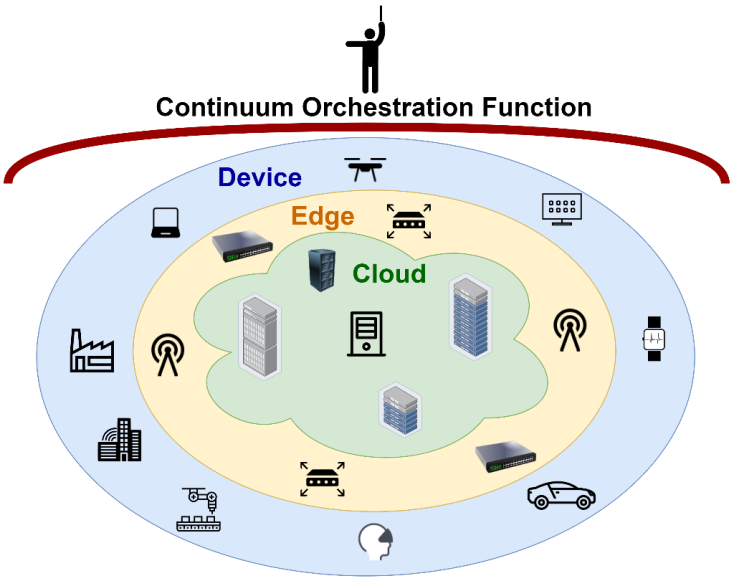


Figure 2-21: Continuum Device-Edge-cloud management orchestration.

The integration of the extreme edge is a challenge itself, since the environment encompasses a variety of extraordinary number of devices, whichy could be highly volatile and heterogeneous. In such highly complex context AI/ML can be used to process the big amount of data coming from different infrastructure sources (extreme edge or cloud) [YW22] and finding cross-relation hidden pattern, to trigger different orchestration actions.

In 6G there will be a need for an efficient algorithm capable of orchestrate the entire infrastructure with such high number of devices, as well as the large amount of generated data. There are two approaches to implement such algorithms, either reactive or proactive [LML14]. A reactive solution is when via threshold-based the system automatically reacts to the requested demands, while proactive solutions can rely on using AI/ML techniques to predict future demands in the system and to arrange resource provisioning in advance. The goal in a proactive solution is to start generating predictive insights that can flag potential issues before they become a problem, while a reactive solution will start solving the problem once it has already detected. In predictive orchestration, the capacity of providing additional resources before the actual demand makes avoiding the delay during resource allocation, and it is more efficient than reactive orchestration as shown in [ZLH+19].

6G needs to support mass deployments and mission critical interaction [MBM+20]. Automotive applications will depend on the 6G extreme performances (e.g., almost zero perceived latency [RDS21][CGY+22]), this is where AI/ML techniques applied to proactive orchestration will contribute to reduce service instantiation time, among others. In order to get this, AI/ML algorithms need to gather the useful data, and process and analyses them to predict future scenarios. Such prediction can be useful for reliability purposes when potential hazards could emerge from connectivity failure. For instance, in predictive orchestration, link redundancies, QoS handling, offloading, etc. can be better guaranteed if they can be predicted.

Predictions can be performed relying on the well-know AI/ML paradigms: supervised learning [STS16], unsupervised learning [Wit14] and reinforcement learning [DS21]. Commonly used AI/ML models for prediction are RNN (in general) or LSTM (a specific type of RNN) [She20]. The common approach is typically focused on time series forecasting [DS21], in order to make predictions based on historical data. Since predictions are based on past trends and patterns, by analysing complex historical data, proactive orchestration can be performed.

Advanced monitoring and analytics mechanisms are essential to perform predictive orchestration strategies. These mechanism should be enabled to access to both: infrastructure metrics and application data, in order to support decision making based on correlating data from these different domains. Combining such data can lead to finding a hidden patterns, e.g., in the users behaviours, which can be used to efficiently provision or allocate resources.

2.2.2 Distributed AI for automated UPF scaling in low-latency network slices

Current research trends in AI are focusing on the distribution of AI, moving inference models and/or model training from a centralised location in the cloud closer to where the data are generated, towards the edge of the network or even at the extreme edge. The final aim of this approach is to reduce the communication overhead, average latency and, thus, mitigate the network congestion and, in some cases, the energy consumption. In [ZCL19], a survey that investigates the different edge, hybrid and cloud solutions has been conducted. In particular, some aspects have been evaluated: the reduction of communication overhead, the accuracy of the ML solution, the convergence time and so on, underlying the advantages and disadvantages for each solution. In some specific cases, the communication was reduced up to the 87%. For instance, having different Multi-access Edge Computing (MEC) locations where the model is

only inferred, would save a certain amount of bandwidth and thus energy, because data is not sent to the central entity for the inferring process. However, this research trend is omnipresent in the B5G/6G network, where a plethora of mechanisms like Federated Learning (FL) will be part of the communication generation.

In the context of the B5G/6G network, this architectural enhancement is a potential technical enabler to improve the efficiency and the level of automation for orchestrating the B5G/6G network slices and virtual functions in use cases requiring low E2E latency. For example, taking into consideration an edge application, where different UEs are moving with different patterns to different locations, becomes a challenge to estimate a priori the UE(s) geographical position. This could cause a fluctuation of the network traffic with the consequences of a fluctuation of the usage of resources in a given edge location.

The standardisation bodies are focusing towards this direction as well, adopting either centralised or more distributed solutions for data analytics functions, with the aim of enhancing network automation and data-driven closed-loop control in the next generation communication networks. In particular, 3GPP has specified in Release 17 - among several architectural enhancements - a distributed version of the Network Data Analytics Function (NWDAF) [TS23.288], the core network function capable of collecting and elaborating monitoring information from different sources, i.e., the NFs, the Operation, Administration and Management (OAM) system or even external sources, for data analytics purposes. Such pieces of information can be related, for example, to specific UEs, (mobility, communication pattern, etc.), NFs, network slices, or the network as a whole. From an architectural perspective, as depicted in Figure 2-22 multiple distributed and geo-localised entities of NWDAF could be deployed at the Edge Compute Nodes to collect and analyse data related to specific geographical areas, while a centralised NWDAF instance can receive only aggregated data from the distributed entities

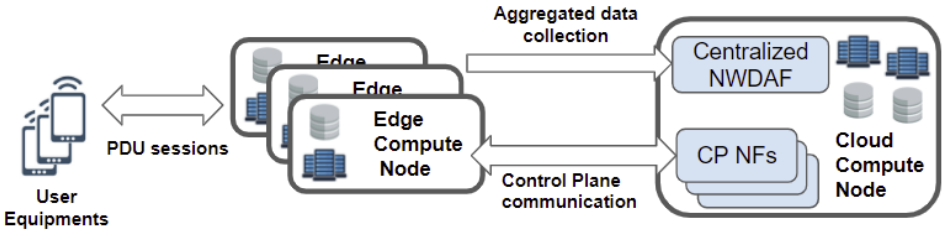


Figure 2-22: Architectural diagram of NWDAF data collection and aggregation.

Therefore, each distributed NWDAF, serving a specific geographical area, dramatically reduces the data communication and latency overhead, as well as the processing load, with respect to the fully centralised solution.

In this context, we propose an application of the distributed analytics concept, as provided by multiple collaborating NWDAFs. In particular, the proposed solution adopts AI techniques to trigger the pre-emptive auto-scaling of local UPFs, placed at the network edge in support of low latency communication services. A single UPF instance can handle multiple Protocol Data Unit (PDU) sessions, however the resources of a UPF instance are finite. As traffic load increases, to avoid degradations in service caused by finite resources, more UPF instances can be deployed and started, and likewise, an idle UPF instance can be terminated when the traffic is low.

Optimal auto-scaling is an excellent candidate for AI application, in particular a distributed approach which moves away from today's highly structured, controlled, and centralised architecture to a more flexible, adaptive, and distributed network of devices. A distributed NWDAF instance can be deployed in a strategic location with the ability to autonomously monitor

the network in the target geographical area. In parallel, an AI agent, also deployed in distributed edge computing resources, will consume and process the data from the NWDAF, apply the AI algorithm and models, and control the respective UPF scaling as depicted in the Figure 2-23. In general, within the Edge Compute Node the local NWDAF collects data from the UPF and it exposes to a monitoring entity. This entity not only collects the data from NWDAF, but also from other sources if any. Then, after a pre-processing, an AI agent performs a decision about the pre-emptive auto-scaling operation on UPF itself (Figure 2-23).

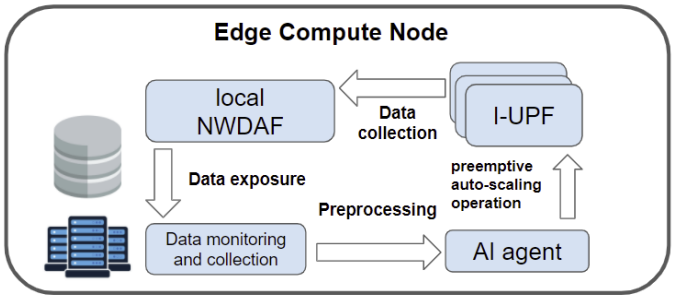


Figure 2-23: Closed-loop pre-emptive auto-scaling of UPF within Edge Compute Node.

To use such models, UPF load information available from the NWDAF, including CPU, memory, and disk usage, can be supplemented with user plane data like bandwidth, latency, packet loss, etc., as well as UE-related information (mobility, position, etc.) to get accurate predictions of future network conditions. In the distributed NWDAF, including user plane data into predictive analysis model training is expected to greatly increase the achievable accuracy of actual network conditions, however it raises several issues including security, privacy and data flow issues that would occur if implementing a standard centralised optimisation model. The proposed solution aims to validate the reduction of the orchestration time, affecting the accuracy and latency of inferencing process. However, these two KPIs could have affect to the network energy efficiency, depending on the number and type of edge compute nodes.

3 6G network as an efficient AI platform

The main goal of the research presented this chapter is to enable and enhance the global operation of AI services over the radio network, considering computing capabilities as a native part of future networks. The solutions presented address most of the KPI relevant for AI-driven communication & computation co-design. In Section 3.1, we describe methods to increase the availability of ML inferencing at the UE by improved mobility solutions, to increase reliability by accounting for low-quality connections, and to increase overall availability and reliability. Section 3.2 focuses on scalability for the anticipated applications that require a dense deployment of user devices and network equipment and proposes solutions for the demanding communication of AI agents distributed over massive deployments. Finally, communication and compute resource allocation to orchestrate 6G systems (e.g., power, bandwidth) with a connect-compute perspective is the topic of Section 3.3.

3.1 Seamless and pervasive in-network AI operation

In this section, we present results that enable and enhance the global operation of AI services over the radio network. By putting the UE in the focus, we show how a UE carrying a local ML model can seamlessly exploit the knowledge of large parts of the network, for example for inferencing tasks (Section 3.1.1), how the UE can prepare for offline operation in an anticipated network impairment case (Section 3.1.2), and finally how to implement distributed learning schemes where computation-intensive training is processed in the cloud and only small adjustments are handled at the edge devices (Section 3.1.3).

The results in this section address the following target Key Performance Indicators (KPIs): AI agent availability, reliability, and latency for resilient communication and compute network services for distributed AI applications in large scales, network and UE energy reduction by workload offloading and learning/inferencing task delegations, as well as improved inferencing accuracy.

By the results in this Section, we can increase the availability of ML inferencing at the UE by improved mobility solutions, increase reliability by accounting for low-quality connections and allowing AI agents to hand over their local information before an anticipated connectivity impairment, increasing overall availability and reliability. We can also mitigate latency issues due to radio handovers by association to different AI agents in mobility events.

Better signal processing models at the UE level improve spectral efficiency and, thus, network efficiency. These improvements are balanced by an increased computational cost due to model learning, resulting in a major trade-off. However, this additional computational burden can be reduced by using efficient NN hardware and the sharing of hardware at data centres and/or BSs. We also support more accurate signal processing models at the UE side for a better spectral efficiency.

3.1.1 AI-as-a-Service - enabling a UE to seamlessly exploit network knowledge

This section addresses the problem of how to enable a UE carrying a local ML model (obtained by e.g., training a NN) seamlessly exploit the knowledge of large parts of the network, useful to its locally undertaken inferencing tasks, by attaching to/ detaching from different learning systems across multiple operator areas. By "learning system" we refer to any structure involving one or

multiple AI agents deployed across a network coverage area (by a single or multiple network operators). Examples are centralised learning, FL, transfer learning, and distributed learning.

To the best of our knowledge, state-of-the-art in-network learning implementations prevent seamless exploitation of network knowledge by a UE, due to several challenges, both related to the communication medium particularity and to the communication and compute resource availability, as documented e.g., in [CGH+21], [LYX+20]. In summary, these in-network learning system operation issues are (with emphasis on FL as an exemplary learning structure):

- Lack of robustness to mobility events, e.g., UE mobility, but also mobility of the coverage-providing mobile network entity, e.g., an Unmanned Aerial Vehicle (UAV) with an AP mounted to it.
- Possible unexpected temporary or long-term learning system pause, due to the unavailability of an AI agent (e.g., a FL aggregator). This may occur due to a detected security attack, for example a data poisoning attack, compromise/ hijack of the entity hosting the aggregator (such as an edge cloud server) or simply hardware/ software errors.
- UEs/ members of the learning system experiencing low-quality connectivity to the network node collocated with the FL aggregator. Such low-quality connectivity may occur due to resource contention or due to e.g., deep signal fades experienced during the (iterative) FL model training period.
- Increased model training (or, aggregation, in case of a FL setup) latency, due to the lack of available compute resources at the AI agent side, or, in the existence of learning "strugglers" in case of FL, i.e., UEs of low capability being unable to produce their local model updates in a timely fashion.

To address such issues, the proposed solution is based on the availability of an AI Information Service (AIS) and its corresponding AI API, implemented over a network interface, which is interoperable across multi-operator network deployments. The AIS and its API will enable a user via a user interface or a client application to request a parameterisation of the device's locally available learning model from the network, given a number of performance requirements set by the user, the client/server application or a UE profile. Section 6.3.2 of [D5.1] explains a signalling flow for requesting and delivering a ML model satisfying AI agent selection criteria provided by an AI consumer (e.g., UE). As it can be seen there, apart from the AIS consumer, the involved entities are the ones of the AI repository function and the AI policy enforcer, as defined in [D4.1].

From a methodology/ algorithm perspective, the design goal is to route an inferencing task to the most relevant and available AI agent with a maximum tolerable E2E latency and energy consumption level. To achieve this, two solution components relate to: (i) user device (or machine, such as a robot) association to a radio node providing coverage, this radio node being in proximity to an AI agent hosting an ML model relevant to the inferencing task requested by the end user (or machine) (ii) matching the *metadata* of the AI agent revealing the purpose of the hosted ML model to the ones of the task; the degree of the similarity of the response to the request parameters is critical to the success (in terms of user requirements translated to demanded inferencing accuracy, consumed energy and end-to-end latency) of the inferencing query.

With respect to an exemplary scenario, where the requesting user device/ machine can simultaneously connect to multiple network nodes carrying AI agents, instead of a single network attachment point for requesting a specific model, each of a multitude of network nodes may have its fully trained model available within an AI agent. In an Intelligent Transport System (ITS) context, this may, for example, correspond to neighbouring vehicles deriving models based on their local observations and possibly combined with information from other sources (e.g., other

vehicles, the network, edge nodes, road side units, etc.). In this case, the data structure describing the task and model (e.g., NN) I/O features and characteristics (filtering criteria) which aims to be sent as part of the message body of a request to the AIS is the following: i) description of a use case; ii) required input features to the trained (NN, etc.) ML model in the UE; iii) required output features to the (NN, etc.) ML model in the UE; iv) required characteristics of the (NN, etc.) ML model in the UE (e.g., number of NN layers, NN nodes per layer, number of NN inputs, number of outputs, size per input/ output data point - in bits, maximum latency, etc).

This data structure (AI agent selection filtering criteria) may be announced, for example through a broadcast or multicast connection to neighbouring nodes, to any suitable recipient within range in case of non-availability of the AIS. The respective signalling framework appears in Figure 3-1 below.

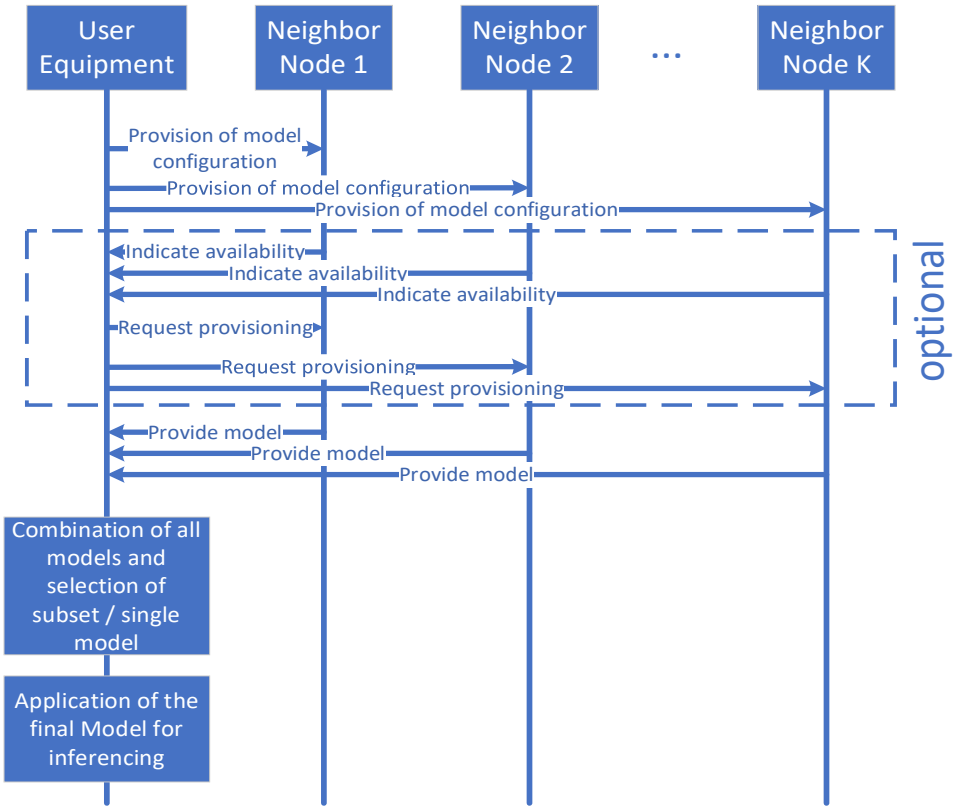


Figure 3-1: Signalling flow for requesting/delivering of new relevant ML models from a multitude of nodes per some filtering criteria.

The proposed solution is applicable to safety and dependability-critical environments (automotive, industrial automation and others). KPIs of specific interest are: flexibility, mobility support, AI agent availability, AI agent reliability.

3.1.2 Network impairment resilience of autonomous agents

For critical applications, it is important to increase resilience towards connection problems on the AI agent side. Abnormal bearer session release (i.e., bearer session drop) in cellular telecommunication networks may seriously impact the QoS of mobile users. Even worse, autonomous devices, vehicles and robots can critically depend on radio communication and it can be necessary to prepare edge devices for periods with reduced connectivity.

The latest ML technologies based on high granularity, real-time reporting of all conditions of individual sessions, give rise to data analytics methods to predict quality issues ahead of time.

Connection quality can drop due to fading phenomena or due to congestion events at higher layers. It would be of interest for a predictive ML-based time series analysis solution to be able to differentiate the root cause of these QoS drops so that appropriate counter measures can be taken in each case, in connection to the explanations of these predictions as we describe in Section 5.2.2.

For example, time series analysis combined with ML can be used to predict session drops well before the end of session. Towards this end, we collect labelled training data from developer mobile applications and build predictive ML models by using the data rate and network measurement time series.

In our preliminary experiments, we trained our models using tree ensembles, logistic regression and support vector machines. All these models share the favourable property that the model consists of a small set of vectors or constants and the prediction function of a single event has minimal compute cost. These models can be implemented on a mobile device with minimum power consumption. Key new element in our solution is the use of Fisher kernels over the dynamic time warping distance of time series [DVB15]. In this method, the model consists of a set of 50-200 sample events. The prediction is computed by an appropriate, normalised linear model of the distance over the individual time series.

We use a data set that contains recordings of personally owned smartphones with 565 GB raw data in 2383 log files, 6090 hours logged (254 days) of:

- detailed sensor data (most of the data): accelerometer, heat etc.;
- phone and network provider properties: Android version, phone type etc.;
- locations when available;
- cellular network data: signal strength, IDs, network events;
- network data: wifi and mobile; signal strength, IDs etc.;
- implicit network data: ping time, packet loss etc.;
- target features: call events, user call drop labels, high level network drop events.

We consider all available logs of frame error rate, signal power, handover, Transmission Control Protocol (TCP)/ Internet Protocol (IP) message and service packet information, and even Global Positioning System (GPS), gyroscope. We log events of call drop and lost network connection. For modelling, we also collect normal events, e.g., calls normally terminated by user or partner.

To train our call drop models we get the last 30 seconds of the calls. We assign “dropped call” labels to the last x seconds of the dropped calls. The variable x is 5 seconds in the last model, but we tried other settings too. Every second in the interval is a training/ testing data point, with the target variable “dropped” or “no problem”, and with lots of features reflecting signal strength, proximity etc. up to that time.

The variable x is to be considered as how much time the model looks ahead in time, answering questions like “according to the data measured up to this point, is a call drop to be expected within the following x seconds?”.

We also drop the last y seconds of the time series, because they correlate with the target variable too much (e.g., users move away the phone from their ears when experiencing sound problems, resulting in a huge change in proximity or light sensor values - these are to be considered as the direct effects of a call drop, and not related to the root cause).

Figure 3-2 shows how the selected measure of accuracy, the Area Under The curve (AUC) of the best model changes as the function of the time before the end of the session if we vary how much we use data from the very end of the session. We drafted three types of models, using only signal strength features, using all the available features, and using everything except the signal strength.

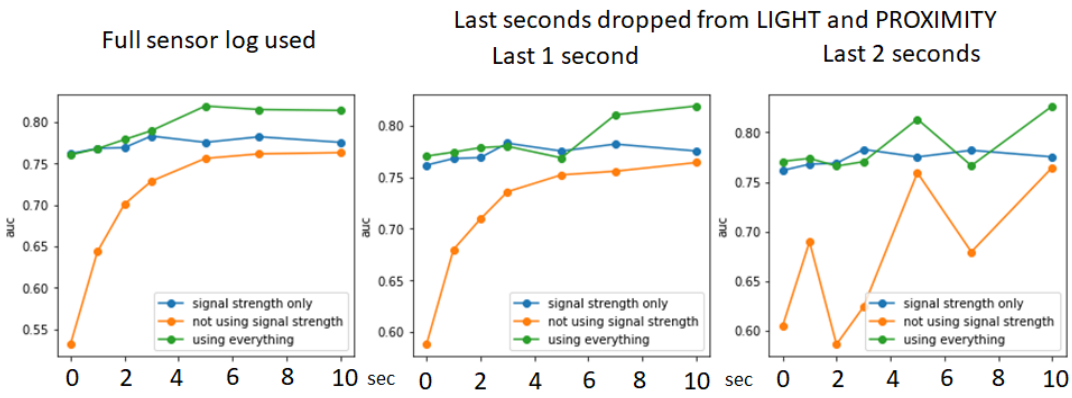


Figure 3-2: Experimental results for predicting dropped calls, using features for signal strength and other smartphone sensors. Horizontal axis shows prediction time before the actual drop, in ms, while the vertical axis shows prediction quality in term of AUC. In the three graphs, from left to right, we cut the last 0, 1000, and 2000ms of the data at the end of the sessions, since in the last seconds, user interaction may already be the result of the perceived connection loss. Correspondingly, the contribution of the sensor information gets lower as we cut the end of the session.

In general, accuracy of the model (AUC) becomes slightly better for larger time periods ahead, up to 5 seconds. Dropping misleading sensor values also helps.

The methods above use discrete time points to train and test a model. We are also experimenting with methods using the time series as a whole, checking e.g., the similarities between the signal strength curves measured and the curves of the dropped calls, but these are not yet discussed here.

3.1.3 Distributed low-complexity model learning

Several IoT use cases, such as metering for smart cities or smart sensors for Industry 4.0, require the use of many autonomous battery-powered devices, which need to be both low-power (battery life of at least 10 years to avoid repeated replacement) and low-cost (around $1-10) to be widely adopted. Such constraints imply the use of low-power and cheap hardware. Often, those same devices face challenging propagation environments, e.g., water meters deployed in pits, parking sensors in underground premises, inventory (depth) sensors in warehouses, etc. For such devices, running pre-trained models – or inferencing – seems realistic, thanks to the availability of low-power Neural Processing Units (NPUs) and current research on low-complexity models [LDL+21, NML+18]. However, extensively training those models on the same hardware is likely unfeasible or too energy consuming. Distributed learning schemes where computation-intensive training is processed in the cloud and only small adjustments are handled at the edge devices are therefore of interest.

In this work, we explore a scheme where a constrained UE device is assisted by computation resources hosted by the network to perform a complex signal processing task using neural networks. As shown in Figure 3-3, we further consider the task of equalizing an OFDM modulated frame using paired models and transfer learning techniques. In this approach, the constrained device hosts 3 NNs M_c, M_s and M_t that contribute to the channel equalisation task: model M_c is assumed to be a complex equalisation model that, once trained, captures a broad spectrum of channel and hardware impairments while model M_t is a simpler model, e.g. Zero Forcing Equalizer, that capture simple channel dynamics. The rationale of using two models of different complexity lies in the assumption that the underlying phenomenon we are trying to model has dynamics of different order of magnitude, e.g., we expect power amplifiers non-linearities or IQ channel imbalance to vary little while the phase of propagation paths is random. We envision the M_s model to capture such fast dynamics while M_c is also capable to capture slower dynamics, at the expense of an increased complexity. Finally, M_t is a transfer model that parameterises parts

of M_c using M_S's output after training. Different strategies are considered to convert the result of learning the simpler model and its outputs into a parameterisation of the more complex model, including a direct reparameterization of M_C using the output of M_t and the weight-freezing strategy [SM-01], where model M_t selects a subset of M_C parameters eligible for retraining.

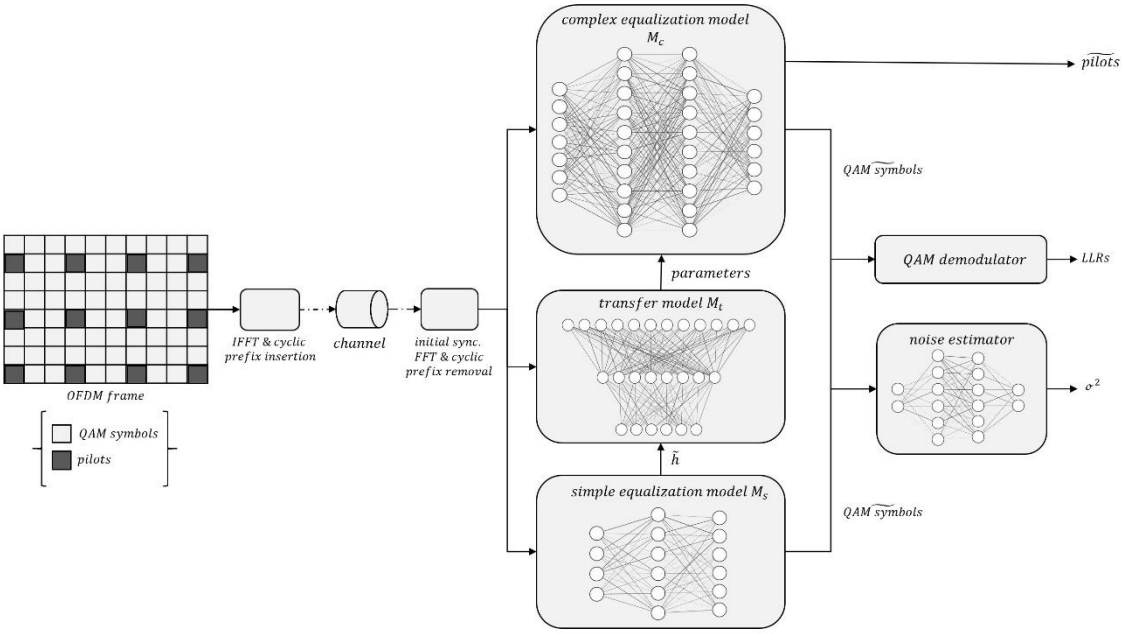


Figure 3-3: Overview: channel estimation with transfer learning.

One common aspect in the proposed approaches lies in the distribution and scheduling of the learning tasks and the retraining policy. As illustrated in Figure 3-4, when the pilot signals are first received, the UE trains and executes M_s to perform initial communications and transmit received pilot signals and a subset of data to a computation resource in the cloud. This computation resource oversees the initial training of the complex model M_c and of the training of the transfer model M_t using the channel estimate produced by the simple model. Once the parameters of the models are learned, those are sent back to the UE for exploitation.

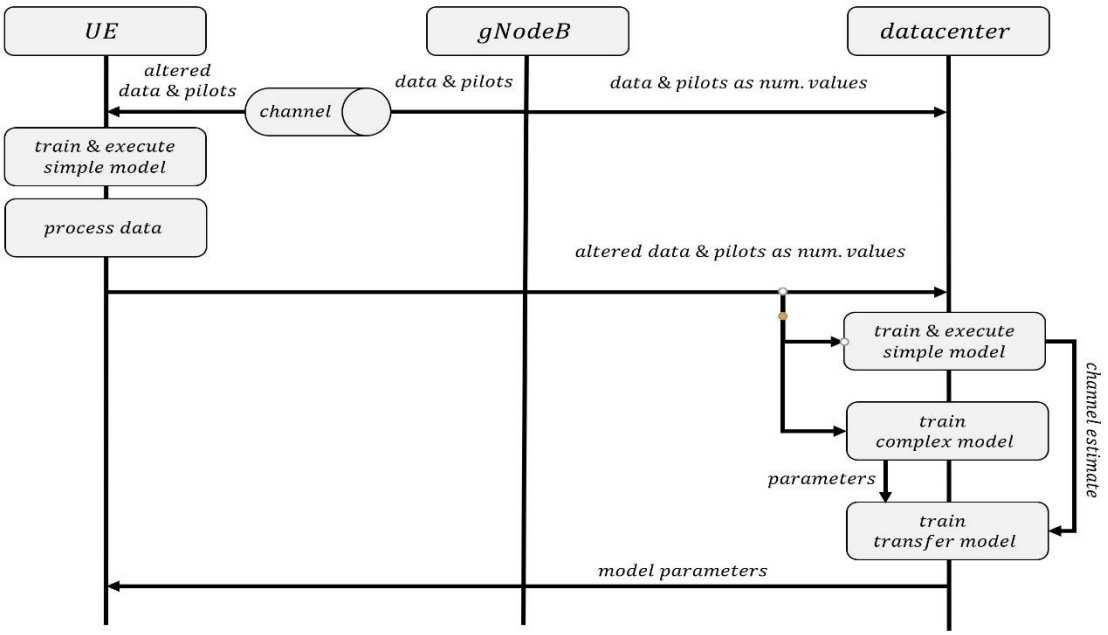


Figure 3-4: Delegation of the training task to cloud-based computation resources.

As shown in Figure 3-5, following this initial phase, the UE initially uses (trains & executes) M_s upon receiving frames. If the processing of the data reveals errors in the frame, e.g. invalid CRC or abnormal noise estimate at the demodulator, the UE may execute the transfer model M_t to parameterize M_c and attempt to better equalise the received frame. Subsequent errors in the processing may trigger a request from the UE to a computation resource for the retraining of M_t and M_c, as illustrated in the initial phase.

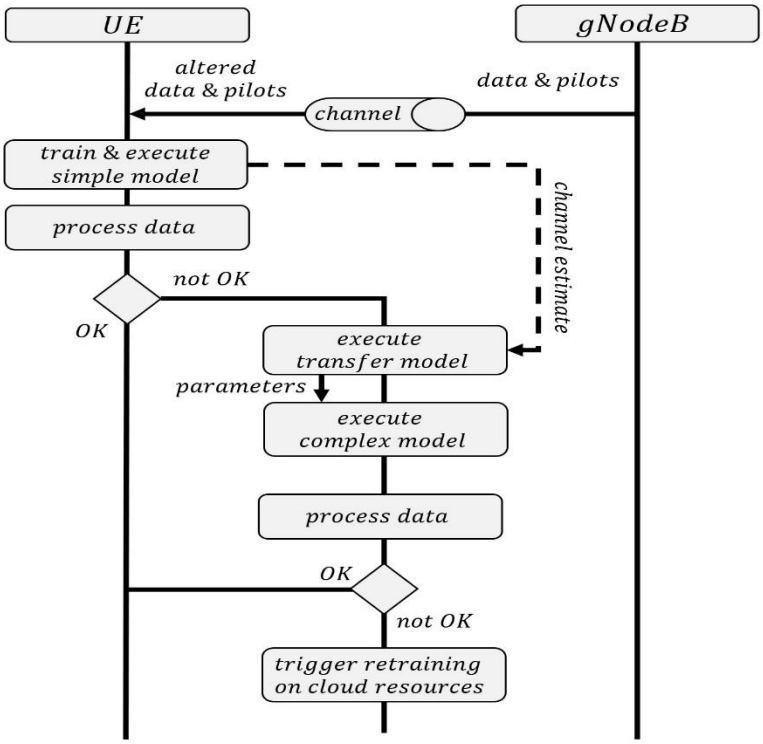


Figure 3-5: Model exploitation & retraining policy.

The objective of this distributed learning scheme is to improve channel equalisation performance by allowing constrained devices to exploit complex models, whose training is unfeasible due to computation or energy consumption limitations. By allowing a local update of the complex equalisation parameters through transfer learning, the approach balances the channel equalisation performance with the cost of transmitting data and parameters to and from in-network computation resources.

3.2 Scalable solutions for distributed AI applications

Many of the use cases envisioned in 6G networks require dense deployment of user devices and network equipment for coverage, communication reliability or other reasons. The communication of AI agents distributed over these massive deployments may be demanding on the network in various ways. This section introduces several aspects of handling large population of AI agents depending on the actual task. In Section 3.2.1 the problem of straggler mitigation is addressed in FL architectures, where data is naturally distributed among the wireless devices, and a joint model is trained collaboratively by sharing gradients or weights. In large FL systems both data sharing requirements and communication capabilities can be highly uneven, which can slow down the training process. In Section 3.2.2, the deployments of Spiking Neural Networks (SNNs) and

neuromorphic architectures are analysed, as, in general can be very efficient for mass deployment of distributed AI due to inherent sparsity of communication. Section 3.2.3 considers an in-network Radio Access Network (RAN) AI application of multi-cell multi-user MIMO, where the beamforming for large number of users must be solved with limited observability (centralised training and distributed inference) as an enabler for scalable application of the algorithm.

3.2.1 Federated ML model load balancing at the edge

Given a large number of heterogeneous sensors connected to FL nodes at the edge, the goal is to provide a low latency and high quality distributed ML service. In the use case scenario of Figure 3-6, a variety of sensors are connected to AI compute nodes through radio BSs. For accurate and low latency operation, each AI node needs access to sensors of most types, and the connection load needs to be balanced. Based on the Timing Advance (TA) information, the connection of sensors to AI nodes can be reconfigured; however, in addition to handover costs, the state of a sensor may also need to be migrated to the new AI node. In our experiment, we propose a method to dynamically rearrange the connection structure.

In our proposed dynamic reconnection solution, our goal is to provide load balancing to remedy potential hot spots and data type diversity to ensure quality balance for the federated learners. Load and diversity balance is necessary to make sure each node can equally contribute to the FL task dynamic load rebalancing by reconnecting sensors to nodes in the radio network if (i) load is uneven or (ii) some nodes receive insufficient variety of data for serving local models, for example when certain crucial type of sensor is not connected to a FL node.

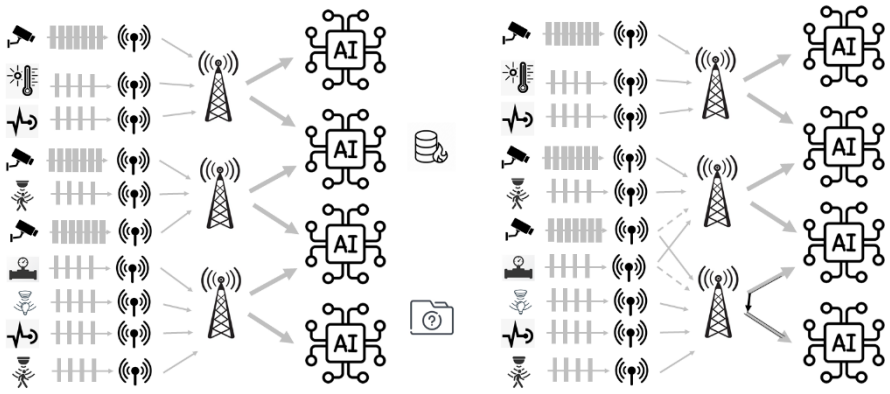


Figure 3-6: Left: a FL hot spot with too much data (top) and an insufficient data with one type (camera) missing (bottom). Right: a reconnection decision causes state migration between the bottom AI agents.

When processing mobile sensor streams with highly skewed and nonstationary data distributions, data processing nodes might receive imbalanced load both regarding amount and type of sensors. Two negative impacts of imbalanced connection patterns are slow tasks that delay the completion of the whole stage of computation, and uneven knowledge of the environment resulting on suboptimal model learning and prediction.

To mitigate imbalanced connection patterns, an adaptive, on-the-fly reconnection algorithm is proposed that continuously recomputes improved connections, given the observed distribution, by taking both radio physical constraints and state (e.g., metadata, past aggregated values, features computed in sliding windows, etc.) migration costs into account. We note that trade-off between cost of strugglers versus state migration is key in our proposed solution.

Exact solution to load balancing is a bin packing problem, hence the exact solution is computationally prohibitive.

We propose a dynamic reconnection method to associate sensors to AI nodes, by a heuristic load balancing solution, which includes a new histogram counting function for the federated AI components. Our solution for a simplified setup is descried in [ZSB+21] based on [G14], who formalised and developed partitioning functions for stateful operators based on a combination of consistent and explicit hashing.

For improving balance with minimal necessary migration, our method is a heuristic combination of an explicit partitioner for top imbalance and a random partitioner for connecting the remaining sensors producing roughly the same load. We observe imbalance when either of the following conditions hold, with experimentally selected values of the load factors c_1 and c_2:

- Sensor load exceeding c_1 times the average load, measured sensor data load in the vicinity of a given node (data hot spot);
- Less than c_2 times the average number of a given sensor type in the vicinity of a given node (data type imbalance).

Whenever either of the two thresholds are violated, a reconnection action is initiated.

For a reconnection decision, we also try to make minimal modifications to the previous partitioner to reduce migration costs. We reuse the previous connections and rerouting only those sensors that would contribute to imbalance. We select an imbalance threshold that triggers the reconnection decision. Our method to update an existing connection pattern and prepare for a potential reconnection is as follows. First, we select a brute force solution for the imbalanced nodes. To do so, we first compute the random location for each affected sensor, which would be their default connection without imbalance, so that we can reduce the amount of future reconnections. Next, we evaluate the balance if we keep all other connections intact. Finally, we use the random connection function to rearrange the remaining sensors. For computing the random connection partners, we use consistent hashing [KR06]. The main goal is to thoroughly evaluate the possible selection of the heuristic parameters in our solution.

In the initial experiment, we use generated Zipfian data, with some data sets including real distributions. We take 4M randomly generated four-character words, and a 4M element parametrized Zipfian datasets of 100K distinct items, with an exponent between 1–3. As a real Zipfian distribution data, we take 4M tags of LastFM music listening records. For testing the behaviour of the repartitioning algorithm, we consider the data in original order, and also sorted by tag frequency in increasing, and in decreasing order. We also generate concept drift in the form of changing the underlying distribution added, by concatenating multiple sequences ot the above types.

We measured and compared the running time, load balance, and state migration cost of our partitioner to the Uniform Hash Partitioner (Hash), to our implementation of partitioning methods Readj, Redist, and Scan from [G14], and to partitioning strategy Mixed from [F+17]. Experimental evaluation based on simulation data will be added in D4.3.

3.2.2 Scalable and resilient deployment of distributed AI

Some AI architectures are well positioned to exploit the massive data distributed over wireless devices, but their design must be harmonised jointly with 6G network features to unlock their full potential. Biologically inspired NNs, and, in particular, SNNs, are often mentioned as the next generation of AI systems. SNNs typically adhere to neuromorphic computing principles, thus, are thought to inheriting advantages, such as fast computation and power efficiency. When used in conjunction with neuromorphic hardware, SNNs become particularly well suited for edge AI

systems even in battery-powered IoT devices. On the other hand, they pose radically different requirements on the communication infrastructure, which raises research questions on the characteristics and special requirements of AI traffic and their impact to communication network design.

SNNs mimic the operation of biological neurons and their spike-based communication: in these NNs, all the information is encoded and communicated among the neurons as asynchronous binary signals (spikes) by leveraging the timing of discrete events. Examples for devices generating spike type of data are neuromorphic or event camera [RRK+19], [I20], artificial cochlea, skin [LTY+19], touch sensors or Neuralink's brain-machine interface directly generating spikes as output signal. The increasing availability of such neuromorphic devices interconnected by next generation wireless networks will enable building large scale distributed NNs, which we refer to as Distributed Wireless SNN (DW-SNN). DW-SNN could help in realising the 6G vision of interactions among the human world (senses, bodies, intelligence, and values), the digital world (information, communication and computing) and the physical world (objects, organisms).

SNNs exhibit several properties that make it an appealing computational framework to be considered in a wireless environment. Highly power efficient operation is a significant advantage, which translates to low computing requirements in constrained devices. Neural activations are known to be sparse in SNNs. Sparseness in spike communication leads to smaller bandwidth requirements, and further energy efficiency by decreasing the time of radio transmissions. Some of the feasibility aspects of a DW-SNN architecture will be analysed, with the aim to explore and showcase the potential advantages of the joint design of SNN and the communication network. The inherent resilience of SNNs to spike transmissions losses opens the opportunity to relax the communication quality requirements, and eventually, design AI-native radio transmission techniques based on spike communication.

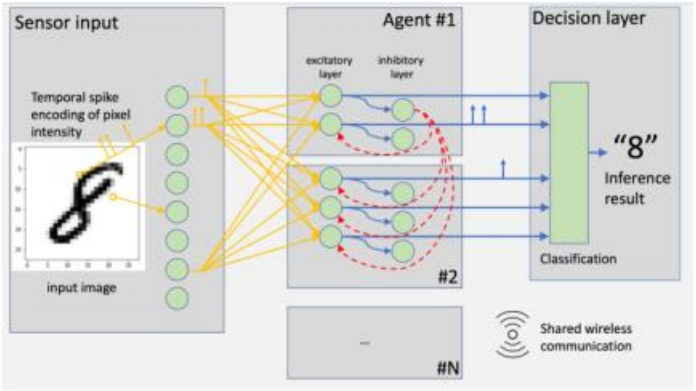


Figure 3-7: High level architecture of the investigated SNN system.

We investigate the various phenomena arising in DW-SNNs by selecting a spike-based implementation of the well-known MNIST hand-written digit recognition problem. The architecture of the investigated SNN MNIST system is depicted in Figure 3-7. The model consists of: (i) an input layer, (ii) excitatory and inhibitory layers, (iii) an accumulator to support the inference decision. Excitatory neurons trigger spikes that positively contribute to other neuron's potential to trigger new spikes, while spikes from inhibitory layers emit spikes to decrease this potential.

In a realistic deployment, the (neural) AI agents are expected to be distributed over devices that communicate over shared wireless channels. For example, in use-cases like the ones of collaborating mobile robots aided by distributed sensor networks on a factory floor, or vehicles and infrastructure cooperating in traffic junctions of a smart city, the devices can be highly

localised, reusing the same spectral resources, which calls for efficient allocation of these resources. The aim of this analysis is to understand the impact of losses on different spike traffic types (like input spikes or internal excitatory and inhibitory spikes), and to consequently understand which types of differentiated traffic handling mechanisms can be adopted in such upcoming mobile systems. The aim is to significantly improve the application-level performance metrics, such as the inferencing accuracy.

We consider a network model of a fully shared channel with limited capacity for spike communication capable of supporting different priorities for input, excitatory and inhibitory spikes. In this model the channel bandwidth is shared and there is a delay constraint on spike transmission, which results in increased loss probability for down-prioritised spikes. We focus on the interoperability between a priority-based network scheduling and the inhibitory feedback mechanisms of a SNN, both of which regarded as highly dynamic systems. This interoperability is simulated in three contexts: (i) a baseline scenario where all traffic types are treated equally without prioritisation, (ii) a case where input traffic is prioritised over excitatory and inhibitory spikes (which hereinafter we refer to as internal traffic) and (iii) a case where internal traffic is prioritised over the input spikes (Figure 3-8).

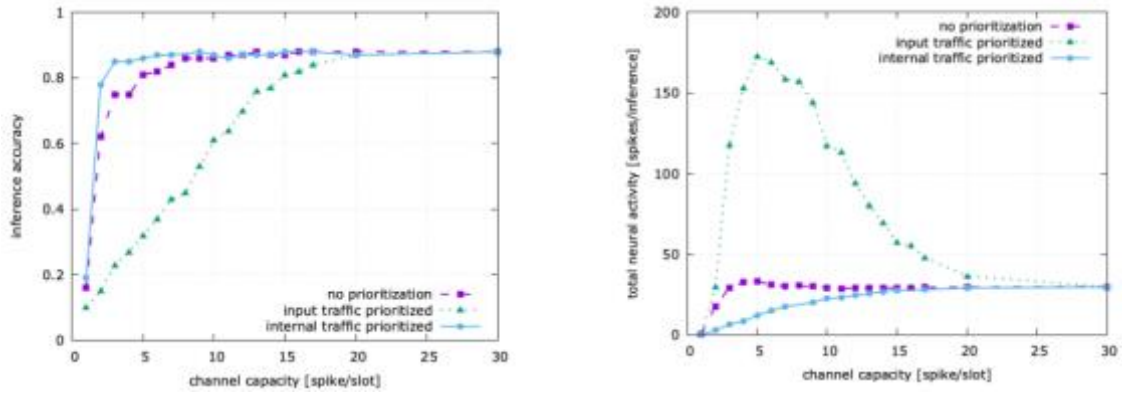


Figure 3-8: Inference accuracy (left) and total neuron activity (right) in case of different spike prioritisation scenarios.

We observe that prioritisation of input traffic degrades the performance in case of limited SNN channel capacity, where it affects the excitatory spikes required directly for classification. The best results on accuracy are achieved by prioritising the internal traffic, i.e., both excitatory and inhibitory spikes over the input, which can be attributed to two factors. First, internal spikes are likely to be more influential for classification accuracy than inputs, since excitatory spikes represent already processed information and features extracted in multiple iterations, which was condensed from hundreds of input spikes. On the other hand, the intensity of internal spikes is reduced by the input through negative feedback, thus creating a self-regulatory system, where internal traffic will never suppress the input completely.

From this analysis, we found that DW-SNN accuracy remains tolerant to high spike losses, due to the redundant structure of SNN. It is also shown that there is a benefit for wireless networks supporting SNN in their ability to differentiate the treatment of SNN internal traffic with respect to input spikes. Having higher priority treatment for internal traffic reduces the overall neural activity regardless of the channel capacity allocated to SNN traffic. This policy also gives the highest inferencing accuracy when few channel resources are available for SNN traffic. This implies that, to optimally handle SNN traffic, a wireless network should be aware of the different classes of the transferred spikes, to be able to consequently apply different traffic prioritisation (e.g., QoS) and reliability methods (e.g., via different radio stack configurations, for example at

MAC or PHY layers) to these different spike types. In this architecture the inferencing accuracy can also be increased incrementally by allowing more steps to compute for the neurual network, which gives the flexibility of a trade-off between the accuracy and latency KPIs.

3.2.3 Multi-agent ML for multi-cell multi-user MIMO

In this section, the beamforming problem in multi-cell multi-antenna systems is considered. Achieving optimal beamforming performance requires complex inter-cell interference coordination in the sense that each BS should exhibit cooperative behaviour to maximise the signal power to a desired user, while minimising the interference power received by other users in the multi-cell environment. This problem poses two main challenges: i) multiple actors (or agents) with partial channel condition observability and ii) multi-dimensional continuous action space. The first challenge is a direct result of practical limitations of accessible information by local agents distributed in a MIMO interference-prone network, and the second challenge comes from the fact that multi-dimensional precoding vectors should be optimised for multi-antenna BSs based on a certain transmit power constraint. In general, the optimisation problems are nonconvex and difficult to solve using traditional approaches.

The multi-cell, multi-user precoding problem can be seen as a multi-agent system that learns to coordinate transmission schemes (or action policies) in interaction with other BSs (or other agents). We propose a Multiagent Deep Deterministic Policy Gradient (MA-DDPG)-based approach that can learn an optimal precoding strategy in multi-cell multi-user MIMO systems under the assumptions of local Channel State Information (CSI) and no inter-cell data sharing. As a case study with tractable performance analysis, we consider a Multiple Input Single Output (MISO) interference channel (IFC) in which two BSs, each equipped with multiple antennas, serve two single antenna users, making the precoding problem tractable by the numerical methods. For instance, in this two-user MISO IFC setup, we can obtain the achievable rate region by using the numerical method proposed in [JLD] and derive the Pareto-boundary of the rate region.

Figure 3-9 illustrates MA-DDPG model for the MISO IFC setup. As shown in the figure, BS $i \in \{1,2\}$ desires to send the data symbol d_i to UE i. The BSs employ n_t transmit antennas and each UE is equipped with a single receive antenna. BS i employs a linear precoding vector w_i of size n_t-by-1 prior to transmission over the air, which transforms the data symbol d_i to the size n_t-by-1 transmitted vector $x_i = w_i d_i$. The channel from the BS i to the desired UE and the other UE are represented by an 1-by-n_t channel vector h_i and g_i, respectively.

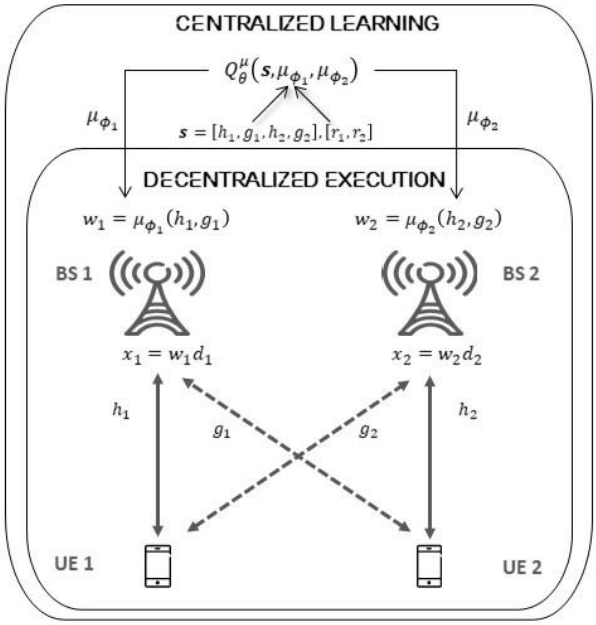


Figure 3-9: Schematic of Multiagent Deep Deterministic Policy Gradient in Multiple Input Single Output Inferference Channel.

The MA-DDPG algorithm allows centralised training with decentralised execution in which local actors μ_{ϕ_i} with partial channel observability $[h_i, g_i]$ can learn a globally optimal policy with the aid of centralised critic Q_θ^μ with global information $s = [h_1, g_1, h_2, g_2]$ at training time and execute the learned policy based only on partial observations at execution time. At the same time, the deterministic policy gradient (DPG) algorithm enables the agents to learn a multi-dimensional continuous policy. The MA-DDPG framework is adopted to improve the quality of the received signals in MISO IFC by alleviating the intercell interference.

3.3 Communication & compute resource allocation

This section presents a vision on how to design and orchestrate 6G systems with a connect-compute perspective that, besides the mentioned enhancements of communication performance, considers computing capabilities as a native part of future networks. The goal is to effectively manage and orchestrate radio (e.g., power, bandwidth) and computing resources (e.g., workload assignment and delegation, CPU scheduling, service placement) to enable 6G as an efficient AI platform able to pervasively and seamlessly offer intelligence capabilities at the edge, with the goal of processing the myriad of capillary data continuously collected by UEs, sensors, vehicles, etc. As technical enablers of this vision, presented in this section, MEC and Compute-as-a-Service (CaaS) play a key role, together with a joint orchestration of radio and computing resources to enable energy-efficient and fast reliable inference, as well as the optimal placement of AI functionalities across intelligent network nodes. The proposed solutions target KPIs and Key Value Indicators (KVIs) that entail: i) energy efficiency at both devices and network side; ii) latency, involving both communication and computing phases; iii) Inference accuracy and/or reliability, properly defined depending on the specific applications; iv) Trustworthiness.

3.3.1 Flexible computing workload assignment (CaaS)

In this section, the proposal is to have computing resource sharing as a native capability of the 6G network. APIs specialised in dispatching task delegation requests and fetching connection details of the in-network computing nodes available and capable of undertaking the workload are proposed to be open and available to any network component both at network infrastructure and user side. The referred computing nodes may be implemented at any suitable level, ranging from a distant data centre to a nearby edge cloud node and anywhere in between.

Nevertheless, the incurred challenge is that the available computing platforms may be of varying complexity – some platforms may provide Common-Off-The-Shelf (COTS) standard CPUs, while others may consist of a highly heterogeneous setup, typically including CPUs, Graphics Processing Units (GPUs), Field Programmable Gate Arrays (FPGAs), custom AI accelerators and the like. Ideally, to facilitate ease of use for a workload delegation service, the exact configuration of a targeted platform would need to be abstracted from the API in order for the user to only need to provide service requests designed for a standardised platform.

The CaaS related API is proposed to provide the following services:

1. *Pre-installed services*: A Service Provider (SP) offers pre-installed services. The user is able to access pre-installed applications which are, thus, under full control of the SP. The SP can choose those applications which are suitable to be executed on its platform and will provide an optimum configuration exploiting the available resources to the maximum extent possible.
2. *Virtual Machine (VM) based resources*: in this case, a SP offers VM services, relying on so-called ConfigCodes. i.e., the API provides access to an abstracted computational platform, which is independent of the underlying hardware and available physical resources. The user is, thus, developing code for such a unified architecture, which is then mapped onto the target platform by the SP.

   One of the candidate VM approaches builds on the solutions provided by [ETSI17]. An (elementary) Radio VM ((e)RVM) is introduced, comprising in particular Data Objects (DOs) and Abstract Processing Elements (APEs) which are connected through an Abstract Switch Fabric. A two steps compilation process is applied: first, the front-end compilation step provides so-called ConfigCodes which are related to the abstracted architecture in Figure 3-10. Any target platform will then perform the back-end compilation step to transcode the ConfigCodes to the specific available set of hardware resources. This approach combines code portability with execution efficiency.

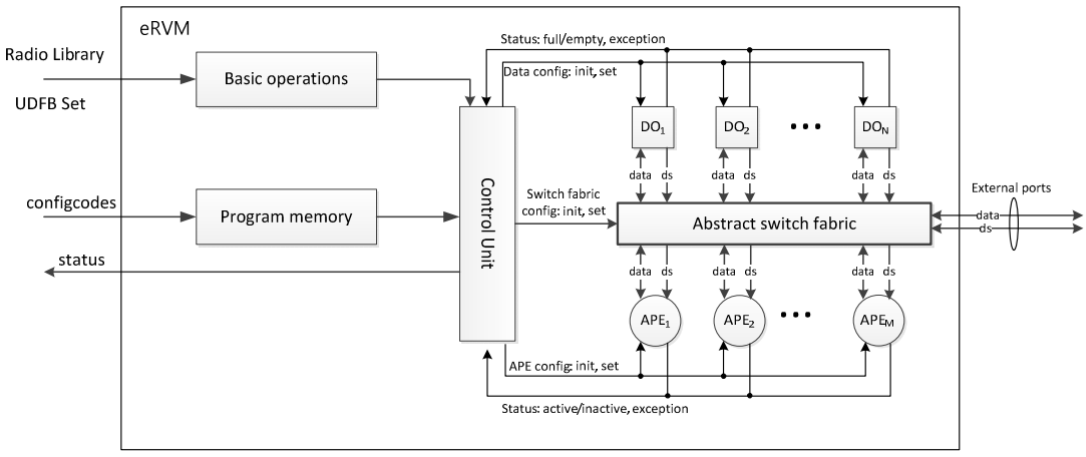


Figure 3-10: Abstracted architecture of a Radio Virtual Machine (RVM) [ETS17].

3. *Open Execution Environment*: in that case, a SP offers access to a specific set of resources, typically comprising computational resources, memory, etc. A user would be required to develop code which is tailored to the specific needs of the concerned hardware. The usage of a different platform will likely require an adaptation of the code.

In the specific case of CaaS being offered by edge cloud components, some of the key requirements are as follows:

- Workload may be distributed across multiple local and/or distributed components to optimally exploit locally available compute resources;
- a user may be moving and, thus, attaching to different (distant) edge cloud nodes in the future; consequently, a transfer of the compute task and/or (intermediate) processing results is implemented.

The basic principles of the "follow me computer" CaaS paradigm are outlined in Figure 3-11 below. Further details of the proposed approach and results aim to appear in D4.3. KPIs of relevance is the ones of AI agent availability and network energy efficiency, while KVI of relevance are the one of flexibility.

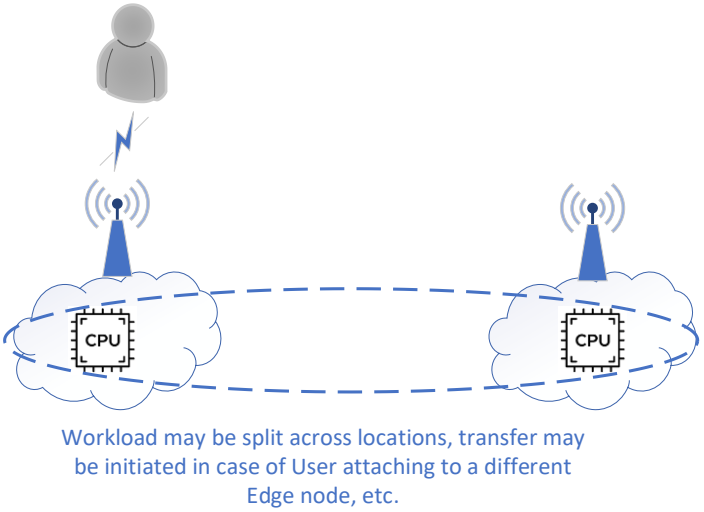


Figure 3-11: The "follow me computer" CaaS approach in a multi-operator 6G network.

3.3.2 AI workload placement for energy, knowledge sharing and trust optimisation

Nowadays, new technologies become widely available, bringing significant improvements in the way applications and services are delivered to society. Nevertheless, besides the improvements, a number of potential novel risks and harms are created, which are associated to the operation of the respective technologies. Such an example is the AI enabler, which is gradually part of all modern information and communication technology systems. AI delivers great benefits for economies and society and supports a more fair, safe, and inclusive decision-making. At the same time, this decision-making must be operated by nodes in a trustworthy and sustainable manner. Hence, there is a great need for managing the AI operations, especially in decentralised scenarios; this can be succeeded with the design of a novel AI management system, applicable in B5G / 6G architectures.

Physical nodes, e.g., user devices or edge/cloud servers, that undertake the execution of AI workloads, comprising diverse AI models/algorithms/mechanisms can face trust level problems, traffic problems, due to the extended usage of highly-challenging services in terms of datarates,

ultra-low latency machine-to-machine communications, AI-related knowledge sharing or energy consumption problems. Some key areas of this problem are the critical applications with safety and trust considerations, the Human - AI collaboration with human-in-loop when, for example, decision-making should stop the automated mode for a reason and a human should take charge of it. Another key area is the case where there may be a need of fairness appraisal when data traffic comes from different countries or continents assuming there are no trust zones issues, and finally AI services with energy footprint constraints and more.

The main challenge is to optimise the placement of the AI algorithms to the various physical nodes with respect to the energy consumption of the overall network towards sustainability, the traffic, and the trust of these physical nodes. The research targets for this task is to investigate algorithms for solving optimisation problems for allocating AI models/algorithms to physical nodes optimally following (meta-)heuristic techniques for low computing requirements and as a result, efficient (near optimal) AI placement solutions. One of the crucial aspects that need to be taken into consideration is the data availability, along with respective constraints on their movement along the various network segments. Towards utmost privacy, but also scalability, the placement of AI workload must consider the required data vicinity, e.g., in terms of network hops/available bandwidth for transferring the data, etc. Besides the above, special features/characteristics of AI mechanisms/algorithms, different input data models, behaviour models, knowledge sharing needs among nodes/segments of the network, as well as considerations related to trust level and more, will be taken into consideration.

The most relevant KPIs for assessing the AI workload placement/migration are network energy efficiency (percentage of energy consumption reduction with the proposed AI workload placement algorithm compared with a baseline/random placement algorithm), AI agent reliability (ensure that a selected, trustworthy node, allocated an AI operation, timely respond/notify in case of failure, missing data, etc.), latency (AI workload placement/migration algorithm execution time), AI agent availability (ensure high availability of AI agents) and a relevant KVI is trustworthiness.

To this end, a new functional entity for managing AI mechanisms is identified. A new network service will be developed on AI governance/management, foreseen for B5G/6G systems This work is part of an innovative approach related to managing, monitoring, and trusting of AI mechanisms in B5G/6G.

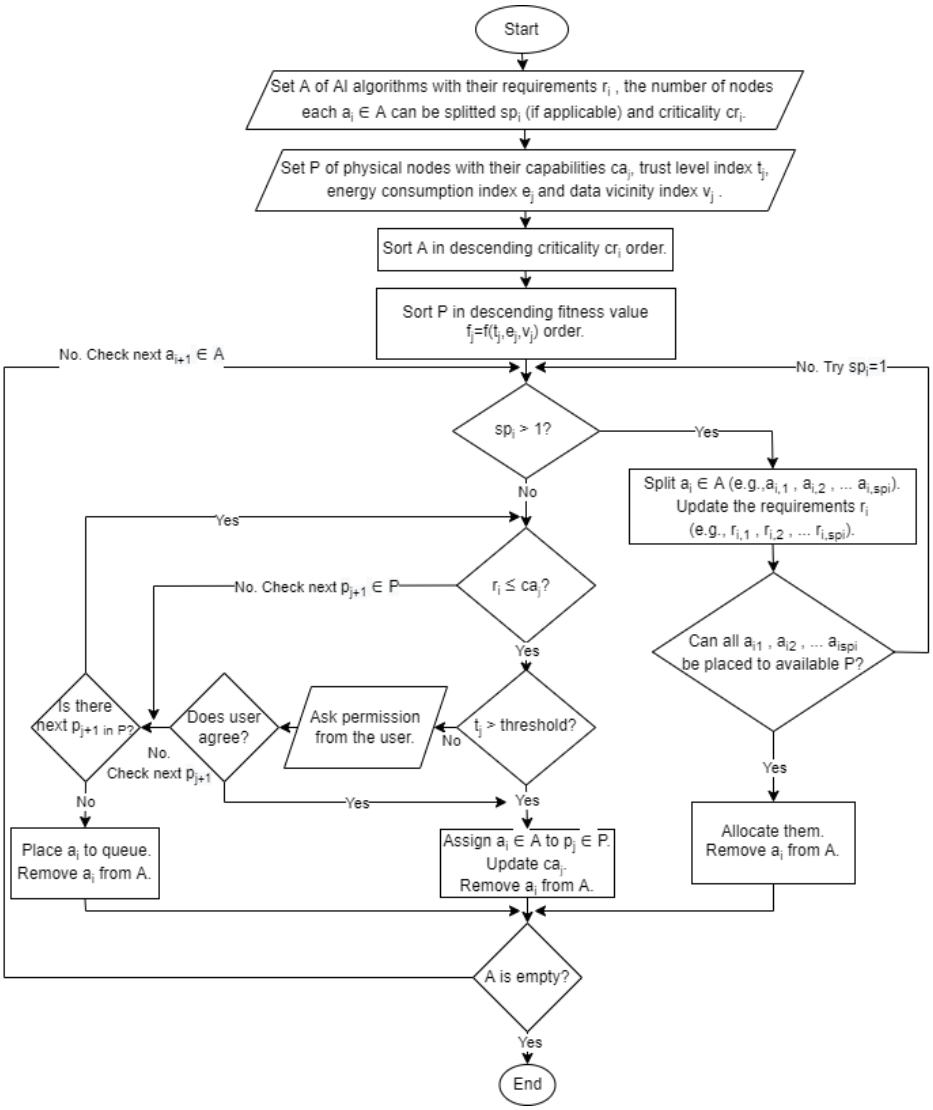


Figure 3-12: Overview of AI workload placement algorithm.

Figure 3-12 shows our approach of solving this problem (overview of the algorithm). We are developing a heuristic algorithm/solution method, which takes as input the set of the AI algorithms/mechanisms with their computational requirements (CPUs/GPUs, Random Access Memory-RAM etc.), criticality level, and their ability to split in more than one node. Additionally, it takes as input the set of physical nodes (cloud, MEC hosts, extreme edge nodes of the network) with their computational capabilities (available CPUs/GPUs, RAM etc.), communication capabilities (bandwidth and network links' information), their trust level index which shows how safe, fair etc. it is to use them, their energy consumption index which indicates how energy efficient it is to use each physical node, and their data vicinity index which provides information related to the location of the data needed for each AI algorithm. We model the system as a graph of compute nodes/physical nodes.

The set of the AI algorithms is sorted based on criticality level with the intention to allocate first the most critical algorithms/mechanisms. The set of the Physical nodes is sorted accordingly based on a fitness value which is a function of trust level, vicinity and energy consumption index. The algorithm distributes the AI workload to multiple nodes, taking into account the splitting capability of the AI workloads, and perform the allocation-distribution based on the respective trust level and computational requirements and capacity of the nodes. Moreover, it checks if there

is a node with lower trust level than a threshold and asks the user to choose if this node should be utilized or not. Finally, it checks with descending criticality order the physical nodes (with descending fitness value order) having the needed or more than needed computations capabilities to execute each AI algorithm. The output of the algorithm is the placement of the AI algorithms to the physical nodes. The results/evaluations will be provided to D4.3.

3.3.3 Joint allocation of communication and computation resources for inference at the edge with low energy devices

Learning and inferencing at the edge is a complex and challenging task from several perspectives, due to the fact that data must be: (i) collected by end devices such as sensors, UE. etc. (i.e., involving data sampling criteria), (ii) possibly pre-processed/encoded (e.g., data compression, quantisation, etc.), and (iii) finally processed (i.e., occupying computing resources in edge servers) to output the result of training or inference phases. One of the challenges lies on the fact that all network resources (communication and computation)) are involved and should be managed, possibly *jointly*, in order to strike the best trade-off between energy consumption, E2E delay (both comprising communication and computing) and learning/inferencing accuracy/reliability/trustworthiness. This section focuses on the inferencing phase of the edge intelligence paradigm [ZCL+19], with the goal of energy efficiency. Relevant KPIs/KVIs are energy efficiency, end-to-end latency, and inference accuracy.

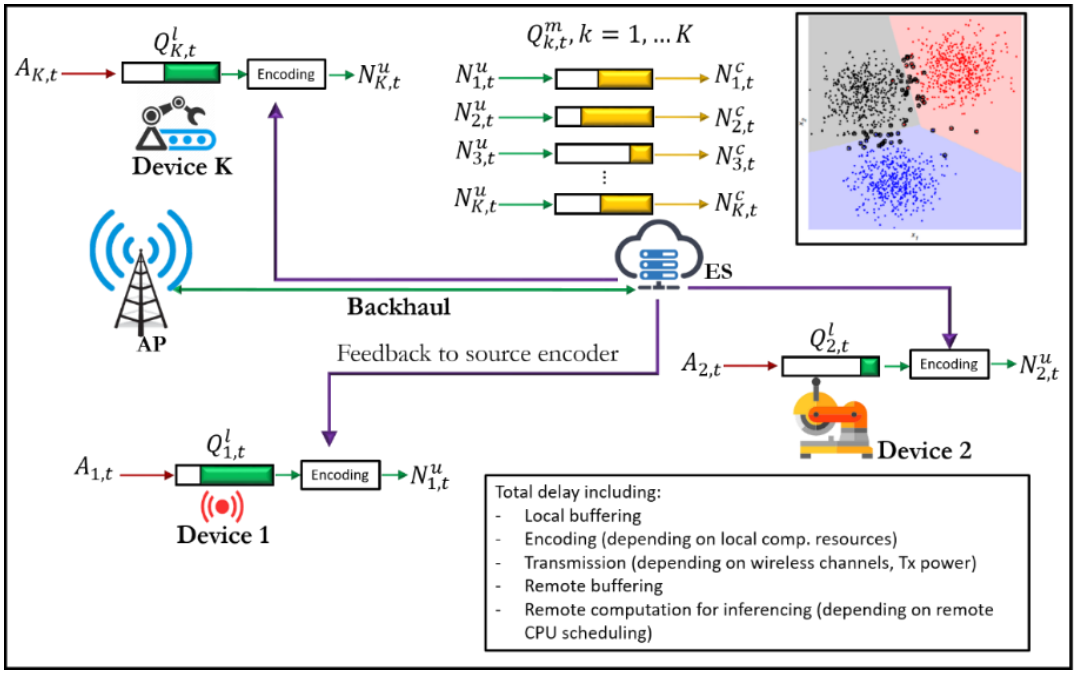


Figure 3-13: Scenario for dynamic edge inference.

As an example, Figure 3-13 shows the reference scenario, in which end devices upload data to an edge server that runs the inference task. In this case, as briefly described in the figure, data experience several delays, including communication and computation buffering, as well as encoding, transmission and remote computation for the final inference. Also, a closed loop between the edge server and the source encoder of the devices is used to opportunistically shape data representation, based on confidence measures that the edge server eventually measures online and feeds back to the devices (e.g., via a downlink control channel), thus, dealing with highly time varying context parameters, such as wireless channels. As pointed out in [ZCL+19], there are several aspects to be tackled when performing edge inference tasks, and also different

solutions to improve the efficiency of the overall procedure. Specifically, it is possible to use model compression techniques to reduce the memory footprint [HPT+15], model selection strategies to adaptively select the inferencing model online [TMW+18], or application-oriented optimisation, for instance, adapting frame rate and resolution of video streaming for resource efficient classification [JAB+18], [RCZ+18]. Other works explore the trade-off between energy, delay, and accuracy [GAL21],[LCL+17], [LHO+18]. The work in [LSL+22] provides a recent survey and vision on the edge AI paradigm in 6G. The problem focused in this section aims to explore the mentioned edge AI trade-offs and involves the following control variables, to be jointly optimised online: i) Data encoding strategy (e.g., compression, quantisation, etc.); ii) local computing resources for data pre-processing; iii) uplink transmit power of end devices; iv) edge server CPU scheduling, being the latter shared among all users.

As already mentioned, the aim of this study is to minimise the energy consumption of end devices, under system stability (i.e., finite E2E delay) and long-term constraints on the inference reliability. In our system, stability is achieved when both local communication and remote computation buffers are stable (i.e., their averages do not grow to infinity). The energy consumption of devices accounts for local computing energy to encode data and transmit energy to upload data to the edge server through the wireless connection with the AP. To this end, we formulate a long-term problem to deal with system dynamics, such as radio channels and data arrivals, whose statistics are supposed to be unknown in advance. Thanks to Lyapunov stochastic optimisation tools, we are able to translate the long-term problem into a per-slot program very easy to solve, which only hinges on instantaneous knowledge of context parameters, involving radio channels, data arrivals, and queue states. More technical details can be found in [MBD+22].

Numerical results

We show numerical results on a CNN based edge classification task over the CIFAR-10 data set [Kri09]. Numerical results are shown in Figure 3-14, which represents the energy-delay-reliability trade-off. In particular, in Figure 3-14(a) we show the trade-off between E2E delay (y-axis) and energy consumption (x-axis) of six different devices with different requirement in terms of inference reliability. We can notice how, as the consumed energy decreases, the E2E delay

increases.

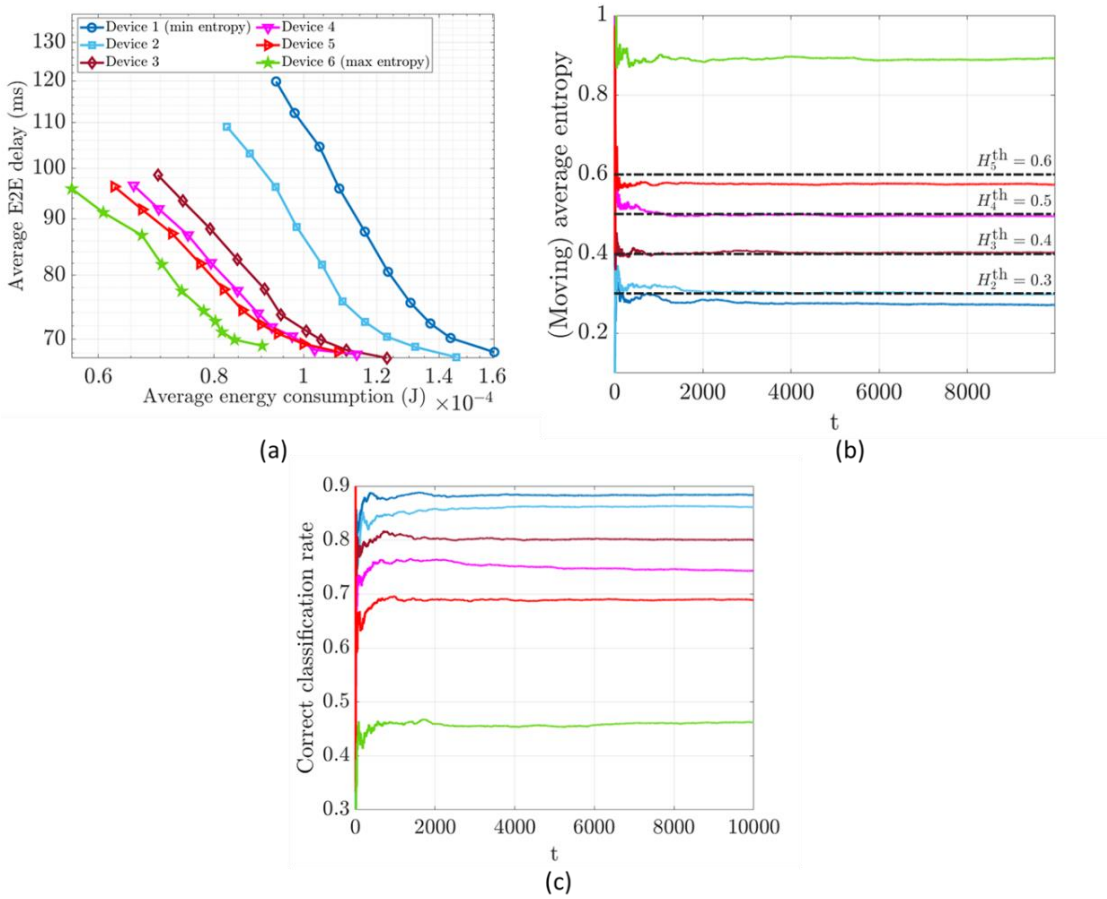


Figure 3-14: Energy delay-reliability trade-off in edge AI.

Let us now focus on our two benchmarks, represented by the blue curve (best inference performance) and the green curve (worst inference performance). Here, the reliability is measured by the entropy at the output of the CNN, thanks to a posteriori probabilities. In particular, the lower is the entropy, the higher is the inference reliability. We can appreciate that the highest reliability (H = 0.27, which translates into around 88% accuracy, as visible in Figure 3-14(b) and (c) that show the time average values of entropy and accuracy as a function of iteration index t, respectively) comes at the cost of a higher energy consumption (for a given E2E delay), due to the fact that more bits per example must be transmitted. On the other hand, achieving the "green solution" (i.e., minimum energy consumption) comes at the cost of a very low reliability (H = 0.88, which translates into around 45% accuracy). However, among these two "extreme" cases, one can select different values of target entropy that the method is able to guarantee (see Figure 3-14(b) with the corresponding black dotted horizontal lines), while achieving lower energy consumption as compared to the highest accuracy case, but a much higher reliability and accuracy with respect to the minimum energy case. For instance, looking at the orange curve, we can conclude that a very small degradation in the accuracy (around 2%) is able to guarantee an appreciable gain in terms of energy-delay. Similar conclusions can be drawn for the other curves. Future work will include the whole network energy consumption, for a holistic view of edge AI systems.

4 AI/ML as an enabler for 6G network sustainability

AI/ML methods constitute a great opportunity that can and should be leveraged to improve network sustainability. In this objective, at least three different strategies can be envisioned:

The first one is to use AI methods to improve the network's sustainability by enhancing the energy efficiency of the network. The disadvantage of using learning-based method (AI) as opposed to the classical signal processing methods is the additional burden of training the AI model. The high complexity of training phase could be offset by the lower than state-of-the-art complexity of the inference phase. Solutions for this strategy are proposed in Section 4.1.

The second category is to apply frugal design to AI methods for which the training phase is designed to be sustainable. This approach is desirable for online methods, where a certain function in the network requires constant update of the AI model, e.g., online fine tuning. Section 4.2 is devoted to this approach.

The third direction views communications beyond reconstructing the exact symbols of the source at the destination. The focus here is on semantic communications, which leads to saving unnecessary energy consumed to recover the exact messages. In Section 4.3, we investigate this strategy in more details.

4.1 Sustainability by complexity reduction

4.1.1 Low complexity radio resource allocation in cell-free massive MIMO

Radio Resource Management (RRM) is an important task in any communication network, which helps improve the system performance by efficient utilisation of available resources, such as power and bandwidth. Conventionally, radio resource allocation problems are solved using optimisation or heuristics-based methods, considering CSI and QoS requirements of the users. These methods have several challenges, such as high computational complexity and requiring precise CSI, resulting in sub-optimal solutions in complex and non-convex problems, lack of flexibility and parameter sensitivity, and inaccuracy of the model-based resource allocation methods (due to channel modelling issues and hardware impairments). Novel communication architectures such as cell-free massive MIMO networks and high-frequency communication systems have an increased system complexity due to the large number of antenna elements in the transceivers and the increased AP deployment density for a high-frequency Radio Access Technology (RAT). RRM becomes more challenging in such systems due to the increased system complexity and high dimensionality of the resource allocation problems. The aim of this section is to investigate the potential of AI/ML approaches to reduce the processing complexities and timing overheads in performing resource allocation tasks in cell-free massive MIMO networks. In the literature, several studies have proposed DL-based power control for cellular and cell-free massive MIMO systems [AZB+19, CCB+20, LSY+20, SZD18, ZNG20]. Most of the existing studies focus on a supervised learning approach where a model is trained to learn the mapping between the inputs (user locations or channel statistics) and the optimal outputs (power allocations) obtained by an optimisation algorithm. The unsupervised learning algorithms proposed in [LSY+20, RMR+21] for K-user interference channel power control problem and for the uplink power control in cell-free massive MIMO eliminate the need of knowing the optimal power allocations during model training.

An unsupervised learning-based resource allocation approach is proposed which does not require the optimal resource allocations to be known during model training as in supervised learning, hence it alleviates the need of generating a large dataset with thousands of samples by solving the computationally complex optimisation problem. Thus, the proposed approach makes the data preparation and model training simpler, more practical and flexible, since the deep learning model can be easily retrained in a changing environment over the time. Complexity gain and flexibility are the target KPIs where the reduced computational complexity of the ML solution and its flexibility in adapting to changing environments and system configurations are evaluated in comparison to existing optimisation-based approaches.

One problem scenario considered is an uplink of a limited-fronthaul cell-free massive MIMO network with hardware impairments, where the objective is to maximise the sum throughput of the network. A Deep NN (DNN) which takes in large-scale channel coefficients as input is trained in an unsupervised manner to learn the optimal user power allocations and fronthaul capacity allocation between CSI and data in order to maximise the network sum throughput. Input to the model are large-scale channel coefficients of the users in the network, and the outputs are the user power allocation vector and the capacity allocation vector of the APs. The loss function for model training is defined as the negative value of the sum rate which is a function of large-scale channel coefficients and the trainable parameters θ of the DNN. This loss function is differentiable with respect to the trainable parameter set θ which allows training the model via Stochastic Gradient Descent (SGD) method. Mini-batch gradient descent approach is used to reduce the complexity of the SGD. In each iteration of the training, a set of channel realisations are generated from its distribution, and the average loss is calculated over the mini-batch. During training, the model learns parameters θ to minimise the loss which maximises the sum rate and outputs the power allocations and fronthaul capacity allocations. The DNN could be used in two modes; offline training mode, where the model is trained offline for a large dataset with different channel instances, or online training mode, where the offline trained model is retrained in each channel instance allowing further customisation and fine-tuning of model parameters based on large-scale channel inputs in each channel realisation, to further optimise the sum rate performance.

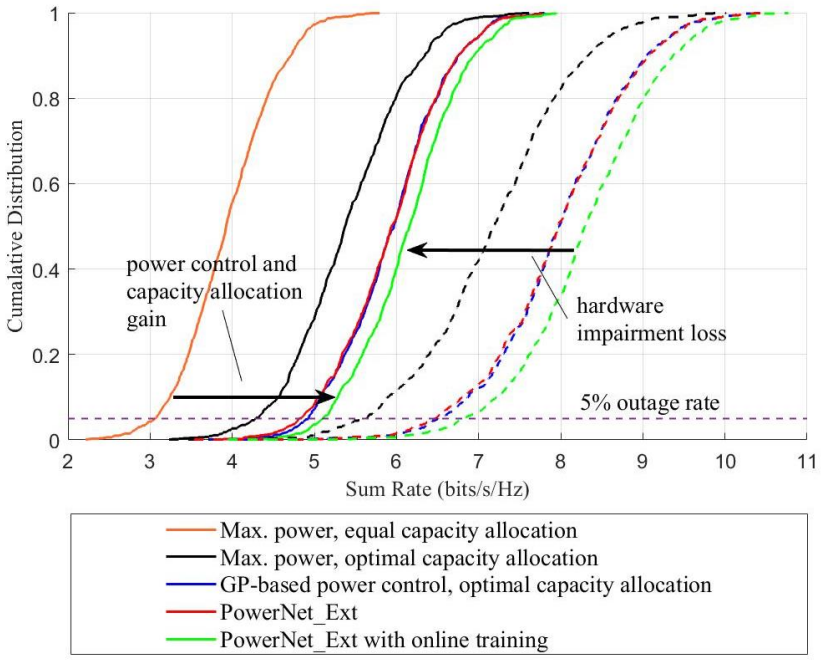


Figure 4-1: Sum rate performance comparison between proposed method (PowerNet_Ext) and baseline (optimisation-based) with and without hardware impairments in the transceivers. Dashed lines: Ideal transmitters and receivers without hardware impairments. Solid lines: With hardware impairments in the transmitters and receivers.

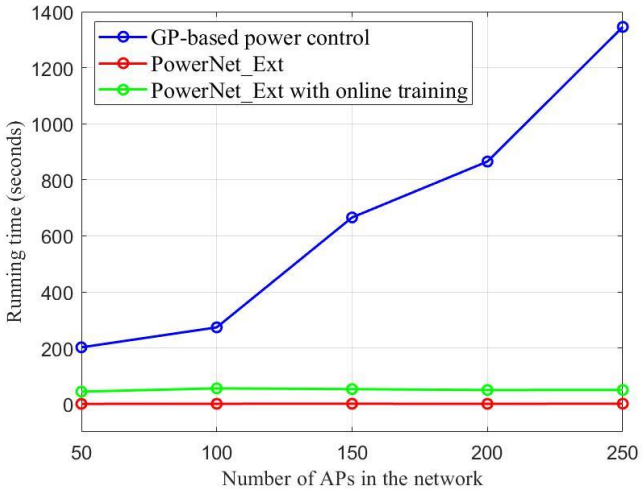


Figure 4-2: Recorded CPU timing for the CVX solver and the PowerNet_Ext to produce outputs for 100 channel realisations for different network configurations.

As observed from Figure 4-1 and Figure 4-2, the deep learning-based method achieves similar or better performance to the optimisation-based resource allocation (geometric programming approach) with a significantly lower time-complexity. The time-complexity of the DNN does not drastically scale with the number of users and APs as the optimisation-based algorithm. While the time complexity of the optimisation-based algorithm exponentially scales when the number of APs or users are increasing, the time complexity of the DNN only increases slightly when the system parameters are scaled. The model PowerNet_Ext, which learns the outputs from only using the offline training, takes less than 0.01 s for calculating the outputs. One-shot output calculation which only involves matrix multiplication and addition operations to produce the DNN outputs in contrast to the iterative steps in the optimisation algorithm is one reason to the significantly reduced time complexity of the DNN. It should also be noted that the optimisation algorithm was implemented using Matlab CVX, whereas the DNN implementation is done in Tensorflow, whose implementation differences may have added to the timing differences in the two approaches. Lower computational complexity and acceptable performance of the proposed DL-based approach makes it a potential candidate for practical implementation.

4.1.2 Supervised learning based sparse channel estimation for RIS aided communications

Reconfigurable Intelligent Surfaces (RISs) enable a reconfigurable wireless propagation environment, where the propagation path of signals can be modified with software-controlled scattering [QR20]. Generally, this is achieved by inducing a phase shift of the waves impinging on a passive element at RIS, where the phase shift can be controlled electronically. RIS consists of many such elements, and by intelligently controlling them the spectral efficiency of the communication can be improved. However, the proper control of RIS needs channel information to be available at the entity performing optimisation (in most cases, the RIS is assumed to be

controlled by the BS, where the BS computes the optimum phase shifts based on available channel information and send control signals to configure the RIS). Yet, in practice, it is challenging to obtain channel information, as an RIS consists of large number of passive elements, which do not have any sensing capability in general. It is necessary to estimate both the direct channel and reflected channel through RIS.

Channel estimation for RISs requires the design of activation patterns for the RIS elements. The impact of RIS activation pattern on the channel estimation performance is investigated in [TE20], where an optimal codebook is proposed based on a minimum variance unbiased estimator. The angular domain sparsity in mmWave channels can be exploited to reduce the dimensionality of the parameter space, which results in more accurate results with reduced pilot overhead. A sparse representation of the concatenated BS-IRS-user (cascaded) channel is derived in [PJH+20], which convert channel estimation into a sparse signal recovery problem. The double-structured sparsity of the angular cascaded channels is leveraged in [XDL21] to propose a Double-Structured Orthogonal Matching Pursuit (DS-OMP) based cascaded channel estimation scheme. A channel estimation scheme that gives an accurate computation of the channel will be beneficial to the effective utilisation of RISs, which in turn can enhance the energy efficiency of the network. Therefore, in this work we propose a channel estimation method based on the sparse representation of the channel, while a NN is used to estimate angular parameters. The solution is a one pass method compared to iterative traditional sparse estimation techniques, while results show better accuracy compared to [TE20].

We consider the uplink of a mmWave network, where an RIS is used to assist the communication [DMR+22] . We assume that the user lacks LoS with the BS, and the RIS has LoS to both BS and user. Based on this model, we develop a compact representation for the RIS channel. An angular domain sparse channel model is considered by discretising the angle of arrivals (AoAs). The channel estimation problem is formulated step by step for the case where Angles-of-Arrival (AoAs) lie exactly on the discrete grid (on-grid), and the case where AoAs can take any discrete value deviating from discrete grid (off-grid). A sparse estimation method is proposed for the on-grid case, based on Orthogonal Matching Pursuit (OMP), and a NN based approach is used for comparison. In the off-grid case, a twostep procedure is used to perform channel estimation, where first, on-grid AoAs are estimated using a NN and then off-grid AoAs are calculated by predicting the residuals using another NN. The NN that is used to predict the AoAs is shown in Figure 4-3. The input to this network is the received pilot signals, where the real part and imaginary parts are separated and concatenated.

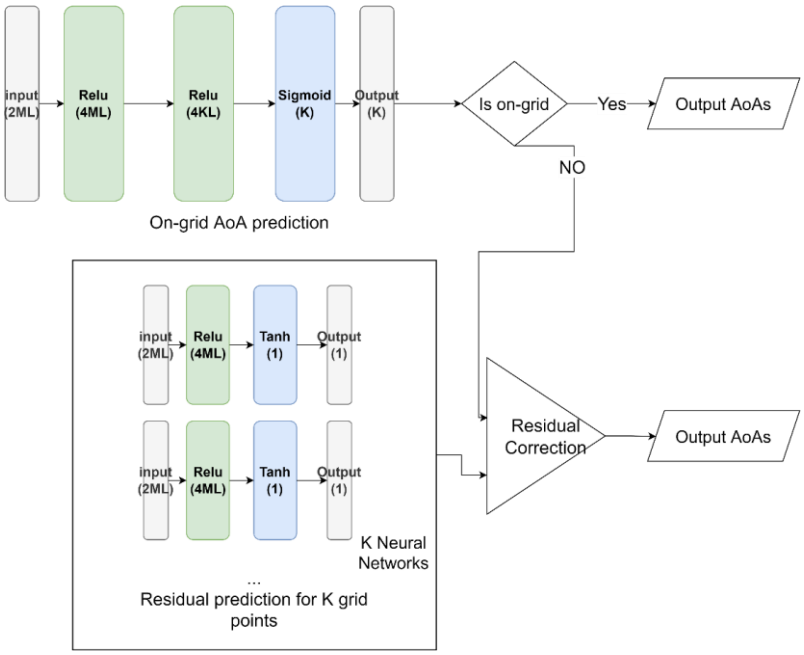


Figure 4-3: NN architecture for AoA calculation.

When evalulating the results, the main KPI considered is the channel estimation error. Since the proposed method relies on sparse estimation, it requires a smaller number of parameters to be estimated compared to traditional methods like least squares estimation. This improves the performance of channel estimation, as shown in Figure 4-4. The performance of NN based channel estimation is compared with Least Squares (LS), and the deterministic algorithm under both perfect and imperfect AoA values, where in the perfect case, residual errors of AoAs are assumed to be perfectly known. In the NN method, first on-grid AoAs are predicted, and then residual values are predicted based on the active AoAs, which corresponds to the error between the perfect on-grid angles and the actual observation. The proposed NN has been able to predict the AoAs giving a close performance to the situation where grid points are perfect. The NN based solution outperforms both LS and deterministic algorithm. However, in the direct channel, a saturation of performance is seen at high transmit power, which is due to power leakage under grid imperfections.

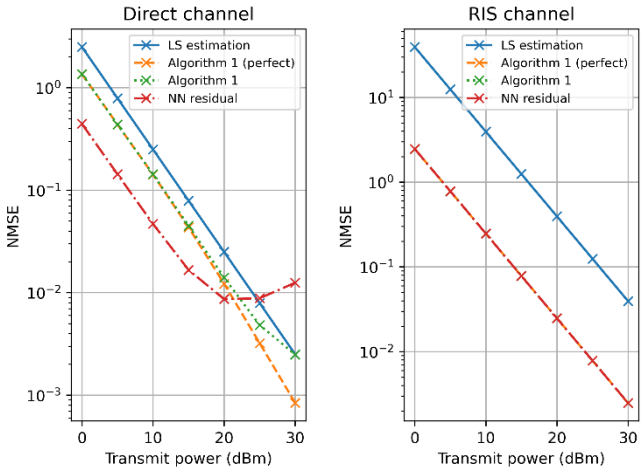


Figure 4-4: Comparison of performance of proposed methods with LS estimation for both direct and reflected channels, in the off-grid case.

4.2 Frugal ML

This section presents different possible ways of relaxing the required size of learning data and size and number of ML models that are needed for high-performance inferencing in the first case and for high-performance signal processing (for channel estimation) and channel charting (useful for downlink transmissions) in the second case. Exploitation of any available (side) information, such as the evaluation of the significance of a data point to the expected ML model update or knowledge of the physical structure and/ or temporal evolution of the wireless channel is shown to reduce demands in sample complexity for channel estimation and enable application of lean channel learning designs (offline or online) not requiring large-sized NNs to guarantee high communication performance -and, therefore, large pools of processing, memory and storage resources across the network. For the channel charting task, a channel distance measure is proposed leading towards a frugal implementation of a channel chart without sacrificing the needed accuracy for effective transmissions based on channel charts.

4.2.1 Over-the-air model learning for data-frugal and resource-efficient network AI operation

Over-the-air ML model learning is impacted by joint *data uncertainty* (entropy) and *channel uncertainty*, due to wireless channel volatility, such as interference, noise, and channel model mismatches [LZZ+19]. The incurred challenge is how to consider both uncertainties in the ML model learning process (assuming e.g., the ML model is instantiated and maintained at the edge of the RAN, e.g., at an edge cloud server in proximity to a wireless AP) and design transmission and radio resource allocation policies to enhance in-network learning capability in an efficient way. Efficiency refers to both radio signalling and data storage/ memory savings, therefore, translated into E2E network energy and cost efficiency.

The proposed approach is to distribute available radio resources beyond time (e.g., power, frequency) to wireless devices participating to the learning process for transmission of learning data according to how much significant the learning data points are for the original training or update of an ML model. A first question is *how* to quantify data importance for ML model training. Different data contextual attributes may be exploited, e.g., the *age-of-data*, equivalently, the time elapsed between the generation of a data point (e.g., a sensor measurement) and the time of decision at the user device (or machine) on whether to transmit this data point in order for the ML model to use this data point for learning updates. Another possible criterion to quantify data significance can be the locations where data sets are generated; this approach may be of specific relevance to e.g., smart factory scenarios involving collaborating robots. A second question is *which* radio resources, beyond time, can be scheduled in way such that transmissions of significant learning data are prioritised over others.

When it comes to evaluating the significance of a data point (or a data set in batch mode) *before* scheduling its transmission and allocating resources to it, a third fundamental question is *where* this evaluation should be carried out. One option is to do that “a priori”, or, before data point transmission at the data source (e.g., user device, robot or machine). In this case, the latest version of the ML model (or a compressed version thereof or its metadata) needs to be already available to the device for local data point evaluation. In such a setting, the question is how to provide minimal, albeit still accurate feedback of ML model updates to devices providing learning data for local data significance evaluation. Foreseen advantages and caveats of this approach are: (i) important and not outdated data points will only be transmitted – communication and UE energy-efficient operation (advantage); (ii) requires frequent centralised model broadcast – there is a price to pay in energy consumption, UE storage availability & occupation of downlink resources; also

calls for regular CSI feedback (disadvantage); (iii) data point importance may be "outdated" when data point is received by the radio AP; this results to degrading AI inferencing quality (disadvantage).

Another option is to evaluate the significance of learning data "a posteriori", or, as much as possible close to the ML model to be updated, *after* transmitting the data point in the uplink. In that case, expected advantages and caveats are the following:

- There is no need to (frequently) broadcast centralised model updates; this approach is downlink radio resource-efficient and, therefore, energy-efficient at radio infrastructure side. It is also storage-efficient at the user device, machine or robot side as there is no need to allocate storage and memory resources to host the updated centralised ML model. There is also no need to send CSI feedback to participating devices (advantage);
- non-important data points will be transmitted inevitably, as they will be only evaluated upon reception; as a result, less resources will be available to transmit and store upcoming significant data (disadvantage);
- network scheduling of device transmissions may lead to non-exploitation of important data arriving at these devices in the meantime – reducing AI generalisation capability (disadvantage).

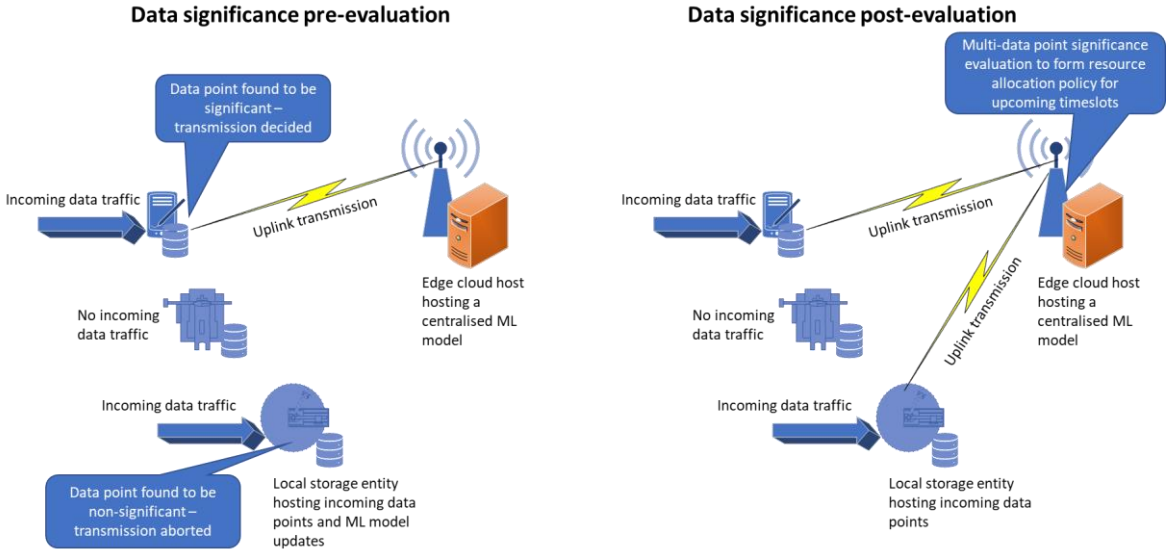


Figure 4-5: Data significance pre- and post-evaluation driving resource allocation in a factory setting.

In this section, the aim is to conduct a performance comparison of the two approaches for different system scenarios and identify regimes where either the on-device (or, a priori) or the centralised (or, a posteriori) data set importance evaluation approach, as illustrated in Figure 4-5 provides performance and network sustainability benefits. In terms of 6G KPIs, the performance comparison aims to chiefly focus on the metrics of inferencing accuracy and E2E latency, along with the ones of generalisation capability, ML model convergence speed, data storage efficiency, spectral efficiency, communication/ signaling overhead reduction and, eventually, E2E energy efficiency. Performance analysis results aim to be presented in D4.3.

4.2.2 Low complexity channel estimation using NNs mimicking MMSE

One approach to perform channel estimation using NNs is by supervised learning. Fitting an NN to perform channel estimation with high performance that fits a particular channel model has been shown by many previous research works [CMW21+a]. This approach usually leads to larger NNs. Trimming or pruning these larger NNs is possible, however, the resulting architecture could be different for each deployed channel model. This leads to many complications when it comes to hardware implementation.

In contrast to this approach, we propose a more explainable and leaner NN architecture for this task. The mainstream way, which is common in the DNN literature for image and natural language processing, is to begin with a large NN and then prune the NN to reduce the size and complexity. Instead, we propose to benefit from the special structure of the problem of channel estimation and begin with a minimal design and repeat this design when we require to learn more complex channel models. This minimal design, coined as Turbo-AI in [CMW+21a], in our solution is inspired by the Minimum Mean Squared Error (MMSE) estimator.

The MMSE estimator inspired NN design for channel estimation has been reported first by [NWU18]. We then extend this design to accommodate multiple antennas with a large number of subcarriers. The essence of our approach [CMW+21a, CMW+21b] is to use the same architecture in order to address the channel estimation task for large antenna arrays with much lower complexity than the original work. The complexity reduction gain that is obtained by Turbo-AI stems from leveraging the Kronecker decomposition of the covariance matrix. We show many versions of Turbo-AI and its versatility in fitting into many conditions and requirements in terms of complexity and accuracy of the estimation.

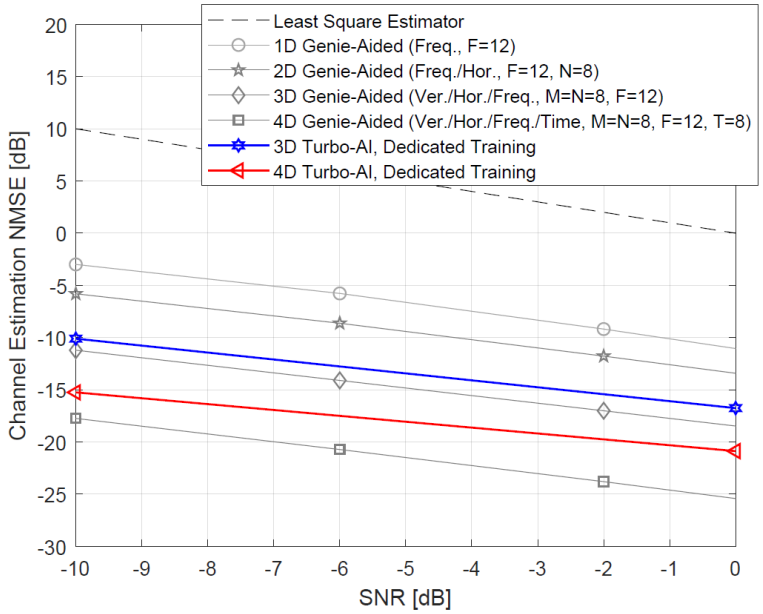


Figure 4-6: Turbo AI method compared with lower complexity method developed based on MMSE formulation in terms of NNs. This type of design paradigm allows for producing a solution for wide range of computational complexity limits.

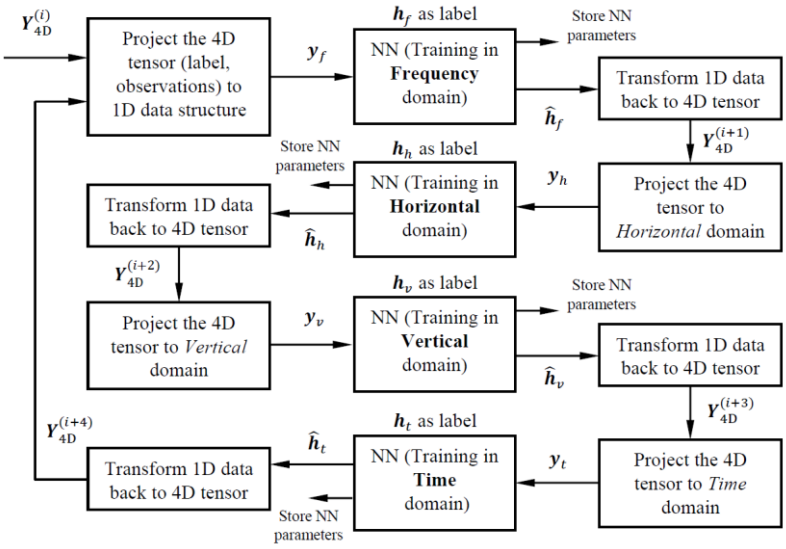


Figure 4-7: The Turbo-AI idea repeats a small NN that is inspired by MMSE to handle the correlation in different directions of the channel tensor, namely: frequency, horizontal spatial, vertical spatial, and time.

In Figure 4-6, the performance of our proposed solution, Turbo-AI, has been compared with MMSE the Neyman-Pearson optimal solution. The modular structure of the Turbo-AI is depicted in Figure 4-7, in which we extend the results to consider frequency, time, and spatial domains of the channel tensor for better estimation performance. Although this extension comes naturally from the design that is proposed by the initial Turbo-AI in [CMW+21a], we need to further investigate its characteristics. The modular structure introduced in Figure 4-7, allows for targeting multiple complexity vs accuracy trade-offs.

One further advantage of Turbo-AI compared to the classical MMSE is its run-time sample complexity or frugality. The classical MMSE requires to estimate the covariance matrix of the signal, which demands a large amount of data samples. The Turbo-AI, however, performs well on a single observation of the channel. The covariance matrix estimation and inversion is learned implicitly within the NNs, avoiding huge computation burden and latency in collecting data. The 6G relevant KPI of channel estimation error from Table 7-1, directly fits into the objective of this solution.

4.2.3 Deep unfolding for online unsupervised correction of physical models used in channel estimation

Channel estimation is a very important step of the communication chain since it allows to choose appropriate precoders and to properly demodulate received signals. With a great number of antennas and/or subcarriers, channel estimation is a very challenging task. Fortunately, it can be greatly eased by the use of prior information. The prior information can traditionally be of two distinct natures:

- **Physical**: in that case, prior knowledge is drawn from the physics of wireless propagation. For example, it amounts to assume that the channel is a linear combination of a few steering vectors. Using a physical prior knowledge typically leads to the use of sparse recovery methods to carry out channel estimation. Physical prior knowledge is good since it can be used immediately and is shared by every user but relies heavily on assumption that may be violated in practice, due to hardware impairments or simplistic models.

- **Temporal**: in that case, prior knowledge is drawn from previous estimations of the channel, and it comes under the form of a prior distribution. Using a temporal prior knowledge typically leads to the use of Bayesian methods (MMSE and its variants) to carry out channel estimation. Temporal prior knowledge has the advantage of not relying on any simplistic physical model, so that it is relatively immune to hardware impairments. However, the channel distribution requires time to be estimated (so it cannot be used immediately) and is different for each user. Moreover, it requires memory to store past channels.

The proposed feature focuses on physical prior information, and, more specifically, tries to correct its potential defaults due to hardware impairments (such as imperfectly calibrated antennas as simulated here) and imperfect physical models. In order to do so, a recent deep learning technique called deep unfolding [GL10, MLE21] is used. It corresponds to considering an iterative algorithm as a NN, the parameters of which can be adapted to training data. More specifically, a sparse recovery method called matching pursuit [MZ93] is unfolded, resulting in a NN that can be initialised with an imperfect physical model that will be corrected by gradient descent, while the system encounters new channels to estimate.

The proposed NN is called mpNet, it takes as input noisy channels denoted **x** resulting from a least-squares channel estimation that go through several layers having the structure shown on Figure 4-8. The matrix **W** is the weight matrix of the network, it is initialised with a set of steering vectors (according to the available physical model), and the nonlinearity HT_1 corresponds to the hard thresholding operation that sets to zero all but the greatest entry (in modulus) of its input. The number of times the structure of Figure 4-8 is replicated corresponds to the number of matching pursuit iterations that are unfolded, which, in turn, corresponds to the number of estimated channel paths, and the output of the network is a channel estimate $\hat{\mathbf{h}}$. This number is allowed to be different for each realisation, so that mpNet is a varying-depth NN, which allows it to be SNR-adaptive (more iterations are required at high SNR). The network is trained online in an unsupervised way, with a cost function of the form $\frac{1}{2}\left\|\mathbf{x} - \hat{\mathbf{h}}\right\|_2^2$.

In summary, the main features of mpNet are its initialisation with a physical model, its ability to adapt to the SNR via its varying depth and its ability to be trained online thanks to its unsupervised cost function and low number of parameters (the matrix **W** being shared by all layers), which makes it intrinsically frugal.

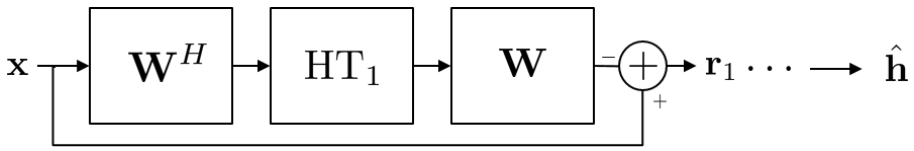


Figure 4-8: One layer of mpNet.

The proposed NN has been assessed on realistic synthetic channels drawn from the NYUSIM channel simulator [SR16], considering an Uniform Linear Array (ULA) with 64 antennas and a single subcarrier at 28 GHz, and the results are shown on Figure 4-9. On the figure, the blue curve corresponds to mpNet (initialised with an imperfect physical model comprising uncertainties about the gains and locations of individual antennas), the purple one corresponds to the use of an imperfect physical model as is, where uncertainties are not taken into account, the orange one corresponds to using simply least squares estimation, and lastly the red one corresponds to a physical model that is perfectly calibrated initially (unrealistic in practice). The main message of this experiment is that the resilience brought to mpNet by online learning makes it adapt over

time in order to optimise performance by correcting an initially imperfect physical model, even if the system features are suddenly affected by some incident (as is the case after 100k channels are estimated, where some antennas are suddenly broken, 10% on the left and 30% on the right), which is not the case of concurrent methods. In order to get more detailed explanations and experiments, see the associated paper [YL22].

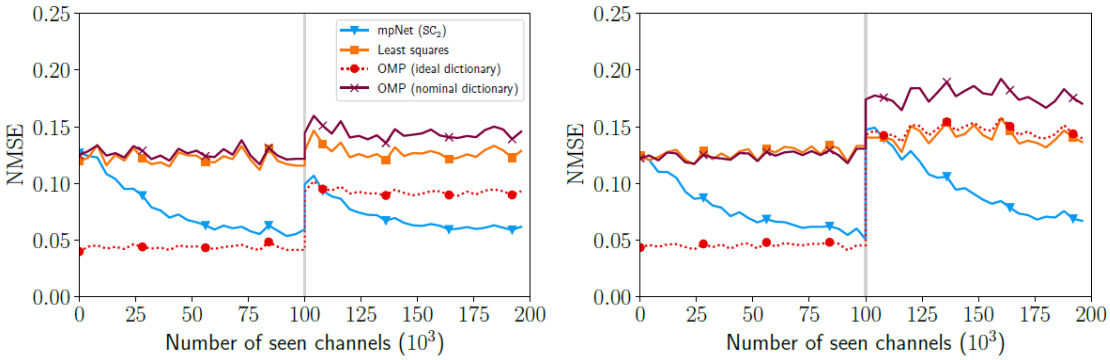


Figure 4-9: Comparison of mpNet to several baselines. On the left, 10% of the antennas (chosen uniformly at random) are broken after 100k channels are estimated. On the right, 30% of the antennas are broken.

This work is promising and requires several improvements. First, it should be extended to wideband channels, in order to assess the potential of mpNet to correct other hardware impairments that were not considered here, such as carrier frequency offset. Moreover, it should be tested in more realistic settings. This includes using a more realistic spatial user distribution than the uniform distribution of NYUSIM [SR16] and assessing the training time (instead of presenting results as functions on the number of seen channels) with a realistic spatial distribution of users. Moreover, a quantitative comparison in realistic conditions between the use of temporal and physical prior information (with and without correction with deep unfolding) would be very valuable from a practical perspective, in particular with other metrics more related to communication performance such as the BLER. One could also imagine a combination of the two kinds of prior knowledge within a single method, for example using physical prior knowledge to improve the channel distribution estimates that are used within Bayesian channel estimators.

4.2.4 Efficient channel charting

Channel charting [SMG+18] is a fully unsupervised learning task. Indeed, its objective is for a multi-antenna BS to build a low-dimensional map (called chart) of the radio environment based on uplink channel measurements, without requiring access to the users' actual locations. The chart should reflect as much as possible the physical reality, in the sense that the charting function should preserve spatial neighbourhoods. Predicting this way the relative locations of users from channel measurements has many potential applications, ranging from SNR prediction [KAS+20] and pilot reuse [RLD+20] to user grouping, proactive handover management or beam-finding (see [SMG+18] for more details on potential applications).

What makes channel charting particularly interesting compared to classical positioning methods is its fully unsupervised nature. Indeed, no link with the application layer in order to get locations from a Global Navigation Satellite System (GNSS) is required (even offline to build a dataset). Only channel measurements are needed, which are readily accessible from the RAN. Moreover, having access to the relative locations of users instead of their absolute locations is sufficient for most applications that need to assess the proximity of users. In that sense channel charting can be seen as an unsupervised alternative to radio maps [BLD+19].

In the seminal paper on channel charting [SMG+18], the raw second order moment of channels is used as input feature in order to reduce the influence of small scale fading which is irrelevant to the channel charting task. Such features have the disadvantage of being of dimension equal to the square of the channel dimension. On the other hand, it was more recently proposed to use the channel autocorrelation as input feature [GLL+20]. This has the advantage of yielding features of the same dimension as the channel that are also quite insensitive to small scale fading. However, using autocorrelations automatically makes the features translation invariant in the angular and delay domains, which is a potentially harmful property with respect to the channel charting task, especially for channels in LoS or comprising a dominant path. In contrast, the method proposed in this contribution does not require to square the channel dimension nor introduces any invariance in the angular or delay domain.

The general strategy of the proposed approach is to first design a distance measure between channels that preserves spatial neighbourhoods and then use it to compute chart coordinates via a nonlinear dimensionality reduction method. In order to set up the proposed method, a distance measure between channels which locally reflects physical reality (locations of the corresponding users) is proposed. It has the advantage of being insensitive to small scale fading and takes the expression:

$$d(\mathbf{h}_k, \mathbf{h}_l) = 2 - 2\frac{\left|\mathbf{h}_k{}^H \mathbf{h}_l\right|}{\|\mathbf{h}_k\|\|\mathbf{h}_l\|}$$

Once this distance is computed for all channels in the training data, it is used within a nonlinear dimensionality reduction method to obtain the chart. Isomap [TDL00] was the chosen method because of its ability to reflect geodesic distance (see [L21] for a precise introduction to the distance measure and its use with Isomap). The proposed method has the advantage of being much less complex than previously proposed ones.

The proposed method was empirically assessed on realistic synthetic channels taken from the Quadriga channel simulator [JRB14]. It was compared to methods proposed in [SMG+18], according to the continuity (CT) and trustworthiness (TW) measures. Both measures evaluate the capacity of the assessed charting methods to preserve neighbourhoods (of size K, which is a parameter), and are both between 0 and 1 (the higher the better, 1 meaning a perfect preservation of the K nearest neighbours around each training point). Results are shown on Figure 4-10, where it can be seen that the proposed method performs similarly or better than the baselines (especially for the continuity measure), while being more computationally efficient. These results are promising and show the potential of the proposed distance measure for carrying the channel charting task in a frugal way.

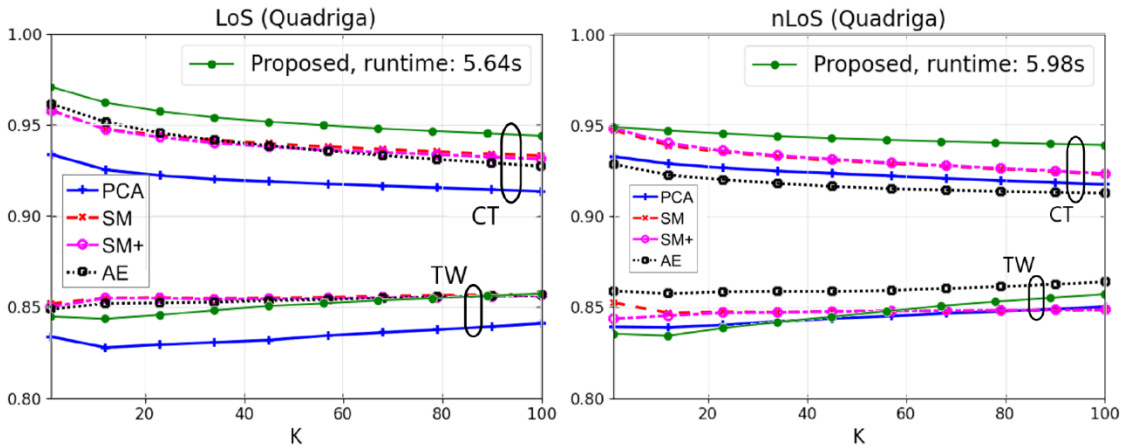


Figure 4-10: Comparison of the proposed approach to several baselines on channels generated with the Quadriga channel simulator. Continuity (CT) and trustworthiness (TW) measures are given (the higher the better) as a function of the size of the considered neighbourhood K.

However, the presented method is only a first step. Indeed, it could be used to guide the structure of a NN and to initialise it. This would allow the method to treat channels in a sequential way and to be adapted online. By the way, it has recently been proposed to incorporate the timestamp information in a channel charting pipeline, in conjunction with a triplet loss, leading to impressive results [FDO+21]. Inspired by this approach, one could, for example, imagine combining a model based structure inspired by the proposed distance measure with a triplet loss learning to enhance performance. The structure would yield frugality to the approach, while triplet loss allows to use the timestamp information, which potentially greatly enhance performance.

4.3 Learning semantics: An Opportunity for Effective 6G Communications

This section presents a study on semantic communications [CB21], [SC21] a new paradigm envisioned as a key enabler of future 6G networks. In particular, while in classical Shannon's information theory the goal of communication has long been to guarantee the correct reception of transmitted messages irrespective of their meaning, semantic communications focus on transmitting only relevant information, i.e., the one sufficient for the receiver to capture the meaning of a message. In their seminal work [Sha48], [Wea53], Shannon and Weaver identified three levels of communication: (i) Level A - the technical problem: how accurately can the symbols of communication be transmitted? (ii) Level B - the semantic problem: how precisely do the transmitted symbols convey the desired meaning? (iii) Level C - the effectiveness problem: how effectively does the received meaning affect conduct in the desired way?

In this study, level B is the focus. Relevant KPIs/KVIs are energy efficiency, reliability, and inference accuracy. To this end, a novel architecture incorporating level B to classical level A communications is presented. This architecture enables representation learning of semantic symbols for effective semantic communication and is exploited in a first study involving text transmission, with a numerical example in which the sender and receiver may speak different languages. In this architecture, information from a binary source is encoded with semantic information extracted using neural attention mechanisms [VSP+17], to produce a sequence of semantic symbols. In contrast to very recent state-of-the-art works [XQL+21], [WQL20], which propose an E2E system for semantic text and speech transmission, a new loss function is defined here, to capture the effects of semantic distortion to communication. This new perspective can save significant communication bandwidth through the concept of semantic compression. A semantic message is a sequence of symbols learned from the "meaning" underlying data, which then have to be interpreted at the receiver. This enables to dynamically trade semantic compression losses with *semantic fidelity* [BBD+11] (i.e., the semantic interpretation correctness).

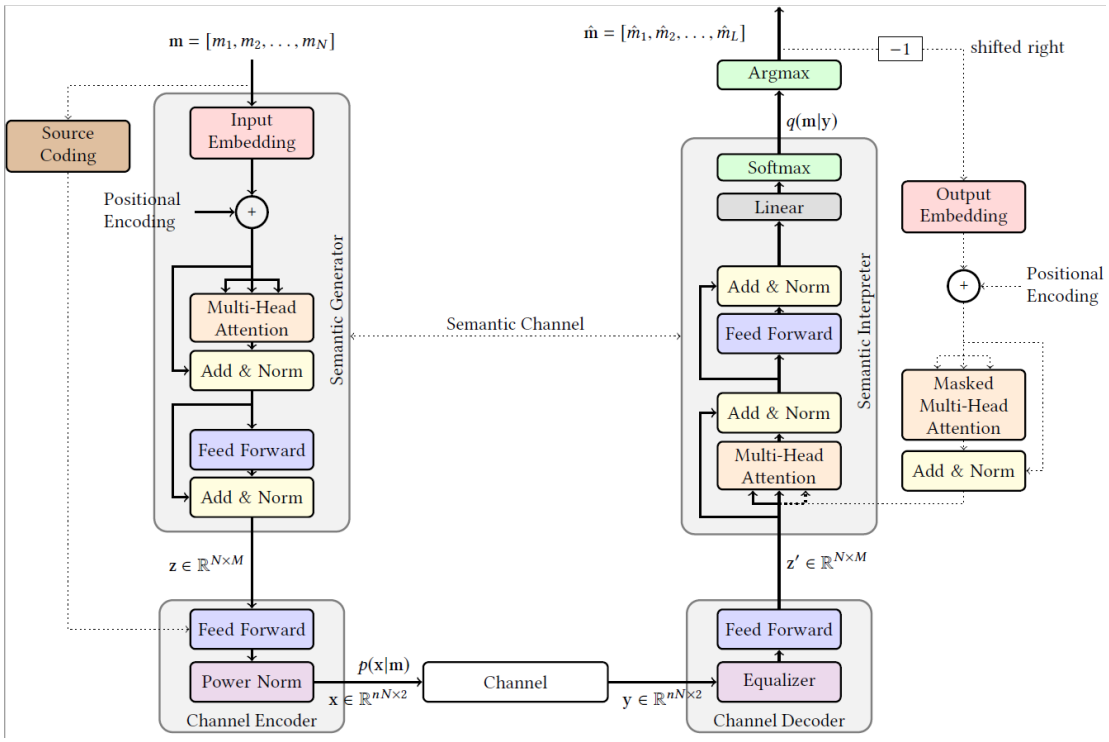


Figure 4-11: Transformer-based semantic communication system architecture [SC21].

Figure 4-11 shows the proposed E2E semantic communication system, composed of a source coder $S(\cdot)$, a semantic generator $G(\cdot)$, a channel encoder $E(\cdot)$, a channel decoder $D(\cdot)$, and a semantic interpreter $I(\cdot)$. A detailed description of the architecture is provided in [SC21]. However, the main focus is on the semantic encoder (based on a multi-head attention block [VSP+17]), which maps an input sequence into new symbols belonging to a semantic representation subspace. The multi-head attention is a technical solution that can extract specific characteristics of inputs sequences. More details can be found in [VSP+17]. Inversely, the semantic interpreter (also based on a multi-head attention network) decodes the semantic message. For more details on the technical solution, the interested reader is referred to [SC21].

Numerical simulations are performed in the context of Natural Language Processing (NLP), especially when the sender and receiver speak a different language. In this context, messages are formed and communication parameters are set to maximise the correct interpretation of semantic messages rather than error-free bit decoding at the receiver. Two performance metrics are considered: (i) Average transmission rate (bits/s), i.e., a classical communication related performance indicator; (ii) A measure incorporating the accuracy vs. complexity trade-off, defined as the ratio between the semantic correctness (i.e., the complementary of the semantic error), and the number of symbols used to encode the message. While the definition of the latter is straightforward, the semantic error takes different forms depending on the context [KP20] (e.g., mean square error, cross-entropy or Bilingual Evaluation Understudy (BLEU) score in NLP [PRW+02]). Numerical results are reported for text transmission as in [XQL+21]. The reference scenario considers a transmitter communicating with a receiver by sending a block of sentences (sequence of words) through the wireless channel, using the previously described semantic communication system. To this end, the transmitter learns to map each word to a sequence of semantic symbols that the receiver has to interpret. Note that such a mapping is learned from the data available at the source. Hence a word can have different symbols representation depending on the sentence it belongs to and the underlying meaning conveyed by both the word and the sentence. This is in contrast to traditional Level A communication, where each word is always

mapped to the same symbol. In the considered scenario, once received symbols are interpreted back to words, the transmission accuracy in terms of BLEU score is measured, which counts the difference of words (or group of words - n-grams,) between the intended sentence and interpreted one [PRW+02]. To provide an example, "1-gram" means comparing word by word, "2-grams" means comparing groups of two words by two words. The value of the BLEU ranges from 0 to 1, with 1 indicating that the interpreted message is the one as the reference. The dataset from the Tatoeba Project (translation from English to French data available at http://www.manythings.org/anki/) has been used for the simulations. In Figure 4-12, the impact of the SNR and the source entropy ($H_{\mathcal{M}}(M)$) on transmission accuracy is shown. The source entropy is changed by modifying the distribution of source messages. In Figure 4-12, it can be observed that the performance slightly decreases when the entropy increases since there is more information to convey to the receiver. Also, the proposed scheme achieves a BLEU score of 1 for SNR $\geq$ 5 dB.

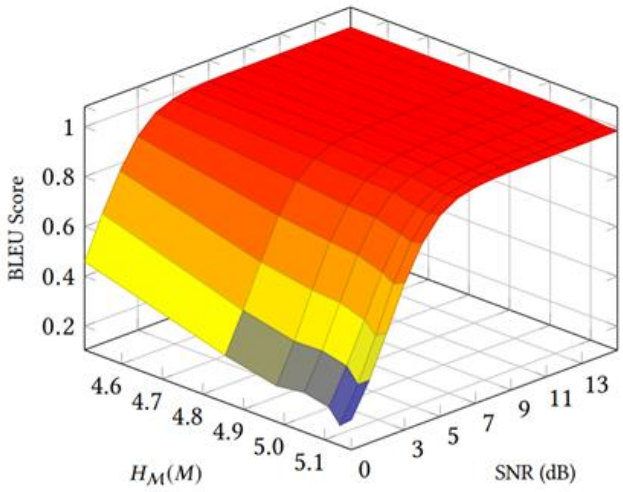


Figure 4-12: Impact of the SNR and HM(M) on the accuracy. Here we use n = 6 symbols/word over AWGN channel [SC21].

Impact of languages mismatch. The next result assesses the performance in a scenario where the transmitter speaks French and the receiver must understand in English. In this case, the sender and the receiver have different alphabets. This further introduces complexity in symbols interpretation. Indeed, many words in French are written the same way in English leading to semantic ambiguity. The result is 30% decrease in BLEU score performance as show in Figure 4-13. In the same figure, the performance of the classical approach using Huffman/6-bits coding and a 64 QAM modulation are also shown as benchmarks. The proposed semantic communication clearly outperforms the two benchmarks, especially in the low SNR regime.

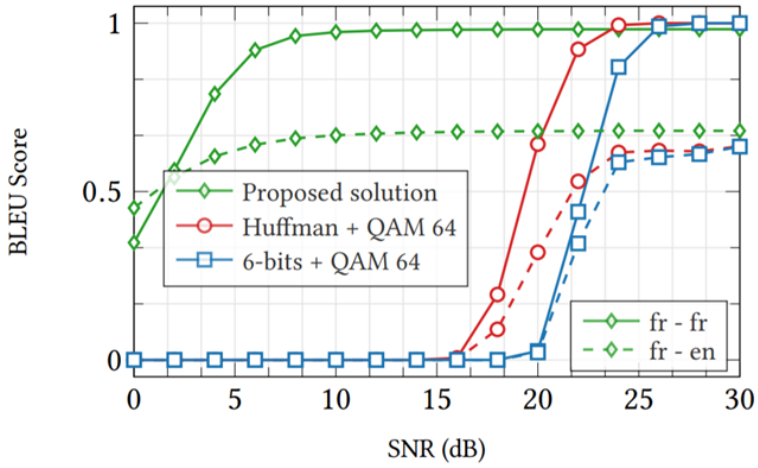


Figure 4-13: 1-gran BLEU Score vs. SNR in the context of AWGN channel French-to-(French/English) translation [SC21].

5 Privacy, security & trust in AI-enabled 6G

In recent years, the usage of ML on massive amount of data has been increasing. Trustworthiness in AI/ML becomes critical for AI-enabled 6G because AI/ML-based decisions are done for autonomy of communication and detection of cyber-attacks. Thus, possible threats to AI/ML need to be studied considering the possible trust relations within the systems. Such threats can be, for example, poison attacks, privacy attacks, the attacks that prevent expansibility etc. Note that the trustworthiness for AI/ML supports the trustworthiness of the overall system, but some other assurances for the overall trustworthiness of system is required.

5.1 Federated Learning and Privacy

To utilise huge amount of data collected from distributed devices, learning a model in a collaborative way is becoming a great demand. Collaborative ML approaches differ in how to use data and model, and trust distribution between entities. One of the most adopted methodologies for collaborative learning is Federated Learning (FL) [KMR15] which is a privacy-friendly mechanism to generate the ML model jointly between clients and server. Each client performs training on their local data where local training results, so-called local model updates, are transferred to the server, so training data never leaves the clients. The server generates the global model by aggregating the local model updates. On the other hand, recent studies demonstrate sophisticated privacy attacks against FL by exploiting the observed gradients/local model updates or using the collected inference results. Privacy attacks to FL, such as membership inference [SSS+17], attribute property inference [MSC+18], deep leakage [ZLH19], and model extraction [TZJ+16] may be posed by a malicious client or the server trying to infer sensitive information during training or inference phase. The adversarial goal of privacy attacks is to gain more information about the training data, FL setting and the model parameters. In this section, some prevention mechanisms against these types of attacks are presented. Furthermore, an approach for privacy preserving clustering is discussed along with some preliminary experimental results.

5.1.1 Privacy preserving clustering: Federated fuzzy c-means

Some of the challenges of the FL scenario have been analysed in [BMR+21], in the framework of clustering algorithms. Specifically, a federated fuzzy c-means (FCM) clustering algorithm has been proposed. The proposal consists in collaboratively learning a global model according to an iterative procedure: at each round, data owners send aggregated statistics evaluated on local data to the server, which is in charge of updating cluster centres and transmitting to the data owners for the next round. Furthermore, it has been shown that such federated version achieves the same results (i.e., final centroids referred to as V_{fed}) obtained by the classical clustering algorithm applied to the overall merged datasets (i.e., final centroids referred to as V_{sum}), while preserving privacy of data owners.

In the experimental analysis, the following scenario has been considered: 20 participants are involved in the federation; furthermore, the impact on the results of a parameter γ has been assessed: it represents the fraction of participants to be sampled randomly at the beginning of each FL round. Figure 5-1 reports the results obtained on a synthetic 2D dataset (xclara), under the assumption of i.i.d. data partitioning across clients: the evolution of the Frobenius norm of the difference in the cluster centres between consecutive rounds is reported (left) along with the Frobenius norm of the difference between cluster centres computed by federated and centralised versions of FCM (right). By comparing the output partitions with the available ground-truth labels, it has been verified that such deviation in the cluster centres does not induce a significant

variation of the output clustering, thus highlighting that the clusters generated by federated and centralised versions are very similar even when only a fraction of participants is involved in the FL procedure.

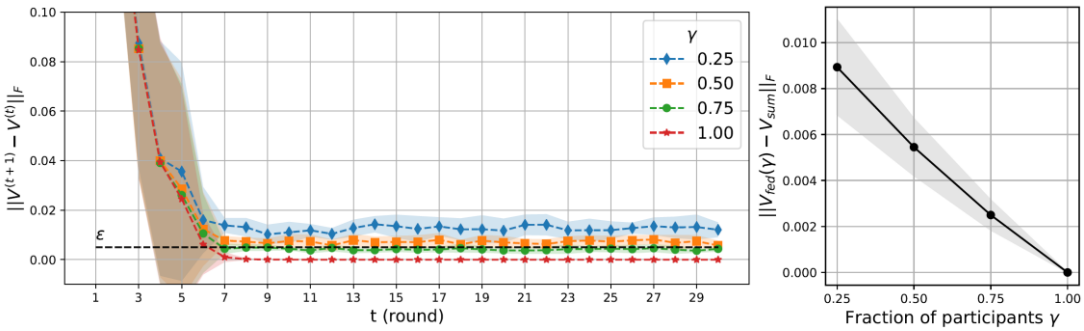


Figure 5-1: Results on xclara dataset. Average values (shaded region indicates the standard deviation). (left) Federated FCM: Frobenius norm of the difference in the cluster centres between consecutive rounds; (right) $\|V_{fed}(\gamma) - V_{sum}\|$ over γ.

5.1.2 Differentially Private Federated Learning

Integrating the intelligence to enable network automation improves UE experience and supports enhanced network decisions. Predicting Reference Signal Received Power (RSRP) can reduce the signalling overhead and enable proactive network actions. ML can be used to predict RSRP values using geographical location information of UEs, however location information is regarded as sensitive and may not always accessible for centralised ML model training. FL algorithm aims to solve this challenge via coordinated learning task among the clients without revealing their local dataset to a central entity. Although FL is a privacy aware framework, there are still privacy issues like membership inference attacks [SSS+17], attribute property inference attacks [MSC+18] during inference phase. As countermeasure, existing privacy enhancing technologies including Differential Privacy (DP), Homomorphic Encryption (HE), and Secure Multi-Party Computation (SMPC) (see D4.1 section 4.2.1 for more detail about DP, HE, and SMPC) are investigated in the literature. In this study, a privacy enhancing technique, DP, which is a data anonymisation technique that introduces a level of uncertainty into the released model to hide any individual user contribution [DMN+06] has been implemented. Our framework aims to prevent inference phase attacks by untrusted users who can query the model using inference interface and try to extract sensitive information from model inference results.

Definition of (ε, δ)-DP: A randomised function M is ε-differentially private if for any subset of the output S in the range of M, and for all data sets D_1 and D_2 differing in a single entry [DR14] as depicted in Figure 5-2:

$$Prob\ [M(D_1) \in S] \leq exp\ (\varepsilon)\ Prob\ [M(D_2) \in S] + \delta \quad \text{(Eq. 5-1)}$$

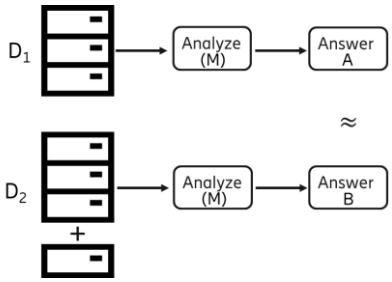


Figure 5-2: Illustration of the idea of differential privacy.

The parameter ε in equation 5-1 is the control parameter for privacy level, also called privacy budget. The parameter δ limits the probability of information accidentally being leaked, and chosen as the less than the inverse of the data size as a best practice. In FL framework, D_1 and D_2 refers to UE's training datasets.

RSRP Prediction via Differentially Private FL

Signal quality prediction is an important feature which can be achieved by the intelligent and proactive management of the network resources. To generate an RSRP prediction framework, we used UE location information in the learning task as the input of the learning model and we obtain the training label from CSI-RS reference signal.

In our FL setting, UEs are regarded as clients and gNB is regarded as FL server. During the FL learning, an initial global model is created and shared with the participating UEs. Using inital parameters, each UE locally trains its model by mapping its location data to RSRP values. Then, each UE sends the trained model parameters to the gNB. gNB collects and aggregates the parameters using FedAvg algorithm and sends back the resulting model parameters to the UEs, which contains implicit mapping of all UEs to RSRP, embedded by local training.

The integration of DP mechanism is conducted by gNB by perturbing the averaged updates using a randomised Gaussian mechanism during learning iterations which is called central-DP. The purpose of the randomisation process is to hide each UE's contribution during inference process.

The implementation of the framework is realized in Tensorflow environment. Tensorflow Federated (TFF version 0.18.0) framework is used to carry out federated computations on distributed data. The NN is created by using the sequential NN in Keras comprised of an input, an output and 3 hidden layers including 10 neurons each. For DP integration, Tensorflow Privacy library (version 0.5.1) is used. The library gives the opportunity to set different privacy levels by adjusting privacy hyper parameters noise multiplier (*z*), gradient clip-scale (*c*) and δ.

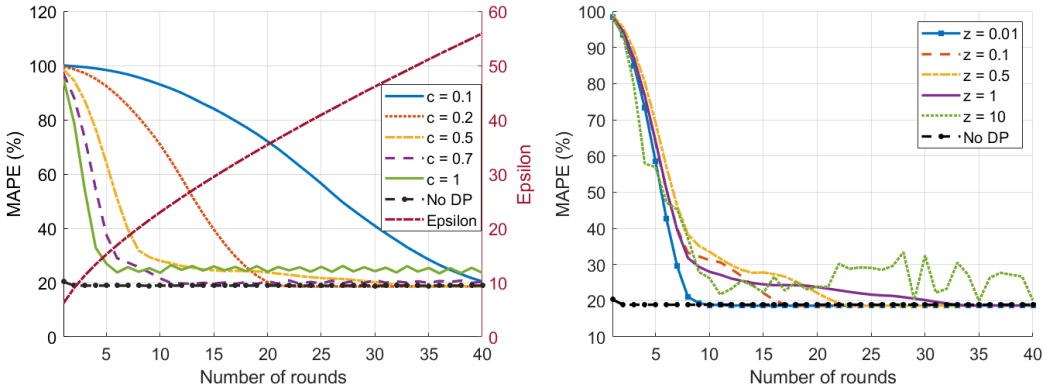


Figure 5-3: Left: Training evaluation loss for different clip-scale values (c) when noise multiplier, z=1 and batch size, B=100. Right: Training evaluation loss for different noise multiplier values when B=100 and c=0.5.

Figure 5-3 shows the evaluations of the implementation by means of Mean Absolute Percentage Error (MAPE) and epsilon (ε) to show the impact of *c*, batch size *(B)*, and *z* during the FL training. In Figure 5-3, on the left side it is demonstrated how changing the clip scale, *c*, impact the training evaluation loss in terms of MAPE over FL rounds when noise multiplier, z=1 and batch size, B=100. Small gradient clipping, e.g., c={0.7, 1} results in faster convergence but higher error with oscillations, because not only the noise variance increases, but also the gradients' sensitivity

to the noise will not be same in different clients. Therefore, the clip scale is chosen depending on the stability and convergence rate. Further, during the training, as the number of rounds increases, i.e., as MAPE decreases, ε increases meaning that privacy decreases. Higher clip-scale values accelerate the convergence, i.e., requires less rounds, thus results in stronger privacy (lower ε), but clip-scale should have an upper bound for sensitivity aspects.

Figure 5-3 right figure demonstrates that MAPE increases with the increasing noise multiplier, *z*. The accuracy loss resulting from DP integration can be observed by comparing with No DP results given in black dotted line. This represents the trade-off between utility and privacy. Targeting stronger privacy comes at the cost of accuracy reduction, on the other hand this should be adjusted to preserve the convergence e.g., if noise multiplier is set as too high, then the training will not converge as in the settings for $z \geq 10$.

In conclusion, this work presents a privacy preserving federated learning for the RSRP prediction framework. The work focuses on two important aspects: (i) performing a local training by using geographical location of UEs as a feature to predict RSRP (ii) providing privacy guarantee by implementing DP to protect the FL framework against inference attacks.

5.1.3 Security mechanism friendly privacy solutions for federated learning

FL is widely used to address the privacy concerns on the distributed UEs data. Although the private data is not sent in cleartext to the server, there are some studies, [KMA+21, BDM+20], that show the local model updates that are sent to the server may leak some information about the private data of the clients. While the privacy enhancing technologies solves the privacy problems, these techniques may make it difficult to protect the AI model construction against malicious data providers (UEs) and prevent security attacks against the model construction, such as backdoor and poisoning attacks. The reason is that the server cannot analyse the data coming from the clients because the data is not in clear text format, and the server can learn only the aggregated result. Thus, it is important to develop privacy solutions which are security mechanisms friendly, i.e., the privacy solutions that allow the execution of security solutions for AI. Thus, we focus on security mechanism friendly privacy enhancing solutions. One approach to provide privacy is to anonymise the owner of the local model updates as proposed in [BDM+20]. To be able to provide anonymisation, multi-hop communication between clients and server can be utilised [BDM+20]. In this case, there can be some security attacks by malicious clients, such as sending multiple local model updates in one round of FL or dropping/altering legitimate clients' local model updates. Our aim is to investigate possible solutions to mitigate these types of attacks. In our trust model, the server is a malicious party who wants to learn about training data of clients; also, the clients are considered as malicious parties who may try to disrupt the global model and also learn about other clients' data. Usage of blind signatures is a good candidate to ensure that the local model updates are coming from the specific clients and the clients can only send one local model update in each FL round. In each round, the server blindly issues one-time usage certificates for clients and then the clients send their local model updates signed using their private keys. When the server receives a local model update, the server checks the signature on the local model update, validates the certificate of the client and if all these verifications are successful and the certificate has not been used before then the server accepts the received local model update and stores the certificate in a table for upcoming checks. Since the server had issued the certificate blindly, the server will not be able to identify the owner of the local model update. To make the identity of the local model updates anonymous, the next hops should be chosen randomly in the multi-hop communication. To limit the number of hops in the communication, a counter such as TTL (Time To Live) or hop limit, which is a mechanism to limits the forwarding of the local model to other

clients within specified time or pre-defined hop-counter) can be used. But not to break the anonymity against other clients, the initial value of the counter should also be chosen randomly. With this solution, malicious clients will not be able to send multiple local model updates and also will not be able to alter other clients' local model updates. Figure 5-4 shows the example interaction between clients and the server.

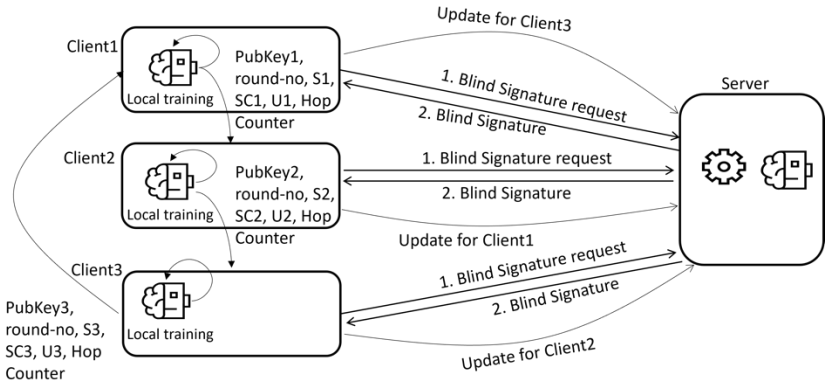


Figure 5-4: Security-friendly privacy preserving federated learning scheme.

In our approach the public and private key pairs are generated for clients in each round of FL where these public keys are blindly signed by the server and the current round number of the FL is included in the signature. Thus, the server does not know which public key belongs to a client due to blindly signing process. Also, the server will not learn the owner of the packets (model updates) since they are sent in a multi-hop communication manner to the server, but it learns whether the model updates come from the legitimate clients. Since the server is able to check whether the used public key in the signature is utilized before or not, it can prevent a client to send more than one local model updates. The random hop counter is included in a packet by the owner of the packet, so that the client who receive the packet cannot recognise whether the sender is the owner of the packet. None of the clients who receive the packet can modify the packet because it is signed by the owner of the packet.

The computational overhead to the normal FL steps is as follows. For each round of FL, the clients need different public and private key pairs that can be generated offline which does not bring any online complexity overhead. Blind signature and model signature are two additional operations to be done. Thus, the server needs to compute n blind signatures, n blind signature verifications, and n normal signature verifications. For the clients, the computational overhead is computation of two blind signature related operations and one local model signature operation. For the communication, each client needs to connect to the server to have a blindly signed public keys and then instead of sending signed local model updates directly to the server, they are sent hop-by-hop.

5.2 Explainable AI

Explainable AI (XAI) is gaining a significant momentum in the context of wireless networks and is expected to be a crucial tool to achieve users' trust in AI-enabled 6G. Explainability is typically assessed both at *global* and *local* level: the former refers to the structural properties of the model and is related to its transparency; the latter is associated with the inference process and aims at providing details about the decision made for any input instance. In this section, the adoption of

XAI models is discussed in the context of Quality of Experience (QoE) prediction and radio network control.

5.2.1 XAI models for QoE prediction

The adoption of XAI models has been investigated for the task of QoE classification: in fact, in current and future wireless network it is not only crucial to maintain high levels of QoS metrics, but also to ensure the fulfillment of QoE metrics, intended as a measure of the end-user satisfaction. As a preliminary step, the performance of tree-based classifiers with different levels of interpretability has been analysed. Specifically, an experimental analysis has been carried out to evaluate the performance of fuzzy and classical Decision Trees (DTs) and Random Forest (RF) on a QoS-QoE classification dataset [RDG+21]. The QoS-QoE dataset [VLP+18] has been generated by resorting to a fully controllable environment for multimedia streaming simulation and collects information about 69129 video streaming sessions. The target task consists in deriving a mapping between the QoS metrics (29 features) and the StallLabel variable (a QoE factor, indicating that the related streaming session included SevereStall, MildStall or NoStall events).

As per Fuzzy DTs (FDT), the Multi-Way FDT (MFDT) described in [SMP17] has been adopted. An example is provided in Figure 5-5 (left). Concepts from fuzzy set theory are used for partitioning input attributes. An example of a strong triangular fuzzy partition is reported in Figure 5-5 (right): the partition of the generic attribute A_f consists of three fuzzy sets ($A_{f,1}$, $A_{f,2}$, $A_{f,3}$).

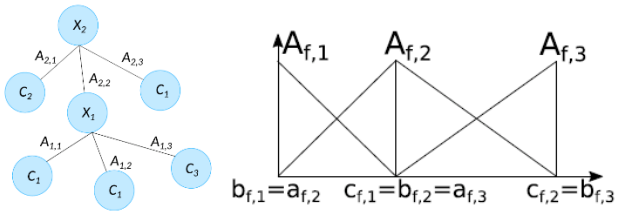


Figure 5-5: (left) Example of fuzzy multiway decision tree.
(right) example of strong triangular fuzzy partition on attribute A_f.

DT induction is obtained through a hierarchical partition of the feature space based on the training data [RDG+21]. In MFDTs, the number of branches from an inner node is equal to the number of fuzzy sets in the partition of the input attribute selected in the node. In the example of Figure 5-5, indeed, each inner node has exactly three child nodes. Whenever a stop condition is met (e.g., maximum depth) a leaf node is created, reflecting the class distribution of objects therein. The *global* interpretability can be expressed in term of complexity of the model, e.g., the number of nodes/leaves. It is worth highlighting that the MFDT implementation has been adapted to enhance also *local* interpretability by purposely tuning the inference strategy (maximum matching) and the fuzzy partitioning (strong triangular uniform).

The following models have been considered:

- Two MFDT models with maximum depth set to 3 and 4 (referred to as MFDT-3 and MFDT-4, respectively);
- Two Binary Decision Tree (BDT) models based on the scikit-learn Python implementation, with maximum depth set to 6 and 11 (referred to as BDT-6, BDT-11), to obtain a complexity comparable to MFDTs in terms of number of nodes)
- A RF model (scikit-learn Python implementation).

The maximum number of fuzzy sets in the fuzzy partitioning procedure of MFDT has been set to 5: this ensures a high level of semantic interpretability thanks to a straightforward labelling of the

fuzzy sets. Indeed, in the limit case of five fuzzy sets, the following labels can be used: VeryLow, Low, Medium, High, VeryHigh. Models have been evaluated using 5-fold cross-validation. In the following, the results obtained after rebalancing through random undersampling are reported (target distribution: {NoStall: 2000, MildStall: 2000, SevereStall: 635}).

Table 5-1: Experimental Results: performance comparison between different tree-based models on the QoS-QoE dataset. Average values.

	F1-measure		Model Complexity		No Stall			Mild Stall			Severe Stall		
	Training	Test	Leaves	Nodes	F1	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall
MFDT-3	0.786	0.812	115.0	143.6	0.871	0.940	0.813	0.686	0.592	0.817	0.555	**0.478**	0.661
MFDT-4	0.817	**0.834**	396.0	500.8	**0.891**	0.942	**0.845**	**0.707**	**0.632**	0.802	**0.559**	0.451	0.761
BDT-6	0.851	0.811	***55.0***	***109.0***	0.873	**0.950**	0.808	0.686	0.592	**0.819**	0.524	0.381	**0.840**
BDT-11	**0.945**	0.803	278.4	555.8	0.869	0.931	0.815	0.661	0.582	0.767	0.515	0.379	0.812
RF	***1.0***	***0.856***	43911.2	87722.4	***0.907***	***0.954***	***0.865***	***0.745***	***0.674***	***0.835***	***0.629***	***0.496***	***0.870***

DTs (crisp and fuzzy) are generally less accurate than RF classifier, which, as an ensemble model, does not feature inherent interpretability. MFDTs achieve comparable or better performance than BDTs in terms of micro-average F1-measure; the performance drop w.r.t. RF is just around 2% for MFDT-4 and 4% for MFDT-3. Due to re-balancing, all models obtain reasonable recall on both *MildStall* and *SevereStall* classes.

Finally, to assess the semantic interpretability associated with the induced decision trees, two examples of rules are reported in the following, extracted from MFDT-4 and BDT-6, respectively.

Table 5-2: "If-then" rules extracted from decision trees MFDT-4 and BDT-6.

Rule extracted from MFDT-4	Rule extracted from BDT-6
IF 100 InterATimesReq is VeryHigh AND 25 InterATimesReq is VeryHigh AND TCPInputPloss is VeryLow THEN StallLabel is SevereStall	IF StdInterATimesReq > 1.30 AND 25 InterATimesReq > 0.86 AND StdInterATimesReq > 1.59 AND 25 InputRateVariation > 186749.00 AND TCPOutputJitter > 0.00 AND 90 InputRateVariation > 473853.50 THEN StallLabel is SevereStall

Depending on the audience, the linguistic representation of numerical variables possibly makes the rules extracted from MFDT-4 easier to interpret than the ones extracted from BDT-6.

With regard to the relevant KPIs/KVIs, BDTs and MFDTs offer a high level of interpretability compared to the ensemble approach (RF). At the same time the average F1-measure drop is quite limited and within 10%. Future developments will entail the evaluation of supervised XAI models in the federated setting.

5.2.2 XAI for radio network control

Changing perspective towards AI for automated radio network control, a promising solution is XAI, whose goal is to investigate tools and techniques aimed at opening the so-called opaque (or

black-box) models (e.g., DNNs) or at devising intrinsically interpretable and accurate models (e.g., rule based systems). This concept can be used to explain and distinguish the effect of network configuration and user device load to network performance KPIs. The main technical difficulty is that network configuration (e.g., policy of allocating radio resources) affects the user device load, while the load itself has a more direct effect on the network KPIs, hence XAI will primarily find the importance of the load and explain the predicted KPI based on load, mostly ignoring the explanation of how network configuration and control affects the KPI.

Our new XAI model is based on SHAP (SHapley Additive exPlanations) and a hierarchical regression model for KPI prediction. We seek potential future applicability of out method to automated network control. In our research, the primary WP4 KPI (as identified in D4.1) to be improved are flexibility, data quality, complexity gain, generalisability and deployment flexibility:

- Flexibility and Generalisability: XAI results can be interpreted as a general knowledge, incorporated in arbitrary settings as expert knowledge.
- Data quality: We can filter outliers and data errors by a pointwise explanation approach, understanding model prediction and error for the outlier data points.
- Complexity gain: The final model can be very simple by focusing on the key concepts rather than artefacts of the training data.
- Deployment flexibility: tree-based regression models can be generated directly as program code; no need for ML or deep learning frameworks.

Our data consists of three multidimensional sets of variables C: Control, L: Load, and the network KPIs.

We build a prediction f(C,L) for a selected network KPI in the form of a ML model, typically Gradient Boosted Tree regression or classification.

Our method to separate the effect of load from the prediction to explain the effect of the Control is based on Shapley explanations [SHAP]. The original SHAP value is defined as $v_{f(C,L)}(S)$ where S is a subset of the variables in C and L and the value v measures the contribution of the set S to the prediction f(C,L). SHAP is calculated by considering subsets of S and subtracting their contribution to the model.

Asymmetric SHAP [AsymSHAP] is a method well suited for our task, since we are free to modify the order of the weights while computing SHAP for subsets, thus we can prioritize C over L. The main observation in Asymmetric SHAP is that certain coalitions S might not be compatible with the entire data set; for example, certain control settings might produce Load distribution significantly different from the global one.

We combine SHAP and mutual information explanations by contrasting Asymmetric SHAP explanations against the mutual and interaction information of subsets of C and L. We also use considerations of mutual information to generate hard explainability examples. The method is descried in full detail in [KVB2022].

We deploy Gradient Boosting Tree regression [GBT] for prediction to obtain f(C,L). In future experiments, we will also incorporate NN.

The first experiments were conducted on pre-Hexa-X EHU 4G mobile radio network data [KVB2022]. The first publication draft also includes generated data where several functional relations are tested and evaluated between C, L and f(C,L), including linear, multiplicative, and more difficult relations. Most difficult relation is for example f(C,L)={C+L} where {.} denotes the fractional value. For an abstract example where C and L are independent uniform in [0,1], the fractional value is independent of both C and L and the individual effect of C and L on the model

are equal. Functions involving the fractional or modulo value or the parity are tested and compared to the real data to identify similar functional relations between Control and Load.

Experiments on the performance management (PM) data from radio access network cells with 15 minutes granularity give models for the average downlink cell throughput. Input features of the model are categorised as described in Figure 5-6.

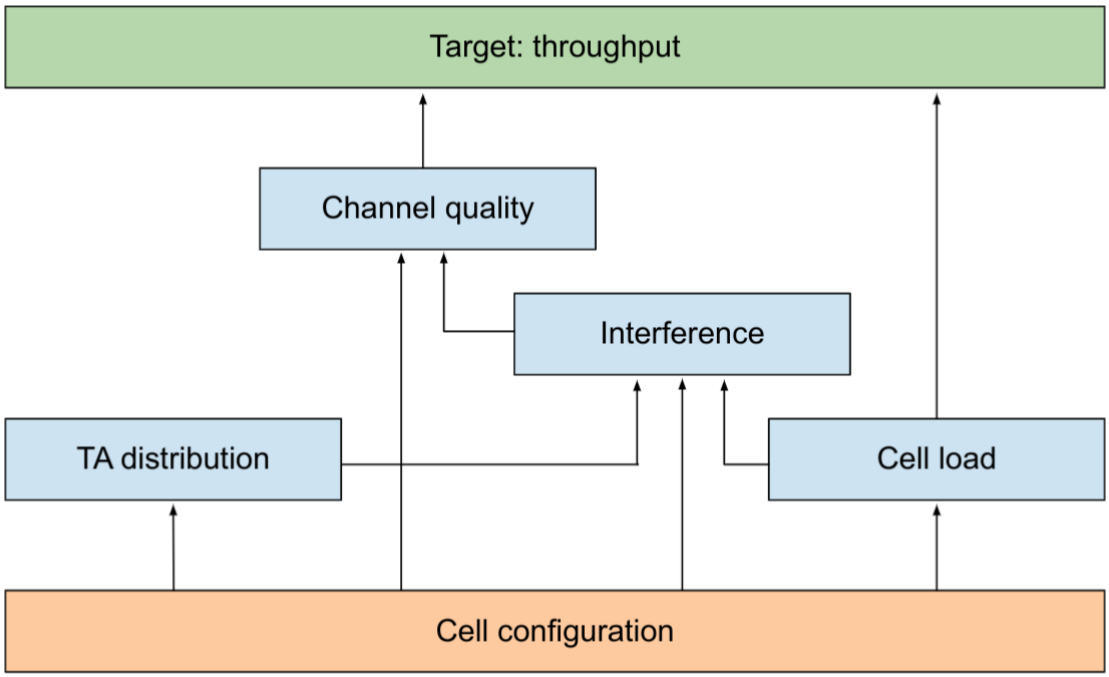


Figure 5-6: Causal structure of the different feature groups in our use case.

We compare feature attribution of the baseline SHAP method and two of our new variants. The new variants receive the casual ordering as input. Here we have a choice regarding the order of TA distribution and cell load, and we can also average the measured importance in the two methods. In Table 5-3, we can observe a substantial difference between the results of the approaches. Both new methods place an increased importance on TA distribution compared to the baseline model. Our first model is encouraged to use features that appear earlier in the causal ordering but is not forced to do so and can unlearn the effect of early features in later rounds. The second method trains separate models in casual order; this method attributes very strong influence on TA distribution, which is the root cause of performance degradation.

Table 5-3: Average of the absolute value of feature attribution made by different methods on EHU 4G mobile radio network data.

	TA distribution	Cell load	Interference	Channel quality
Importances when *Cell load* ordered before *TA distribution*				
Causal-order training	0.2093	3.6073	0.7939	2.9326
Separate models	0.4714	4.4750	0.9445	2.9343
Importances when *TA distribution* ordered before *Cell load*				
Causal-order training	0.4317	3.4371	0.7808	2.9383
Separate models	1.3429	4.1927	0.9344	2.9463
Importances when contributions averaged over causal orderings				
Regular model	0.1658	3.6426	0.8157	3.1973
Causal-order training	0.3205	3.5222	0.7874	2.9354
Separate models	0.9072	4.3339	0.9395	2.9403

As part of future work, we will replace the preliminary non-Hexa-X high protocol level 4G radio data by low level radio data from Hexa-X.

5.3 Design and implementation of Fed-XAI algorithms

FL of DTs and Rule-Based Systems (RBSs), widely recognised as XAI models, requires ad-hoc procedures since their learning stage is generally not based on the optimisation of a differentiable global objective function. Thus, the Federated Averaging (FedAvg) protocol, typically used in the context of NNs, is not immediately amenable. Two possible approaches can be devised for learning DTs and RBSs in a federated manner under the orchestration of a central server. As per the first approach, data owners do not build a local model, but at every step they send some aggregated statistics evaluated on local data to the server so that it can progressively build a global model. As per the second approach, each data owner builds a model based on local data and shares it with the central server, which merges the received models to produce a global, *federated*, model. This approach entails a one-shot communication scheme and not an iterative, round-based, algorithm; furthermore, it requires defining appropriate procedures for model aggregation, necessarily different from the classical FedAvg.

An overview of the latter approach is provided in Figure 5-7.

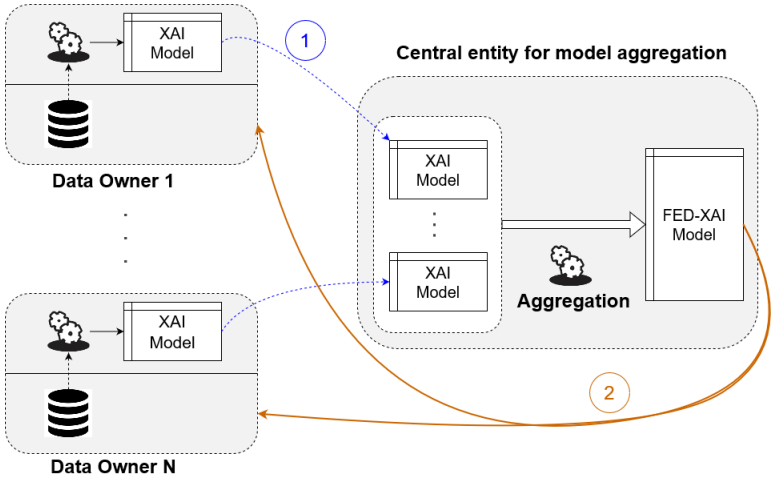


Figure 5-7: Illustration of Federated Learning of XAI models.

Both DTs and RBSs can be represented as collections “IF antecedent THEN consequent” rules. Notably, this representation is completely generic as it is applicable regardless of the target task (regression or classification) and the type of the attributes (e.g., nominal or numeric).

The aggregation procedure consists in juxtaposing rules collected from data owners and resolving possible conflicts. A conflict emerges when rules from different models have the same or similar antecedents, i.e., they refer to identical or overlapping regions of the attribute space, but with different consequents.

As for 6G KPIs and KVIs, the proposed solution targets inferencing accuracy and explainability, respectively.

5.4 Detection and classification of cybersecurity anomalies in 5G network

Computer security is a major issue today, indeed it is estimated that nearly 978 million people in the world are affected by cyberattacks each year (Ministry of the Interior). To prevent cyberattacks from becoming major, several strategies exist. Risks can be prevented with a firewall or antivirus, but if an attacker finds a way to infiltrate a system, it becomes vulnerable. This is why it is important to detect and classify cyberattacks in a network. Moreover, 5G networks are based on TCP/IP networks and share some vulnerabilities with them. Therefore, in order to detect cybersecurity anomalies in the 5G network, we implemented our work on the TCP/IP network.

The attack chosen for the analysis is Heartbleed, this attack targets the Heartbeat functionality of the SSL/TLS protocol in 2014 on the OpenSSL library. TLS is Transport Layer Security formerly Secure Sockets Layer, it is a protocol used in communication between computers. This is used in HTTP or VPN. It breaks down into two phases, the first is the Handshake which allows the encryption from an asymmetric key to a symmetric key. After this handshake, there will be the record time during which the machines will exchange data. If one of these machines does not receive data from a moment, a timeout will be passed and the connection will be closed.

For example, if a client wants to access its critical data over an HTTP connection, there will first be a handshake phase to enable encryption with certificate exchange, then the server will send its data during record time. If the two machines do not communicate, a timeout will be passed and the communication will be reset. To avoid this closure, SSL/TLS has created the Heartbeat feature to allow the connection to continue. It's an echo, request-response, and all these messages have the same format. A Heartbeat message is composed of type, length, payload, and random padding. It can be noticed that the content of our message should be related to the length of our message. In the vulnerable version of OpenSSL, if the length field is malicious, the server leaks data that may be confidential.

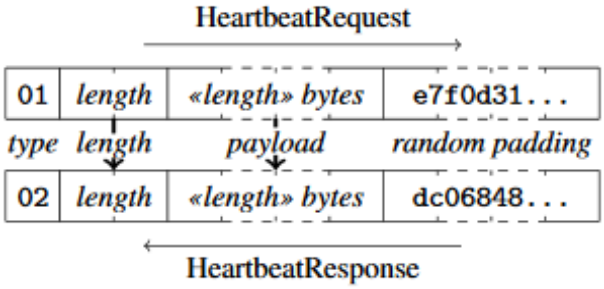


Figure 5-8: Heartbeat messages format.

With the aim of understanding more and to record data on the attack, generate and then track down malicious Heartbeat request is essential. That why it has been decided to simulate virtual network, with VirtualBox, it was composed by a target, an attacker and a detection router. Several tools have then been used to perform the attack and trace it. The first is Nmap for network mapping that is used to discover network and to see if a host is vulnerable to a particular attack. The second is Metasploit it is defined as a software to test vulnerabilities and provoke malicious code execution; it is updated by community to cover vulnerabilities of the CVE database. To analyse the attack traces the third used tools is Wireshark, this software can capture and record packets in a PCAP file and it gives an analysis of these packets (protocols used, packet length…).

After having performed and captured the attack, it is interesting to see now means to detect it. IDS for intrusion detection system, are software that use signature, anomaly or both (hybrid) detection. A signature is a correspondence between a packet and an attack already known. An anomaly detection is a statistical analysis of the flow that can reveal some weird packets ensemble and the attack. Intrusion detection systems (IDS) can be based on a network (network intrusion detection system (NIDS)) or on a host (host intrusion detection system (HIDS)), the difference between these two types is the flow it can analyse, HIDS only analyse the flow that is sent to the host or his environment. Whereas NIDS can analyse all the traffic that occur in a network.

An example of hybrid NIDS is Zeek formerly named Bro, which is a software based on attack detection scripts. Heartbleed script shows two means to control traffic, when Zeek faced a plain SSL heartbeat request, signatures are used to control traffic, but if the SSL heartbeat request is encrypted, signatures are not usable. To warn user, Zeek provide anomalies detection based on filters with arbitrarily chosen static constants. This second detection method show some result, but it is not totally accurate because it is easy for an attacker to bypass simple static filters.

6 Demonstration activities - Federated eXplainable AI (FED-XAI) demo

The demonstration activity, carried out jointly with WP5, focuses on the development of a framework for the FL of XAI models. The scenario has been introduced in [D5.1] and relates to a Vehicle-to-Everything (V2X) use case: several instances of vehicular UEs are connected to a B5G/6G BS and receive a video stream from the cloud or the edge. From an algorithmic perspective, the goal (aimed at contributing to the *Connecting Intelligence* objective) is to show how the UEs can collaboratively train an XAI model for prediction, e.g., for QoE forecasting, without any disclosure of raw private data. The implementation of an edge-side real-time dashboard is intended to leverage and value explainability, thus contributing to the objective of *Trustworthiness*.

The rest of this section describes the design of a UE application with such a Fed-XAI capability and the requirements of the modules involved in the FL process.

6.1 UE application with Fed-XAI capability

The main modules of the UE application with Fed-XAI capability are schematised in Figure 6-1. Notably, a centralised communication topology is considered: a server orchestrates the learning process by aggregating models/updates provided by the involved parties.

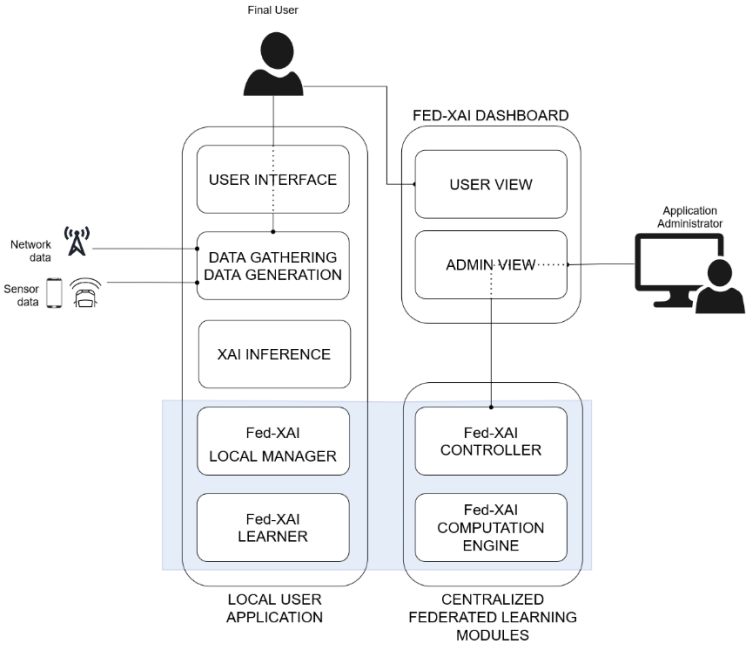


Figure 6-1: Overview of AI stacks at user and network side considered for the FED-XAI demo.

Two kinds of users are envisaged: the final user, who owns and operates on a UE enjoying the application with Fed-XAI capability, and the application administrator, who configures and monitors the FL process. Both benefit from the *explainability* aspect, e.g., by obtaining explanations from a dedicated view of a dashboard.

The **Local User Application** macro-module is composed of four main modules that allow him/her to i) interact with the application, ii) gather and generate data, iii) perform prediction using an XAI model, and iv) participate in the FL process.

The **Fed-XAI** modules (highlighted in blue in the figure) involved in the application that exploits the FL process are the following:

- *Fed-XAI controller*
  - configures and initializes the FL process.
- *Fed-XAI computation engine*
  - server-side "model aggregator" component of the FL process.
- *Fed-XAI Local manager*
  - client-side control entity of the FL process.
  - each associated with a single application running on a UE
- *Fed-XAI learner*
  - performs local XAI model training

Fed-XAI computation engine and Fed-XAI controller modules form the **Centralized Fed-XAI** macro-module.

Actually, also a hierarchical FL mode may be envisaged. In this mode, we suppose that different *Fed-XAI computation engines*, that cannot communicate among themselves, may decide to collaborate to generate an aggregated model from the aggregated models generated by each of them. In this mode, an upper-level *Fed-XAI computation engine* is devoted to generating a new model by aggregating the models generated by the lower-level servers and to sending it back to them.

Finally, in the figure we also show the **FED-XAI Dashboard** module that allows users to check prediction explanations and the application administrator to also setup and control the FL environment.

6.2 Functional Requirements

In the following, the functional requirements of the **Fed-XAI** modules are described. They are defined based on the two approaches for FL of explainable models mentioned in Section 5.3.

Module: **Fed-XAI local manager:**

- **interface with** Fed-XAI controller:
  - can signal/deny availability to participate to the FL process to the Fed-XAI controller;
  - can ask the availability of (and possibly receive) XAI models to the Fed-XAI controller;
- can send the performance of the current XAI model on its local data to the Fed-XAI computation engine;

Module: **Fed-XAI Learner:**

- can train an XAI model based on its local data;
- can update an XAI model based on its local data;
- can evaluate some aggregated statistics of its local data that are useful for building an XAI model;
- **interface with** Fed-XAI computation engine:
  - can send model update along with proper local statistics (aggregated, e.g., number of objects in the local dataset) to the Fed-XAI computation engine;
  - can send aggregated statistics of its local data to the Fed-XAI computation engine.

can receive *cumulative* aggregated statistics from the Fed-XAI computation engine; Module: **Fed-XAI computation engine**

- can initialise an XAI model for the target task in the target application;
- can select a fraction of the available clients for participating in the FL process, either randomly or based on some criteria. This information is provided by the Fed-XAI controller;
- can send/receive an XAI model to/from the Fed-XAI Learner;
- can aggregate received XAI models based on the strategy;
- can receive partial aggregated statistics from a Fed-XAI Learner;
- can evaluate cumulative aggregated statistics;
- can send cumulative aggregated statistics to the Fed-XAI Learner;
- can build a model based on cumulative aggregated statistics received from the Fed-XAI Learner;
- can receive the performance of an XAI model from the Fed-XAI Local Manager;
- can calculate the performance of the federated model on a dedicated dataset or based on the performance on each client.
- can send to the Fed-XAI controller and to the Fed-XAI dashboard some statistics about the FL process;
- **interface with** Fed-XAI controller:
    - can send the global model to the Fed-XAI controller;

Module: **Fed-XAI controller**

- can maintain a collection of available models, each associated with a log/history of the measure of performance;
- can receive a signal of availability to participate in the FL process by the Fed-XAI local manager module;
- can start an FL process, based on client availability;
- can configure the model parameters for the FL process, based on admin input;
- can receive from the Fed-XAI computation engine some statistics about the FL process;
- can configure an FL strategy for model aggregation. The strategy can also implement some sort of clients clustering / personalisation/ selection. This information is shared with the Fed-XAI computation engine module;

Module: **Fed-XAI dashboard**

- can receive the information for explanation from the XAI inference module;
- can characterise the explainability of an XAI model;
- can show an explanation about any decision made by the UE application (XAI model) concerning her / his use of the application;
- can allow the administrator to configure, execute and monitor the FL process, interacting with the Fed-XAI controller;

The requirements defined above are those of a generic real-world application with Fed-XAI capabilities.
For the purpose of the demonstration activity, the design of the prototype may entail the implementation of a subset of the defined requirements.

A feasibility study will lead to the selection of the most suitable FL framework, which will have to be properly adapted to support the above-mentioned requirements for the novel Fed-XAI capability. This will be followed by the development of the application, the integration into a real-time distributed testbed and the experimental validation. Further details on the implementation of the FED-XAI demo from an algorithmic perspective will be provided in the final Hexa-X WP4 deliverable (D4.3).

7 Conclusions

This chapter provides a summary of the work conducted in T4.2 and T4.3 in terms of project targets and relevant KPIs/KVIs.

The Connecting Intelligence research challenge that is tackled by project Hexa-X is translated into a number of quantifiable targets, which are the following:

- Increased AI algorithm robustness to system parameter volatility, lower complexity and significant Bit Error Rate (BER)/ BLock-Error Rate (BLER) gain, as compared to classical approaches (Target **T1**, in relation to task T4.2);
- Increased AI algorithm robustness to system parameter volatility, lower complexity and efficient resource utilisation and rate gain as compared to classical approaches (Target **T2**, in relation to task T4.2);
- Resilient communication and compute network services for distributed AI applications in large scales (e.g., applications with >1000 collaborating AI components (Target **T3**, in relation to task T4.3);
- The accuracy of an XAI model within (<10%) of "black box" solutions (e.g., Deep Neural Networks - DNNs) - (Target **T4**, in relation to task T4.3);
- Energy reduction of a factor of (>10) at the infrastructure level and a factor of (>100) at the user devices' side, as a result of (network & application) workload offloading and learning/inferencing task delegation (Target **T5**, in relation to task T4.3);
- Increased trustworthiness of AI through privacy and security enhancing technologies; using differential privacy to evaluate privacy versus communication utility trade-off (Target **T6**, in relation to task T4.3).

In what appears in the Table 7-1 below, the elaborated technical enablers are listed, recapping on the problems that are tackled, the proposed solution framework and the targeted 6G KPIs/ KVIs, along with the quantifiable targets of relevance to each technical enabler. It should be noted that the degree of accomplishing these well-defined targets for each technical enabler will be discussed in deliverable D4.3.

Table 7-1: Technical enablers and related 6G KPIs/ KVIs to be addressed.

Technical enabler (title)	**Problem/ challenge to be addressed**	**(Initial) proposed solution framework**	**Targeted 6G KPIs/ KVIs**	**Quantifiable Targets**
LiDAR aided human blockage prediction for 6G (Section 2.1.1.1)	Prediction of dynamic blockage to communication links caused by human movements in indoor scenarios.	Use infrastructure-mounted LiDARs to monitor the indoor environment and detect, track and predict movements to determine future blockages based on LSTM	Mobility support Flexibility AI agent availability	T2, T3

		networks and ray casting.		
Graph neural network-based access point selection in cell-free massive MIMO systems (Section 2.1.1.2)	Access point selection in initial access and mobility management in cell-free massive MIMO networks to improve the latency.	A GNN-based solution to predict the candidate APs using a limited number of signal strength measurements.	Mobility support	T2
AI based compressed sensing for beam selection in D-MIMO (Section 2.1.1.3)	Reducing beam selection overhead in scenarios with high beam density, like high frequencies in the mmWave range and distributed MIMO deployments	Application of compressed sensing with learned sensing matrix and neural sparse decoder	Mobility support	T1
Constellation shaping (Section 2.1.2.1)	How to facilitate pilotless transmissions, thereby reducing communication overhead and improving spectral efficiency	Learn a constellation shape and receiver jointly. A learned constellation can be used for blind detection by the simultaneously learned receiver.	Bit rate, spectral efficiency	T1, T2
NN/ML aided channel (de)coding for constrained devices (Section 2.1.2.2)	How to improve the efficiency of FEC mechanisms for short packets in IoT and URLLC use-cases to reduce the number of transmission errors and thus energy consumptions and/or latency.	Design and optimise linear block codes and decoders jointly in an auto-encoder model based on Belief Propagation structures.	BER/BLER gain & bit rate/spectral efficiency improvement. Complexity gain Network & UE energy efficiency Interpretability level and compatibility with legacy systems	T1, T2, T4
Deep learning for location based beamforming (Section 2.1.2.3)	Beam search can be very time consuming when using a large number of antennas at the base station. Knowing where users are could reduce greatly the	Train a NN to directly learn the location/precoder mapping. The NN is structured in a particular way (using random Fourier	Signalling overhead Latency Spectral eficiency Generalisability	T2

	computational burden. However, existing location based beamforming methods assume the existence of a LoS path. In the nLoS case, the proposed methods do not perform well.	features) in order to be able to learn functions of high spatial frequency.		
Machine learning aided beam management (Section 2.1.2.4)	Reducing beam selection latency of APs which have recently changed to active state from dormant state in dense networks.	A DCB-based approach to perform instantaneous beam selection for a recently restored AP using the information from its neighbouring APs, enabling a newly activated AP to instantaneously start serving the users.	Complexity gain, latency	T2
TX-side CNN for reducing PA-induced out-of-band emissions (Section 2.1.3.1)	How to learn a waveform that produces less out-of-band emissions under a nonlinear PA and is accurately detectable at the receiver.	Introduce a light-weight CNN to the transmitter, which is trained based on the measured out-of-band emissions and link capacity.	Bit rate, energy efficiency	T1, T2
ML/AI empowered receiver for PA non-linearity compensation (section 2.1.3.2)	How to enable PA operation in more energy efficient regime by compensating in-band distortions due to PA nonlinearities at the receiver?	A NN-based demapper computes soft bits to the decoder by considering the impact of distortions due to PA nonlinearity using trained model at receiver	Bit rate, spectral efficiency, energy efficiency	T1, T5
AI/ML-based predictive orchestration (section 2.2.1)	To be able to perform orchestration actions in a proactive way based on preditions.	The technical enabler itself is the solution to the problem (i.e,	AI agent availability, AI agent reliability, Network efficiency, UE energy efficiency	T3, T5

		the orchestration function will rely on AI/ML models to provide proactive behaviours).		
Distributed AI for automated UPF scaling in low-latency network slice (Section 2.2.2)	Improvement of the reaction time of automated orchestration on network slices requiring low E2E latency	AI techniques to trigger the preemptive auto-scaling of local UPF placed at the network edge	Inferencing accuracy, latency, and network energy efficiency	T3, T5 (on a single component)
AIaaS - seamless exploitation of network knowledge (section 3.1.1)	How to enable a UE carrying an ML model keep it up-to-date in mobility/ connection interruption regimes.	AI service & API to route an inferencing task to the most relevant and available AI agent subject to latency and energy limitations.	(On device) AI agent availability, AI agent reliability, flexibility, mobility support	T3, T5
Network impairment resilience of autonomous agents (section 3.1.2)	Use of ML to predict mobility/ connection interruption regimes and provide resilience for the UE / AI agent.	Deploy data analytics methods to predict quality issues ahead of time to prepare the UE/AI agent for connectionless operation.	AI agent availability, reliability	T3
Distributed low-complexity model learning (section 3.1.3)	Assist constrained devices in training models using in-network computation resources	Distributed learning schemes and control traffic exchanges to enable such feature at low-complexity/energy cost	Energy reduction, inferencing accuracy	T1, T5
Federated ML model load balancing at the edge (section 3.2.1)	Provide a low latency and high quality distributed ML service of heterogeneous sensors, devices connected to	A dynamic reconnection solution to provide load balancing to remedy potential hot spots and	AI agent availability, Latency Energy reduction	T3, T5

	distributed AI nodes by straggler mitigation.	data type diversity to ensure quality balance for the federated learners.		
Scalable and resilient deployment of distributed AI (Section 3.2.2)	Supporting AI native communication patterns and traffic handling mechanisms on the network edge	Optimised wireless channels and traffic differentiation for neuromorphic and other sparse AI systems to enable operation with high energy-efficiency, utilize tolerance for connection impairments.	AI agent density, Inferencing accuracy, and Latency	T3
Multi-agent ML for multi-cell multi-user MIMO (section 3.2.3)	Optimal beamforming in multi-cell multi-antenna systems requires complex inter-cell interference coordination. Practical limitations exists for sharing the information needed for this coordination.	A Multi-agent Deep RL (MA-DRL) framework. Decentralised actors with partial observability can learn a multi-dimensional continuous policy in a centralised manner with the aid of shared critic with global information.	Inference accuracy Inference	T3
Flexible compute workload assignment (CaaS) (section 3.3.1)	How to delegate/ distribute (generic) processing tasks across the network.	The exact configuration of a target compute platform (CPU, GPU, NPU etc.) is abstracted from the workload offloading API.	AI agent availability, network energy efficiency, flexibility	T5
AI workload placement for energy, knowledge sharing and trust optimisation	Dealing with the trust, traffic, and energy consumption problems, that physical nodes who undertake the	Algorithms solving the optimisation problem, producing the	AI agent availability, network energy efficiency	T3, T5

(Section 3.3.2)	execution of AI mechanisms, face.	allocation of AI algorithms/mechanisms to physical nodes following a heuristic technique.		
Joint allocation of radio and computing resources for edge inference (Section 3.3.3)	Enable inference at the edge on data collected by end users, with the least energy consumption, under reliability and delay constraints	Adaptive optimisation method to jointly select, online, data compression scheme, transmission power, and computing scheduling at a Mobile Edge Host, with performance controlled through suitably defined state variables	Energy efficiency, inference accuracy, Latency	T5
Low complexity radio resource allocation in cell-free massive MIMO (Section 4.1.1)	Reducing the computational complexity in radio resource allocation tasks in cell-free massive MIMO networks.	Unsupervised learning-based DNN to learn the optimal resource allocations to achieve the desired objective (sum rate maximization).	Complexity gain, flexibility	T2
Supervised learning based sparse channel estimation for RIS aided communications (section 4.1.2)	To estimate RIS assisted channel with improved accuracy.	ML based parameter estimation in sparse angular channel model. The sparse representation and model performance improve the channel estimation accuracy.	Channel Estimation Error	T1
Data significance-aware RRM (section 4.2.1)	Dealing with over-the-air learning scenarios where available radio	Pre-transmission or post-reception evaluation of	Inferencing accuracy, latency, E2E energy efficiency	T3, T5

	and storage resources are limited at device and edge RAN.	learning data set importance to prioritise future device transmissions.		
Frugal AI: Channel Estimation (section 4.2.2)	To estimate channel with low computational & sample complexity and close to optimal performance	Implement MMSE in terms of learnable weights of an NN, to avoid the computational complexity required by the MMSE	Channel Estimation Error, Spectral Efficiency	T5, T1
Deep unfolding for efficient channel estimation (section 4.2.3)	Channel estimation algorithms based on physical models are theoretically very accurate but also very sensitive to hardware impairments and modifications.	View the channel estimation algorithm as a NN (deep unfolding technique) that can be optimised and thus adapt in real time to incoming data.	Spectral efficiency Flexibility Convergence Generalisability Frugal AI Resistance to adversarial attack	T2, T3
Efficient channel charting (section 4.2.4)	Channel charting aims at localizing users relatively to one another in an unsupervised manner (without requiring access to GNSS , using only channels). It can be used for several applications, ranging from resource or pilot allocations to beam prediction. Most existing channel charting methods rely on the second order moment of channels and are thus computationally expensive.	Use a specifically designed distance measure that do not require to compute second-order moments and is relevant to the channel charting task. The measure can then be used to obtain the chart, either using classical dimensionality reduction methods (ISOMAP) or within a neural network architecture.	Positioning accuracy Complexity gain Data privacy protection Frugal AI	T3
Neural Network based Semantic Communications	Transmitting only relevant information sufficient for the	Semantic communications with neural	Energy efficiency, reliability, inference accuracy	T3

(Section 4.3)	receiver to capture the meaning, thus saving significant communication bandwidth and improving reliability of the transmission-reception chain	network based semantic extraction and interpretation		
Privacy preserving clustering: Federated fuzzy c-means (5.1.1)	Allowing clients to perform collaborative clustering on their private data	Design an ad-hoc strategy for fuzzy c-means clustering on horizontally partitioned data	AI privacy	T6
Differentially Private FL (section 5.1.2)	Using UE location for RSRP prediction Dealing with privacy attacks targeting to FL model	Introduce a FL model to jointly generate an RSRP prediction model with DP guarantee to protect FL	AI privacy	T6
Security Mechanism friendly privacy solution for FL (section 5.1.3)	Dealing with privacy attacks and malicious behaviour of the clients in FL	Introduce a multi-hop communication along with blind signature to hide the identity of the clients and identify malicious behaviour of clients.	AI privacy	T6
XAI models for QoE prediction (5.2.1)	Investigating trade-off between accuracy and interpretability of tree-based model in QoE prediction task	Adoption of multiway Fuzzy Decision Trees, Binary Decision Trees and Random Forests	Inferencing accuracy, explainability	T4
XAI for radio network control (section 5.2.2)	Explain and distinguish the effect of network configuration and user device load to network performance KPIs.	A hierarchical regression model for KPI prediction with applicability to automated network control.	Explainability, Flexibility, Data quality, Complexity gain,Generalisability, Deployment flexibility	T3, T4, T5
Fed-XAI: federated learning	Enabling collaborative training of explainable	Revisiting the aggregation	Inferencing accuracy, explainability	T4

of explainable AI models (section 5.3)	AI models without violating privacy of data owners.	strategy to be carried out by a central server for dealing with inherently interpretable models.		
Detection and Classification of cybersecurity anomalies in 5G network (section 5.4)	Find up to date attacks and data relative to this attack. Detect and classify these attacks.	1) Find 5G vulnerabilities, understand it. 2) execute the attack on 5G emulated network 3) find detection methods with ML (data analysis and NN)	Data privacy protection, Resistance to adversarial attacks	T6

Table 7-2 gives a summary of KPI's considered in T4.2 along with the corresponding target details as described below.

Table 7-2: KPIs considered for AI-driven air interface design.

Target T1: Increased AI algorithm robustness to system parameter volatility, lower complexity and significant Bit Error Rate (BER)/ BLock-Error Rate (BLER) gain, as compared to classical approaches			
KPI	**Tentative Numbers**	**Contributing solution areas & Technical Solutions**	**Baseline & Verification**
BER / BLER	BER/BLER gain is evaluated against the complexity of the decoding algorithm, evaluated as the number of operations of each type (real/LLR domain: multiplications, additions, etc. and binary operations: XOR, AND, OR, etc.). BER/BLER is evaluated at different Eb/N0 values, typically 0 to 7 dB. Expected gain depends on the code rate and code size, larger gains being expected for smaller codes at lower code rates (up to 1dB in the waterfall region when compared to baseline).	Auto-encoder models, RNN structured decoder. Iterative decoder modelled as a recurrent neural network. Trainable weights for Parity Check matrix, input LLRs, decoder iterations outputs.	Low complexity decoder, applicable to constrained devices, i.e., BP decoding on known codes, e.g. BCH using cycle-reduced parity check matrices and Low-Density Parity-Check (LDPC) codes. Simulation of a transmission chain including the coder, modulator, channel, demodulator and decoder.

Channel estimation error	Values depend on various factors such as transmit power, device positions and scenario, therefore cannot be generalized.	Supervised learning AI driven algorithms can be used to improve this, while existing physical models can be used for efficient designs.	LSE, MMSE estimator results Implementation of algorithms and numerical analysis with random channels
Target T2: Increased AI algorithm robustness to system parameter volatility, lower complexity and efficient resource utilisation and rate gain as compared to classical approaches			
KPI	**Tentative Numbers**	**Contributing solution areas & Technical Solutions**	**Baseline & Verification**
Complexity gain (number of operations)	Processing time/number of operations in the proposed algorithms in each problem scenario where other problem-specific metrics (spectral efficiency, BER etc.) are achieved	Supervised learning, unsupervised learning DNNs to perform radio resource management tasks in cell-free massive MIMO networks. Ex: uplink power control, fronthaul signaling capacity allocation for uplink data and CSI.	The processing complexity of optimisation-based/ conventional algorithms for the considered problem scenarios. Analytical complexity analysis, numerical validation: generating simulation data and comparing processing time of the proposed ML-based solutions against the baseline.
Bit rate & spectral efficiency	Spectral efficiency (SE) gains should be demonstrated against current systems. The analysis should take into account also the signaling overhead, e.g., from DMRS, Cyclic Prefix (CP), etc. Exact numbers depend on the use-case and simulation scenario, but SE gains in the order of 10-20% over 5G should be demonstrated due to overhead reduction.	End-to-end (E2E) optimisation (constellation shaping, CNN-based mitigation of PA nonlinearities)	Conventional TX & RX solutions, e.g., QAM with CP--OFDM and LMMSE detection Numerical simulations with separated training and validation scenarios
Flexibility	Focus on online ML model operation; target is for time for ML model to adapt to radio environment changes to be shorter than channel coherence time. Value of ML model time "reaction time" depends on considered scenario.		

Mobility support	Relevant numerical metrics are Latency (time of outage) , handover failure rate,link failure rate Exact values depend on the scenarios, procedures and signalling defined in 3GPP specification.	Supervised learning AI methodologies like graph neural networks can be used to predict the potential access points Sensing can be used to track dynamics and recurrent neural networks can be used to predict future mobility	Traditional measurement based handover or AP selection, where all the whitelisted cells of the measurement object are measured . Using system level simulations.

Table 7-3 gives T4.3 KPI's and relevant target breakdown.

Table 7-3: KPIs considered for sustainable and trustworthy in-network learning.

Target T3: Resilient communication and compute network services for distributed AI applications in large scales		
KPI	**Targets and improvements**	**Contributing technical solutions**
AI agent availability	Ensure high availability of AI agents in a distributed environment under mobility.	Prediction of mobility, optimised workload movement
AI agent reliability	Ensure high reliability of AI response in the presence of connectivity impairments.	Prediction of impairments in connectivity or missing data
AI agent density	AI agent density is improved by decreasing the communication overhead for AI traffic	Efficient semantic coding, AI-native communication, multiple access techniques.
Inferencing accuracy	Enable flexible accuracy trade-off with latency	Distributed AI for automated UPF scaling in low-latency network slice,
Latency	Inferencing latency of AI workloads is improved by decreasing the impact of mobility, the application of semantic coding and flexible trade-off with accuracy. Training latency: controlled latency for federated learning over heterogenous agents and connections."	FL struggler mitigation, optimised reallocation of AI workload, Distributed AI for automated UPF scaling in low-latency network slice
Target T4: The accuracy of an XAI model within (<10%) of "black box" solutions		
KPI	**Targets and improvements**	**Contributing technical solutions**

Interpretability level	High level of interpretability through the adoption of inherently interpretable AI models or through the adoption of post-hoc XAI techniques.	XAI models for QoE prediction (Section 5.2.1) Fed-XAI: federated learning of explainable AI models (Section 5.3)
Inferencing accuracy	Accuracy of inherently interpretable models (Decision Trees and Rule-Based Systems) within 10% of "black box" solutions	XAI models for QoE prediction (Section 5.2.1) Fed-XAI: federated learning of explainable AI models (Section 5.3)
Target T5: Energy reduction of a factor of (>10) at the infrastructure level and a factor of (>100) at the user devices' side, as a result of (network & application) workload offloading and learning/ inferencing task delegations		
KPI	**Targets and improvements**	**Contributing technical solutions**
Network energy efficiency	Network energy consumption due to computing of offloaded tasks from end users depends on the guaranteed network availability. Full availability is the highest energy consumption baseline (the most beneficial for the users), to be possibly reduced by guaranteeing less availability, thus pushing part of the computation, e.g., back to the user or the cloud. This can help exploring trade-offs between users' and network energy consumption.	Distributed AI for automated UPF scaling in low-latency network slice (on a single component and it needs to be evaluated)
UE energy efficiency	UEs transmit raw data or extracted features to the network instead of running a complex model locally. Offloading learning and/or inference tasks to the network decreases energy consumption of UEs whenever the workload is highly demanding with respect to the amount of data to be transmitted to enable remote computation. This depends on, e.g., model complexity, wireless channels conditions, availability of computing resources at the network side.	Joint allocation of radio and computing resources for edge inference. UEs energy consumption is the objective, to be minimised under latency and inference reliability constraints. (Section 3.3.3) Quantifiable gains to be evaluated case by case (simulation-based performance evaluation with specific exploited models)
Inferencing accuracy	Transmission energy can be reduced by considering data compression before transmission, with possibly a cost in terms of accuracy. This energy reduction must be evaluated by considering a degradation of at most a predefined factor in terms of inference	Joint allocation of radio and computing resources for edge inference. (Section 3.3.3) Inference reliability (which translates into accuracy) is a constraint, to be set a priori depending on the specific application.

	reliability/accuracy, to be defined based on the specific application.	
Target T6: Increased trustworthiness of AI through privacy and security enhancing technologies and AI network intrusion detection capability		
KPI	**Targets and improvements**	**Contributing technical solutions**
AI privacy	Provide a higher level of privacy for user data by tuning the value of privacy budget (ε) using differential privacy. Improving the privacy and security of the FL (Ensuring Lower attack success rate in FL setting) through the usage of proposed multi-hop privacy solution. Also, guaranteeing the same model performance in term of accuracy before and after applying the proposed method.	Privacy preserving clustering: Federated fuzzy c-means (Section 5.1.1) Differentially Private FL (Section 5.1.2) Security Mechanism friendly privacy solution for FL (Section 5.1.3)

8 References

[3GPP] https://www.3gpp.org/

[A19] A. Alkhateeb. “DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications.” In Proc. of Information Theory and Applications Workshop (ITA), pp. 1–8, San Diego, CA, Feb 2019.

[AH18] F. A. Aoudia and J. Hoydis, "End-to-End Learning of Communications Systems Without a Channel Model," in Proc. 52nd Asilomar Conference on Signals, Systems, and Computers, pp. 298-303, Oct. 2018.

[AH21] F. A. Aoudia and J. Hoydis, “Trimming the fat from OFDM: Pilot- and CP-less communication with end-to-end learning,” in Proc. IEEE International Conference on Communications Workshops (ICC Workshops), Jun. 2021.

[AI4AD] https://www.itu.int/en/ITU-T/focusgroups/ai4ad/Pages/default.aspx

[AIML21] 5G PPP Technology Board. AI and ML – Enablers for Beyond 5G Networks. https://5g-ppp.eu/wp-content/uploads/2021/05/AI-MLforNetworks-v1-0.pdf

[AIREG] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

[AME18] A. Abdelreheem, E. M. Mohamed, and H. Esmaiel. “Location-based millimeter wave multi-level beamforming using compressive sensing.” IEEE Communications Letters, vol. 22, no. 1, pp.185–188, 2018.

[ASR+20] S. Ali W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wi-etfeld, K. Mei, H. Shiri, H.-J. Zepernick, T. M. C. Chu, I. Ahmad, J. Hu-usko, J. Suutala, S. Bhadauria, V. Bhatia, R. Mitra, S. Amuru, R. Abbas, B. Shao, M. Capobianco, G. Yu, M. Claes, “6G white paper on machine learning in wireless communication networks”, arxiv, 2020.

[AsymSHAP] C. Frye, R. Colin, and I. Feige. "Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability." *Advances in Neural Information Processing Systems* 33, 2020.

[AZB+19] C. D’Andrea, A. Zappone, S. Buzzi, and M. Debbah, “Uplink power control in cell-free massive MIMO via deep learning,” in 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 554–558, 2019.

[BBD+11] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, “Towards a theory of semantic communication,” in IEEE Network Science Workshop, pp. 110–117, 2011.

[BDM+20] A. Blanco-Justicia, J. Domingo-Ferrer, S. Mart'inez, D. Sánchez, A. Flanagan, and K. E. Tan. “Achieving Security and Privacy in Federated Learning Systems: Survey, Research Challenges and Future Directions.” Eng. Appl. Artif. Intell. 106 (2021): 104468.

[BLD+19] S. Bi, J. Lyu, Z. Ding, and R. Zhang, “Engineering radio maps for wireless resource management,” IEEE Wireless Communications, vol. 26, no. 2, pp. 133–141, 2019.

[BM14] E. Boutillon and G. Masera, “Hardware Design and Realization for Iteratively Decodable Codes”, in Channel Coding: Theory, Algorithms, and Applications by D. Declerq M. Fossorier and E. Biglieri, Academic Press, 2014.

[BMR+21] J. L. Corcuera Bárcena, F. Marcelloni, A. Renda, A. Bechini, and P. Ducange, “A Federated Fuzzy c-means Clustering Algorithm,” In WILF’21: The 13th International Workshop on Fuzzy Logic and Applications, (accepted for publication), 2021.

[BS20] E. Björnson and L. Sanguinetti, "Scalable Cell-Free Massive MIMO Systems," in IEEE Transactions on Communications, vol. 68, no. 7, pp. 4247-4261, July 2020.

[CB21] E. Calvanese Strinati and S. Barbarossa, “6G networks: Beyond Shannon towards Semantic and Goal-Oriented Communications,” Computer Networks, vol. 190, p. 107930, 2021

[CCB+20] T. V. Chien, T. N. Canh, E. Björnson, and E. G. Larsson, “Power control in cellular massive MIMO with varying user activity: A deep learning solution,” IEEE Transactions on Wireless Communications, vol. 19, no. 9, pp. 5732–5748, 2020.

[CGH+21] M. Chen, D. Gunduz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed Learning in Wireless Networks: Recent Progress and Future Challenges," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 12, pp. 3579-3605, Dec. 2021.

[CGY+22] Choongil Yeh, Gweon Do Jo, Young-Jo Ko, & Hyun Kyu Chung (2022). Perspectives on 6G wireless communications. ICT Express.

[CMW+21a] Y. Chen, J. Mohammadi, S. Wesemann, and T. Wild, “Turbo-AI, Part I: Iterative Machine Learning Based Channel Estimation for 2D Massive Arrays,” in Proc. IEEE 93rd Veh. Technol. Conf. (VTC’21 Spring), Apr. 2021.

[CMW+21b] Y. Chen, J. Mohammadi, S. Wesemann, and T. Wild, “Turbo-AI, Part II: Multi-Dimensional Iterative ML-Based Channel Estimation for B5G,” in Proc. IEEE 93rd Veh. Technol. Conf. (VTC’21 Spring), Apr. 2021.

[COINRG] https://irtf.org/coinrg

[CZK21] I. Čilić, I. P. Žarko and M. Kušek, "Towards Service Orchestration for the Cloud-to-Thing Continuum," 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech), 2021, pp. 01-07, doi: 10.23919/SpliTech52315.2021.9566410.

[D4.1] Hexa-X Deliverable D4.1, "AI-driven communication & computation co-design: Gap analysis and blueprint". Online: https://hexa-x.eu/wpcontent/uploads/2021/09/Hexa-X-D4.1_v1.0.pdf

[D5.1] Hexa-X Deliverable D5.1, " Initial 6G Architectural Components and Enablers". Online: https://hexa-x.eu/wp-content/uploads/2022/03/Hexa-X_D5.1_full_version_v1.1.pdf

[D6.2] Hexa-X Deliverable D6.2, " Design of service management and orchestration functionalities". Online: https://hexa-x.eu/wp-content/uploads/2022/05/Hexa-X_D6.2_V1.1.pdf

[DC17] M. Dikmen and C. Burns, "Trust in autonomous vehicles: The case of Tesla Autopilot and Summon," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2017, pp. 1093-1098, doi: 10.1109/SMC.2017.8122757.

[DK20] H. T. Dao and S. Kim, "Effective Channel Gain-Based Access Point Selection in Cell-Free Massive MIMO Systems," in IEEE Access, vol. 8, pp. 108127-108132, 2020.

[DMN+06] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, “Calibrating noise to sensitivity in private data analysis,” In IACR Theory of Cryptography Conference (TCC), New York, New York, volume 3876 of Lecture Notes in Computer Science, pages 265–284. Springer-Verlag, 2006. doi: 10.1007/11681878 14.

[DMR+22] D. Dampahalage, K. B. Shashika Manosha, N. Rajatheva and M. Latva-Aho, "Supervised Learning Based Sparse Channel Estimation For RIS Aided Communications," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 8827-8831, doi: 10.1109/ICASSP43922.2022.9746793.

[DR14] C. Dwork and A. Roth, “The Algorithmic Foundations of Differential Privacy,” Found. Trends Theor. Comput. Sci., vol. 9, no. 3–4, pp. 211–407, Aug. 2014.

[DS21] Dama, F., & Sinoquet, C. Time Series Analysis and Modeling to Forecast: a Survey. 2021. https://doi.org/10.48550/arXiv.2104.00164.

[ETS17] ETSI EN 303 146-4 V1.1.2, “Radio Virtual Machine (RVM)” developed by the European Telecommunications Standards Institute, 2017.

[ETSI] https://www.etsi.org/

[F220] "F2: ML at the Extreme Edge: Machine Learning as the Killer IoT App," 2020 IEEE International Solid- State Circuits Conference - (ISSCC), 2020, pp. 525-527, doi: 10.1109/ISSCC19947.2020.9063056.

[FDO+21] P. Ferrand, A. Decurninge, L. G. Ordoñez, and M. Guillaud, “Triplet-Based Wireless Channel Charting: Architecture and Experiments,” IEEE Journal on Selected Areas in Communications, 2021.

[FGAN] https://www.itu.int/en/ITU-T/focusgroups/an/Pages/default.aspx

[FS20] H. Farhadi and M. Sundberg, “Machine learning empowered context-aware receiver for high-band transmission,” IEEE Globecom Workshops, 2020.[FZF+17] J. Fang, R. Zhang, T. Z. Fu, Z. Zhang, A. Zhou, and J. Zhu,“Parallel stream processing against workload skewness and variance,” In Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing pp. 15-26, June 2017.

[G14] B. Gedik, "Partitioning functions for stateful data parallelism in stream processing." *The VLDB Journal* vol. 23, no. 4 pp. 517-539, 2014.

[GAL21] A. Galanopoulos, J. A. Ayala-Romero, D. J. Leith and G. Iosifidis, "AutoML for Video Analytics with Edge Computing," IEEE INFOCOM 2021 - IEEE

Conference on Computer Communications, 2021, pp. 1-10, doi: 10.1109/INFOCOM42981.2021.9488704.

[GBT] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, *30*, 3146-3154, 2017.

[GCH+17] T. Gruber S. Cammerer, J. Hoydis, and S. t. Brink, "On deep learning based channel decoding" Proceedings of the 51st Annual Conference on Information Sciences and Systems, pp. 1–6, 2017.

[GL10] K. Gregor and Y. LeCun. "Learning fast approximations of sparse coding." In Proceedings of the 27th International Conference on International Conference on Machine Learning, pages 399–406. Omnipress, 2010.

[GLL+20] E. Gonültas¸, E. Lei, J. Langerman, H. Huang, and C. Studer, "Csi-based multi-antenna and multi-point indoor positioning using probability fusion," 2020.

[HMG+12] Hanson, David & Mazzei, Daniele & Garver, Carolyn & de rossi, Danilo & Stevenson, M. "Realistic Humanlike Robots for Treatment of ASD, Social Training, and Research; Shown to Appeal to Youths with ASD, Cause Physiological Arousal, and Increase Human-to-Human Social Engagement". Realistic Humanlike Robots for Treatment of Autism, PETRA 2012.

[HPT+15] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning Both Weights and Connections for Efficient Neural Networks," in Proc. of the 28th Int. Conference on Neural Information Processing Systems - Volume 1, ser. NIPS'15. Cambridge, MA, USA: MIT Press, pp. 1135–1143, 2015.

[HYL17] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 1025–1035.

[I20] Inivation, "Understanding the Performance of Neuromorphic Event-based Vision Sensors", white paper, https://inivation.com/wp-content/uploads/2020/05/White-Paper-May-2020.pdf

[IRTF] https://irtf.org/

[ITU] https://www.itu.int/en/Pages/default.aspx

[JAB+18] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: Scalable Adaptation of Video Analytics," in Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, ser. SIGCOMM '18. New York, NY, USA: Association for Computing Machinery, pp. 253–266, 2018. [Online]. Available: https://doi.org/10.1145/3230543.3230574

[JGH18] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, page 8580–8589, Red Hook, NY, USA, 2018. Curran Associates Inc.

[JLD] E. Jorswieck, E. Larsson, and D. Danev, "Complete Characterization of the Pareto Boundary for the MISO Interference Channel," IEEE Transactions on Signal Processing, vol. 56, pp. 5292–5296, October 2008.

[JRB14] S. Jaeckel, L. Raschkowski, K. Borner, and L. Thiele, “Quadriga: A 3-d ¨ multi-cell channel model with time evolution for enabling virtual field trials,” IEEE Transactions on Antennas and Propagation, vol. 62, no. 6, pp. 3242–3256, 2014.

[K08] M. A. Kramer, “Nonlinear principal component analysis using auto associative neural networks”, in AIChE, pp. 233-243, 1991.

[KAS+20] P. Kazemi, H. Al-Tous, C. Studer, and O. Tirkkonen, “Snr prediction in cellular systems based on channel charting,” in 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), pp. 1–8, 2020.

[KCT+16] P. Kela, M. Costa, J. Turkka, M. Koivisto, J. Werner, A. Hakkarainen, M. Valkama, R. Jantti, and K. Leppanen. “Location based beamforming in 5g ultra-dense networks.” In 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), pages 1–7. IEEE, 2016.

[KHH+22] D. Korpi, M. Honkala, J.M.J. Huttunen, F. A. Aoudia, and J. Hoydis,” Waveform Learning for Reduced Out-of-Band Emissions Under a Nonlinear Power Amplifier,” arXiv:2201.05524 [eess.SP], Jan. 2022, Accessed: Jan. 19, 2022. [Online]. Available: https://arxiv.org/abs/2201.05524

[KKA21] P. Kothari, S. Kreiss, and A. Alahi, “Human trajectory forecasting in crowds: A deep learning perspective,” IEEE Transactions on Intelligent Transportation Systems, pp. 1–15, 2021.

[KMA+21] Kairouz, Peter, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz et al. "Advances and open problems in federated learning." arXiv preprint arXiv:1912.04977 (2019).

[KMM19] K. Koide, J. Miura, and E. Menegatti, “A portable three dimensional LiDAR-based system for long-term and wide area people behavior measurement,” International Journal of Advanced Robotic Systems, vol. 16, no. 2, 2019.

[KMR15] J. Konečný, B. McMahan, and D. Ramage, “Federated Optimization:Distributed Optimization Beyond the Datacenter,” arXiv:1511.03575 [cs, math], Nov. 2015, Accessed: Jun. 08, 2021. [Online]. Available: http://arxiv.org/abs/1511.03575

[KP20] M. Kountouris and N. Pappas, “Semantics-Empowered Communication for Networked Intelligent Systems,” CoRR, vol. abs/2007.11579, 2020.

[KR06] D. R. Karger, and M. Ruhl, „Simple efficient load-balancing algorithms for peer-to-peer systems.” *Theory of Computing Systems*, *39*(6), 787-804, 2006.

[Kri09] A. Krizhevsky, “Learning multiple layers of features from tiny images,” pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[L21] L. Le Magoarou, “Efficient channel charting via phase-insensitive distance computation,” IEEE Wireless Communications Letters, vol. 19, no. 12, 2021.

[LCL+17] Y. Li, Y. Chen, T. Lan and G. Venkataramani, "MobiQoR: Pushing the Envelope of Mobile Edge Computing Via Quality-of-Result Optimization," 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 1261-1270, 2017, doi: 10.1109/ICDCS.2017.54.

[LDL+21] G. Larue, L. -A. Dufrene, Q. Lampin, P. Chollet, H. Ghauch and G. Rekaya, "Blind Neural Belief Propagation Decoder for Linear Block Codes," 2021 Joint

European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2021, pp.106-111, doi: 10.1109/EuCNC/6GSummit51104.2021.9482479.

[LHO+18] Q. Liu, S. Huang, J. Opadere, and T. Han, "An Edge Network Orchestrator for Mobile Augmented Reality," in IEEE INFOCOM 2018- IEEE Conference on Computer Communications, 2018, pp. 756–76

[LML14] Lorido-Botran, T., Miguel-Alonso, J. & Lozano, J.A. A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. J Grid Computing 12, 559–592 (2014). https://doi.org/10.1007/s10723-014-9314-7.

[LSL+22] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," IEEE Journal on Selected Areas in Communications, vol. 40, no. 1, pp. 5–36, 2022

[LSY+20] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," IEEE Transactions on Communications, vol. 68, no. 3, pp. 1760–1776, 2020.

[LTY+19] W. W. Lee, Y. J. Tan, H. Yao, S. Li, H. H. See, M. Hon, K. A. Ng, B. Xiong, J. S. Ho, and B. C. Tee, "A neuro-inspired artificial peripheral nervous system for scalable electronic skins," Science Robotics, 4, 2019.

[LYP+22] L. Le Magoarou, T. Yassine, S. Paquelet, and M. Crussière, "Deep learning for location based beamforming with NLoS channels," In IEEE International Conference on Audio, Speech and Signal Processing (ICASSP), special session on machine learning in beyond 5G wireless networks, 2022.

[LYX+20] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," in China Communications, vol. 17, no. 9, pp. 105-118, Sept. 2020.

[LZZ+19] D. Liu, G. Zhu, J. Zhang, and K. Huang, "Wireless data acquisition for edge learning: Importance-aware retransmission", In International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1-5, July 2019.

[MBD+22] M. Merluzzi, C. Battiloro, P. Di Lorenzo, E. Calvanese Strinati, "Energy-Efficient Classification at the Wireless Edge with Reliability Guarantees," accepted at IEEE Globecom 2022, available online: https://arxiv.org/abs/2204.10399

[MBM+20] Mahmood, N., Böcker, S., Munari, A., Clazzer, F., Moerman, I., Mikhaylov, K., Lopez, O., Park, O.S., Mercier, E., Bartz, H., Jäntti, R., Pragada, R., Ma, Y., Annanperä, E., Wietfeld, C., Andraud, M., Liva, G., Chen, Y., Garro, E., Burkhardt, F., Alves, H., Liu, C.F., Sadi, Y., Dore, J.B., Kim, E., Shin, J., Park, G.Y., Kim, S.K., Yoon, C., Anwar, K., & Seppänen, P. "White Paper on Critical and Massive Machine Type Communication Towards 6G". 2020. https://doi.org/10.48550/arXiv.2004.14146.

[MDE10] R. Maiberger, D. Ezri, and M. Erlihson, "Location based beamforming." In 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel, pages 000184–000187. IEEE, 2010.

[MLE21] V. Monga, Y. Li, and Y.C. Eldar. “Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing.” IEEE Signal Processing Magazine, 38(2):18–44, 2021.

[MRL21] D. Marasinghe, N. Rajatheva and M. Latva-aho, "LiDAR Aided Human Blockage Prediction for 6G," 2021 IEEE Globecom Workshops (GC Wkshps), 2021, pp. 1-6, doi: 10.1109/GCWkshps52748.2021.9681949.

[MRR17] G. R. MacCartney, T. S. Rappaport, and S. Rangan, “Rapid fading due to human blockage in pedestrian crowds at 5G millimeter-wave frequencies,” in IEEE GLOBECOM 2017, pp. 1–7, 2017.

[MSC+18] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, “Exploiting Unintended Feature Leakage in Collaborative Learning,” arXiv:1805.04049 [cs], Nov. 2018, Accessed: Jun. 29, 2021. [Online]. Available: http://arxiv.org/abs/1805.04049

[MZ93] S.G. Mallat and Z. Zhang. “Matching pursuits with time-frequency dictionaries” IEEE Transactions on Signal Processing, 41(12):3397– 3415,1993.

[NML+18] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be’ery, “Deep learning methods for improved decoding of linear codes”, in IEEE Journal of Selected Topics in Signal Processing, vol 12, no. 1, pp. 119–131, 2018.

[NTD+18] H. Q. Ngo, L. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, "On the Total Energy Efficiency of Cell-Free Massive MIMO," in IEEE Transactions on Green Communications and Networking, vol. 2, no. 1, pp. 25-39, March 2018.

[NWU18] D. Neumann, T. Wiese, and W. Utschick, “Learning the MMSE channel estimator,” IEEE Trans. Signal Process., Vol. 66, No. 11, pp. 2905– 2917, Jun. 2018.

[OH17] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer” in IEEE Transactions on Cognitive Communications and Networking, vol.3, pp. 563–575, 2017.

[OWY+21] Ouyang, Y., Wang, L., Yang, A., Shah, M., Belanger, D., Gao, T., Wei, L., & Zhang, Y. “The Next Decade of Telecommunications Artificial Intelligence”. https://doi.org/10.48550/arXiv.2101.09163

[PML+19] J. Portilla, G. Mujica, J. -S. Lee and T. Riesgo, "The Extreme Edge at the Bottom of the Internet of Things: A Review," in IEEE Sensors Journal, vol. 19, no. 9, pp. 3179-3190, 1 May1, 2019, doi: 10.1109/JSEN.2019.2891911.

[PRW+02] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “BLEU: A Method for Automatic Evaluation of Machine Translation,” in Proc. Association for Computational Linguistics (ACL’02), pp. 311–318, 2002

[PS08] John G. Proakis and Masoud Salehi, “Digital communications”, McGraw-Hill, 2008.

[QR20] Q. Wu and R. Zhang, "Towards Smart and Reconfigurable Environment: Intelligent Reflecting Surface Aided Wireless Network," in IEEE Communications Magazine, vol. 58, no. 1, pp. 106-112, January 2020, doi: 10.1109/MCOM.001.1900107.

[RCV+22] Daniel Rosendo, Alexandru Costan, Patrick Valduriez, & Gabriel Antoniu (2022). Distributed intelligence on the Edge-to-Cloud Continuum: A systematic literature review. Journal of Parallel and Distributed Computing, 166, 71-94.

[RCZ+18] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, “DeepDecision: A Mobile Deep Learning Framework for Edge Video Analytics,” in IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, pp. 1421–142, 2018.

[RDG+15] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.. (2015). You Only Look Once: Unified, Real-Time Object Detection, doi: 10.48550/arXiv.1506.02640.

[RDG+21] A. Renda, P. Ducange, G. Gallo and F. Marcelloni, "XAI Models for Quality of Experience Prediction in Wireless Networks," 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-6, 2021 doi: 10.1109/FUZZ45933.2021.9494509.

[RDS21] Rajesh Gupta, Dakshita Reebadiya, & Sudeep Tanwar (2021). 6G-enabled Edge Intelligence for Ultra -Reliable Low Latency Applications: Vision and Mission. Computer Standards & Interfaces, 77, 103521.

[RLD+20] L. Ribeiro, M. Leinonen, H. Djelouat, and M. Juntti, “Channel charting for pilot reuse in mmtc with spatially correlated mimo channels,” in 2020 IEEE Globecom Workshops (GC Wkshps, 2020, pp. 1–6.

[RMR+21] N. Rajapaksha, K. B. Shashika Manosha, N. Rajatheva, M. Latva-aho, "Deep learning-based power control for cell-free massive MIMO networks", IEEE International Conference on Communications (ICC), 2021, pp. 1–7.[RRK+19] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, ”Events-to-video: Bringing modern computer vision to event cameras,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[RRL21] V. Ranasinghe, N. Rajatheva and M. Latva-aho, "Graph neural network based access point selection for cell-free massive MIMO systems," 2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp. 01-06, doi: 10.1109/GLOBECOM46510.2021.9685221.

[SAI007] https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=63078

[SAI008] https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=63499

[SAI010] https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=63960

[SB18] R. S. Sutton, A. G. Barto, “Reinforcement learning: An introduction”. MIT press, 2018.

[SC21] M. Sana and E. C. Strinati, "Learning Semantics: An Opportunity for Effective 6G Communications," *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 631-636, doi: 10.1109/CCNC49033.2022.9700645.

[SH16] Santana, E., & Hotz, G.. (2016). Learning a Driving Simulator, doi: 10.48550/ARXIV.1608.01230.

[Sha48] C. E. Shannon, “A mathematical theory of communication,” The Bell system technical journal, vol. 27, no. 3, pp. 379–423, 1948.

[SHAP] S. M. Lundberg, and S. I. Lee. "A unified approach to interpreting model predictions." *Proceedings of the 31st international conference on neural information processing systems*. 2017.

[She20] Alex Sherstinsky. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 13230, 2020. https://doi.org/10.1016/j.physd.2019.132306

[SHH22] SShimaa A. Abdel Hakeem, Hanan H. Hussein, HyungWon Kim, "Vision and research directions of 6G technologies and applications", Journal of King Saud University - Computer and Information Sciences, 2022, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2022.03.019.

[SM01] M. Islam, Shahjahan, and K. Murase, "A new weight freezing method for reducing training time in designing artificial neural networks" in IEEE International Conference on Systems, Man and Cybernetics, pp. 341-346, 2001.

[SMG+18] C. Studer, S. Medjkouh, E. Gonültas¸ T. Goldstein, and O. Tirkkonen, ¨ "Channel charting: Locating users within the radio environment using channel state information," IEEE Access, vol. 6, pp. 47 682–47 698, 2018.

[SMP17] A. Segatori, F. Marcelloni, and W. Pedrycz, "On distributed fuzzy decision trees for big data," IEEE Trans. Fuzzy Syst., vol. 26, no. 1, pp. 174–192, 2017.

[SR16] M. K Samimi and T. S Rappaport. "3-D millimeter-wave statistical channel model for 5g wireless system design." IEEE Transactions on Microwave Theory and Techniques, vol. 64, no. 7:2207–2225, 2016.

[SSS+17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," arXiv:1610.05820 [cs, stat], Mar. 2017, Accessed: Jun. 27, 2021. [Online]. Available: http://arxiv.org/abs/1610.05820

[SSS+17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," arXiv:1610.05820 [cs, stat], Mar. 2017, Accessed: Jun. 27, 2021. [Online]. Available: http://arxiv.org/abs/1610.05820

[STS16] A. Singh, N. Thakur and A. Sharma, "A review of supervised machine learning algorithms," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 1310-1315.

[TDL00] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," science, vol. 290, no. 5500, pp. 2319–2323, 2000.

[TF22] TensorFlow Developers. (2022). TensorFlow (v2.8.2). Zenodo. https://doi.org/10.5281/zenodo.6574269.

[TMW+18] B. Taylor, V. S. Marco, W. Wolff, Y. Elkhatib, and Z. Wang, "Adaptive Deep Learning Model Selection on Embedded Systems," SIGPLAN Not., vol. 53, no. 6, p. 31–43, June 2018. [Online]. Available: https://doi.org/10.1145/3299710.3211336

[TR 23.700-80] 3GPP TR 23.700-80, Study on 5G system support for AI/ML-based services, Release 18, v0.10. Feb. 2022. Online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4009

[TR22.874] 3GPP TR 22.874, 5G System (5GS); Study on traffic characteristics and performance requirements for AI/ML model transfer, Release 18, v18.2.0, Dec. 2021. Online: https://www.3gpp.org/ftp/Specs/archive/22_series/22.874/22874-i20.zip

[TR22.875] 3GPP TR 22.875, Study on AI/ML Model Transfer Phase 2 (Rel. 19). Online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4043

[TR28.908] 3GPP TR 28.908, Study on Artificial Intelligence/Machine Learning (AI/ ML) management (Rel. 18), v0.0.0, Mar. 2022. Online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3965

[TR33.852] 3GPP TR 33.852, Study on traffic characteristics and performance requirements for AI/ML model transfer in 5G Systems (5GS) (Rel. 18). Online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3766

[TR38.843] 3GPP TR 38.843, Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR air interface, Rel. 18. Online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3983

[TRS22] Tareq B. Ahammed, Ripon Patgiri, Sabuzima Nayak, "A vision on the artificial intelligence for 6G communication", ICT Express, 2022, ISSN 2405-9595, https://doi.org/10.1016/j.icte.2022.05.005.

[TS22] Tsozen Yeh, & Shengchieh Yu (2022). Realizing dynamic resource orchestration on cloud systems in the cloud-to-edge continuum. Journal of Parallel and Distributed Computing, 160, 100-109.

[TS23.288] 3GPP Technical Specification 23.288, Architecture enhancements for 5G System (5GS) to support network data analytics services (Rel. 17), v17.4.0, Mar. 2022. Online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579

[TS28.104] 3GPP TS 28.104, Management and orchestration; Management Data Analytics (Rel. 17), v1.0.0, Mar. 2022. Online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3877

[TS28.105] 3GPP TS 28.105, Management and orchestration; Artificial Intelligence/ Machine Learning (AI/ML) management, Release 18, v1.0.0, Mar. 2022. Online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3970

[TS33.521] 3GPP TS 33.521, 5G Security Assurance Specification (SCAS);Network Data Analytics Function (NWDAF) (Rel. 17), v17.1.0, Sep. 2021. Online:

https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3749

[TSM+20] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. "Fourier features let networks learn high frequency functions in low dimensional domains." arXiv preprint arXiv:2006.10739, 2020.

[TZJ+16] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing Machine Learning Models via Prediction APIs," arXiv:1609.02943 [cs, stat], Oct. 2016, Accessed: Jun. 29, 2021. [Online]. Available: http://arxiv.org/abs/1609.02943

[VLP+18] V. Vasilev, J. Leguay, S. Paris, L. Maggi, and M. Debbah, "Predicting QoE factors with machine learning," in 2018 IEEE Int'l Conf. on Communications (ICC). IEEE, 2018, pp. 1–6.

[VSP+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems, pp. 5998–6008, 2017.

[WBK+05] A. Williams, S. Barrus, R. K. Morley and P. Shirley, "An efficient and robust ray-box intersection algorithm," in ACM SIGGRAPH 2005 Courses, ser. SIGGRAPH '05, Los Angeles, California: Association for Computing Machinery, 2005, 9–es.

[Wea53] W. Weaver, "Recent contributions to the mathematical theory of communication," ETC: a review of general semantics, vol. 10, no. 4, pp. 261–281, 1953.

[Wit14] Peter Wittek, 5 - Unsupervised Learning, Quantum Machine Learning, Academic Press, 2014, Pages 57-62, ISBN 9780128009536, https://doi.org/10.1016/B978-0-12-800953-6.00005-0..

[WQL20] Z. Weng, Z. Qin, and G. Y. Li, "Semantic Communications for Speech Signals," arXiv preprint arXiv:2012.05369, 2020

[XQL+21] H. Xie, Z. Qin, G. Y. Li, and B. H. Juang, "Deep Learning Enabled Semantic Communication Systems," IEEE Transactions on Signal Processing, pp. 1–1, 2021.

[YL22] T. Yassine and L. Le Magoarou, "mpNet: variable depth unfolded neural network for massive MIMO channel estimation", IEEE Transactions on Wireless Communications, 2022.

[YLL19] H. Ye, L. Liang and G. Y. Li, "Circular convolutional auto-encoder for channel coding", in proceedings of IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, pp. 1-5, 2019.

[YM15] S. Yan and R. Malaney. "Location-based beamforming for enhancing secrecy in rician wiretap channels." IEEE Transactions on Wireless Communications, 15(4):2780–2791, 2015.

[YSH+20] Yu, L., Sartran, L., Huang, P.-S., Stokowiec, W., Donato, D., Srinivasan, S., Andreev, A., Ling, W., So, S., Mokrá, S., Dal, A., Doron, L. Y., Young, S., Blunsom, P., Dyer, C.. "The DeepMind Chinese-English Document Translation

System" Proceedings of the 5th Conference on Machine Translation (WMT), , pages 326–337, November 2020.

[YW22] J. Yang and Y. Wan, "The development trend of artificial intelligence in the big data environment," 2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI), 2022, pp. 301-304, doi: 10.1109/IWECAI55315.2022.00064.

[ZCL+19] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," in *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738-1762, Aug. 2019, doi: 10.1109/JPROC.2019.2918951.

[ZLH+19] H. Zhao, H. Lim, M. Hanif and C. Lee, "Predictive Container Auto-Scaling for Cloud-Native Applications," 2019 International Conference on Information and Communication Technology Convergence (ICTC), 2019, pp. 1280-1282, doi: 10.1109/ICTC46691.2019.8939932.

[ZLH19] L. Zhu, Z. Liu, and S. Han, "Deep Leakage from Gradients," arXiv:1906.08935 [cs, stat], Dec. 2019, Accessed: Jun. 29, 2021. [Online]. Available: http://arxiv.org/abs/1906.08935

[ZNG20] Y. Zhao, I. G. Niemegeers, and S. H. De Groot, "Power allocation in cell-free massive MIMO: A deep learning method," IEEE Access, vol. 8, pp. 87 185–87 200, 2020.

[ZSB+21] Z. Zvara, P. G. Szabó, B. Balázs, and A. A. Benczúr, "System-aware dynamic partitioning for batch and streaming workloads," UCC '21: 2021 IEEE/ACM 14th International Conference on Utility and Cloud Computing, Leicester, United Kingdom, December 6 - 9, 2021. ACM 2021, ISBN 978-1-4503-8564-0

Annex A: Ongoing standardisation activities with relevance to in-network AI/ML

Standards Developing Organisations (SDOs), both international ones, such as 3GPP [3GPP], the International Telecommunications Union (ITU) [ITU], the Internet Research Task Force (IRTF) [IRTF] and regional ones, such as the European Telecommunications Standards Institute (ETSI) [ETSI] have recently ramped up their interest in defining/ specifying system and network functionalities with respect to an AI/ML-capable cellular network, each one from a different perspective.

Tables A-1 and A-2 summarise the so far undertaken and current specification activity in 3GPP and ETSI ISG SAI. In terms of relevance to the technical areas covered in deliverable D4.2, focusing on 3GPP recent and ongoing work there are certain synergies both regarding AI/ML-based air interface design and (more intensively) aspects relating to network architecture, management and orchestration when considering in-network AI/ML functionality. Table A-1 provides an overview of relevant specification activities in 3GPP.

Security, privacy and trust aspects (including explainability) seem to be under the spotlight of ETSI specifications (Industry Specification Group - ISG on Securing AI - SAI), while, when it comes to automated network operation, service management and orchestration, specifications have been developed within the ISG on Experiential Network Intelligence (ENI) and Zero touch network & Service Management (ZSM). Table A-2 provides an overview of current specification activities in ETSI ISG SAI (currently open Work Items).

Apart from 3GPP and ETSI, other organisations, such as ITU and IRTF are, at least partly focused on the topic of AI/ML enablement in communication networks. For example, IRTF Computing in the Network Research Group (coinrg) [COINRG] has under its scope the focus on the evolution necessary for networking to move beyond packet interception as the basis of network operation and into computation (relevance to topics covered in Chapter 3 of D4.2). Recent and current activity of ITU, on the other hand refers to: (i) the Focus Group on AI for autonomous and assisted driving (FG-AI4AD) [AI4AD] and (ii) the Focus Group on Autonomous Networks (FG-AN) [FGAN].

Aspects covered by deliverable D4.2 and seemingly uncovered by the so far developed SDO activity, may relate to a number of technical areas/ 6G enablers, such as management of multi-agent learning architectures (i.e., beyond FL), E2E learning for air interface design, semantic and explainability protocols as well as definition of a communication "goal", possibly extending the network slicing concept. APIs needed to realise the AIaaS and CaaS concepts may also need to be specified. Of course, D4.2 (and the whole work of Hexa-X WP4) is focused on the design of algorithms and methodologies to enable AI-pervasive 6G networking, but, at least some aspects may be expected to be part of the future 6G standard from a protocol design and architecture point of view (mostly relevant to Hexa-X WP5 work with which WP4 is closely collaborating with).

Last, but not the least, in Europe, the AI Regulation [AIREG] has been recently developed aiming to lay down harmonised rules on AI and amend certain European legislative acts. This is a very important development, as it will have an impact on how AI (including ML) systems will need to operate to comply with the AI Regulation in Europe.

Table A-1: some 3GPP Technical Reports (TRs)/ Technical Specifications (TSs) of relevance to D4.2 work.

Specification type & number	Specification title	Primary responsible Group & Release	Scope	Reference	D4.2 section(s) of closest topic relevance
TR 38.843	Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR air interface	RAN1	Not available yet - in draft status.	[TR38.843]	2.1, 3.2.3, 4.1, 4.2
TR 22.874	5G System (5GS); Study on traffic characteristics and performance requirements for AI/ML model transfer	SA1 (Rel.18)	"This report captures the study of the use cases and the potential performance requirements for 5G system support of Artificial Intelligence (AI)/Machine Learning (ML) model distribution and transfer (download, upload, updates, etc.), and identifies traffic characteristics of AI/ML model distribution, transfer and training for various applications, e.g. video/speech recognition, robot control, automotive, other verticals. The aspects addressed include: • AI/ML operation splitting between AI/ML endpoints; • AI/ML model/data distribution and sharing over 5G system; • Distributed/Federated Learning over 5G system. Study of the AI/ML models themselves are not in the scope of the TR".	[TR22.874]	3.1, 3.2

TR 33.852	Study on traffic characteristics and performance requirements for AI/ML model transfer in 5G Systems (5GS)	SA1 (Rel.18)	Not available yet - in draft status.	[TR33.852]	3.1, 3.2
TR 22.875	Study on AI/ML Model Transfer Phase 2	SA1 (Rel.19)	Not available yet - in draft status.	[TR22.875]	3.1, 3.2
TS 23.288	Architecture enhancements for 5G System (5GS) to support network data analytics services	SA2 (Rel. 17)	"The present document defines the Stage 2 architecture enhancements for 5G System (5GS) to support network data analytics services in 5G Core network".	[TS23.288]	2.2.2
TR 23.700-80	Study on 5G system support for AI/ML-based services	SA2 (Rel.18)	"This Technical Report will study, […], 5GS assistance to support Artificial Intelligence (AI) / Machine Learning (ML) model distribution, transfer, training for various applications, e.g. video/speech recognition, robot control, automotive, etc. The scope of this study is on how the AI/ML service providers could leverage 5GS as the platform to provide the intelligent transmission support for application layer AI/ML operation […]".	[TR 23.700-80]	3.1, 3.2
TS 33.521	5G Security Assurance Specification	SA3 (Rel. 17)	"The present document contains requirements and test cases that are specific	[TS33.521]	Chapter 5 as a whole

	(SCAS);Network Data Analytics Function (NWDAF)		to the NWDAF network product class. It refers to the Catalogue of General Security Assurance Requirements and formulates specific adaptions of the requirements and test cases, as well as specifying requirements and test cases unique to the NWDAF network product class".		
TS 28.104	Management and orchestration; Management Data Analytics	SA5 (Rel. 17)	"The present document specifies the MDA capabilities with corresponding analytics inputs and analytics outputs (reports), as well as processes and requirements for MDAS (Management Data Analytics Service), historical data handling for MDA, and ML support for MDA. This document also describes the MDA functionality and service framework, and MDA role in the management loop".	[TS28.104]	2.2
TS 28.105	Management and orchestration; Artificial Intelligence/ Machine Learning (AI/ML) management	SA5 (Rel. 18)	"The present document specifies the Artificial Intelligence / Machine Learning (AI/ML) management capabilities and services for 5GS where AI/ML is used, including management and orchestration […]. This document also describes the functionality and service framework for AI/ML management".	[TS28.105]	2.2
TR 28.908	Study on Artificial Intelligence/ Machine	SA5 (Rel. 18)	Not available yet - in draft status.	[TR28.908]	Chapter 3 as a whole

	Learning (AI/ ML) management				

Table A-2: some active ETSI ISG SAI Work Items of relevance to D4.2 work.

Work Item title	**Group Report (GR)/ Group Specification (GS) number**	**ETSI ISG**	**Work Item scope**	**Reference**	**D4.2 section(s) of closest topic relevance**
Explicability and transparency of AI processing	GR SAI-007	SAI	"The intent of this work item is to extend from the published work of SAI to address the issues of design of AI platforms (data, algorithms, frameworks) that are able to give assurance of explainability and transparency of decisions. This is intended in part to also consider the impact of issues arising from regulation of AI to address ethics and misuse and to allow independent determination of bias (a light touch). The report will address both intrinsic and post-hoc analysis of AI systems".	[SAI007]	5.2
Privacy aspects of AI/ML systems	GR SAI-008	SAI	"The purpose of this work item is to identify the role of privacy as one of the components of the Security of AI and proceed with the attempt to define Privacy in the context of AI that covers both, safeguarding models and protecting data, as well as the role of privacy-sensitive data in AI solutions. It investigates and addresses the attacks and their associated remediations where applicable, considering the	[SAI008]	5.1

			existence of multiple levels of trust affecting the lifecycle of data".		
Traceability of AI Models	GR SAI-010	SAI	"The NWI will study the role of traceability in the challenge of Securing AI and explore issues related to sharing and re-using models across tasks and industries. The scope includes threats, and their associated remediations where applicable, to ownership rights of AI creators as well as to verification of models origin, integrity or purpose. Mitigations can be non-AI-Specific (Digital Right Management applicable to AI) and AI-specific techniques (e.g. watermarking) from prevention and detection phases. They can be both model-agnostic or model enhancement techniques. Threats and mitigations specific to the collaborative learning setting, implying multiple data and model owners, could be also explored […]".	[SAI010]	5.1 overall