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 D4.3
AI-driven communication & computation co-design: final solutions

Date of delivery:	30/04/2023	Version:	1.0
Start date of project:	01/01/2021	Duration:	30 months

Document properties:

Document Number:	D4.3
Document Title:	AI-driven communication & computation co-design: final solutions
Editor(s):	Tamás Borsos (EHU), Mattia Merluzzi (CEA) Alessandro Renda (UPI), Johan Haraldson (EAB), Nandana Rajatheva (OUL), Jafar Mohammadi (NOG), András Benczúr (SZT), Leyli Karaçay (EBY)
Authors:	Hamed Farhadi (EAB), Johan Haraldson (EAB), Heunchul Lee (EAB), Jaeseong Jeong (EAB), Seder Erin (NXW), Piscione Pietro (NXW), Tamás Borsos (EHU), Markus Dominik Mueck (INT), Miltiadis Filippou (INT), Leonardo Gomes Baltar (INT), Mattia Merluzzi (CEA), Emilio Calvanese Strinati (CEA), Vasiliki Lamprousi (WIN), Sokratis Barmpounakis (WIN), Ioannis-Prodromos Belikaidis (WIN), Panagiotis Demestichas (WIN), Taha Yassine (BCO), Alessio Behcini (UPI) Pietro Ducange (UPI) Francesco Marcelloni (UPI) Alessandro Renda (UPI), Leyli Karaçay (EBY), Nandana Rajatheva (OUL), Nuwanthika Rajapaksha (OUL), Dilin Dampahalage (OUL), Nipuni Ginige (OUL), Adrián Gallego Sánchez (ATO), Jafar Mohammadi (NOG), András Benczúr (SZT), Dani Korpi (NOF), Quentin Lampin (ORA)
Contractual Date of Delivery:	30/04/2023
Dissemination level:	PU[1]/
Status:	<Final>
Version:	1.0
File Name:	Hexa-X D4.3_v1.0

Revision History

Revision	Date	Issued by	Description
0.1	18.10.2022	Hexa-X WP4	Initial contributions
0.2	09.12.2022	Hexa-X WP4	Additional contributions
0.3	16.01.2023	Hexa-X WP4	Addressing comments and additional contributions in tech enablers
0.4	19.01.2023	Hexa-X WP4	External review 1st round
0.5	20.02.2023	Hexa-X WP4	External review 2nd round

[1] CO = Confidential, only members of the consortium (including the Commission Services)

PU = Public

0.6	11.03.2023	Hexa-X WP4	PMT review
0.7	30.03.2023	Hexa-X WP4	GA approval
1.0	27.04.2023	Hexa-X WP4	Final version

Abstract

This report describes AI/ML mechanisms and algorithms for enhanced performance and sustainable operation of 6G systems and provide design concepts for 6G networks to become an intelligent and trustworthy platform for in-network and application-level AI functions.

Keywords

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

Disclaimer

The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect views of the whole Hexa-X Consortium, nor the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein

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

Executive Summary

This report is the third and final deliverable of project Hexa-X work package four (WP4) “AI-driven communication and computation co-design”. The main objectives of WP4 are to develop AI/ML mechanisms in 6G systems, demonstrating performance efficiency, feasible complexity and unprecedented adaptivity in wireless transceiver and other air interface functionalities, and to provide design concepts for 6G networks to become an intelligent and sustainable platform for in-network and application-level AI functions. These objectives are addressed in this document.

AI/ML-based network performance enhancement methods analysed in this report show significant benefits in spectral efficiency, channel coding and beam management overhead, as well as demonstrating capabilities to compensate the distortions caused by hardware impairments, especially on higher frequency bands of 6G. These challenges are studied from multiple viewpoints: using end-to-end learning techniques to jointly optimize transmitter and receiver side RAN functions, but also by relying on compact receive or transmit side algorithms. Performance improvements introduced by AI solutions are also shown in the management and orchestration network domain. In the comprehensive view of using AI for performance enhancement functions, it is also essential to investigate the complexity and energy footprint of these solutions. These questions are addressed as part of the 6G network sustainability investigation, with well-designed AI architectures demonstrating improved energy efficiency without significant compromises in the performance indicators. The view of network sustainability is further strengthened by providing reduced complexity solutions for optimization problems practically intractable by conventional methods.

We envision 6G networks to become an intelligent and trustworthy platform for AI applications running either in a network domain or over the top. Design concepts for efficient execution are described from three perspectives: supporting network services, challenges of model training and real-time inference functions. Network services are proposed with open interfaces and data structures derived by the requirements of AI applications. Distributed training with special focus on federated learning is investigated with the aim of reducing and balancing the huge data load that would strain the network. On the inference side the challenges of distributed operation, high device density, low latency and high accuracy requirements are addressed by joint communication and compute solutions. The trustworthy operation of AI functions is also covered in the report by enablers for designing AI systems that are transparent, explainable, reliable, and fair, while ensuring data privacy and security. Practical application of the algorithmic solutions for explainable federated learning is demonstrated as part of the joint WP4+WP5 Fed-XAI project demo.

Finally, the targets for the Hexa-X objective Connecting intelligence towards 6G are addressed and mapped to the concepts and solutions described in the document.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Term	Description
3GPP	3rd Generation Partnership Project
6G	Sixth-generation wireless
AE	Autoencoder
AI	Artificial Intelligence
AIaaS	AI-as-a-Service
AIS	AI Service
AoA	Angle of Arrivals
AP	Access Point
API	Application Programming Interface
ARIMA	Auto Regressive Integrated Moving Average
BER	Bit Error Rate
BLER	Block Error Rate
BS	Base Station
BYOM	Bring Your Own Model
CaaS	Compute-as-a-Service
CNF	Containerized Network Function
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CS	Compressed Sensing
CSI	Channel State Information
CTDE	Centralised Training and Decentralised Execution
DFT-s-OFDM	Discrete Fourier transform-spread orthogonal frequency-division multiplexing
DNN	Deep Neural Network
DSP	Digital Signal Processor
E2E	End-to-End
EC	European Commission
FL	Federated Learning
FPGA	Field-Programmable Gate Array
FR1	Frequency Range 1
FR2	Frequency Range 2
GNSS	Global Navigation Satellite System
GPU	Graphics Processing Unit
H2020	Horizon 2020
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Technologies
I/O	Input/ Output

KPI	Key Performance Indicator
LCM	Lifecycle Manager
LiDAR	Light Detection and Ranging
LISTA	Learned Iterative Shrinkage and Thresholding Algorithm
LoS	Line-of-Sight
LSTM	Long-Short Term Memory
LTF	Long Term Forecasting
M&O	Management and Orchestration
MCS	Modulation & Coding Scheme
MDAF	Management Data Analytics Function
MEC	Multi-access Edge Computing
MIMO	Multiple-Input and Multiple-Output
ML	Machine Learning
MTF	Mid Term Forecasting
NLoS	Non-Line-of-Sight
NMSE	Normalized Mean Square Error
NWDAF	Network Data Analytics Function
NN	Neural Network
NPU	Neural Processing Unit
QoS	Quality-of-Service
RAN	Radio Access Network
RAP	Radio Application Package
REST	Representational State Transfer
RIS	Reconfigurable Intelligent Surface
RNN	Recurrent Neural Networks
RRM	Radio Resource Management
SDO	Standards Definition Organization
SNN	Spiking Neural Network
SP	Service Provider
STF	Short Term Forecasting
SVM	Support Vector Machine
UCF	Use Case Family
UE	User Equipment
UPF	User Plane Function
V2X	Vehicle-to-Everything
VM	Virtual Machine

1 Introduction

Hexa-X is the flagship project of 5G-PPP under EU Horizon2020 which develops a Beyond 5G (B5G)/ 6G vision and an intelligent fabric of technology enablers connecting human, physical, and digital worlds.

This is the final 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". Relevant background information has been documented in [HEX-D41] "AI-driven communication & computation co-design: Gap analysis and blueprint". The second deliverable "AI-driven communication & computation co-design: initial solutions" [HEX-D42] focuses on the technical areas related to applying AI/ML to networking and concentrates on associated technical enablers to be investigated within WP4. This final deliverable concludes the investigations of technical areas from D4.2 and summarizes the results, as well as their impact on Hexa-X defined use cases, Key Performance Indicators (KPI), Key Value Indicators (KVI) and quantified targets.

1.1 Objective of the document

The objective of this document is to describe AI-driven solutions and frameworks applied in various 6G network domains. These solutions can be AI/ML algorithms to support a specific network function (e.g., in RAN, O&M) or a joint communication-compute concept to enable efficient execution of AI applications or in-network functions. The report, therefore, presents a detailed investigation of the technical enablers and their impact on the relevant Key Performance Indicators.

Specifically, the document addresses the main work package objectives.

WPO4.1	Elaborate motivations for the application of AI/ML mechanisms in B5G/6G systems and identify the needed challenges to be addressed by these systems during a related technology roadmap.
WPO4.2	Design B5G/6G wireless transceivers and air interface functionalities in a cost and complexity-efficient manner, either by replacing specific model-based parts of the transmission/ reception chain by their AI/ML counterparts, or by adopting an intelligent "end-to-end" optimisation approach.
WPO4.3	Develop data-driven methods for unprecedented adaptivity to the wireless environment and radio hardware impairment mitigation to maximise radio access flexibility and configurability.
WPO4.4	Design concepts for an intelligent and sustainable B5G/6G distributed platform capable to jointly optimise communication, computing & storage resources across energy-aware network elements.
WPO4.5	Provide a communication framework for federated learning (FL) communications, while guaranteeing accuracy and explainability of the models, along with data integrity, security, privacy and trust.
WPO4.6	Define ML applicability contexts for predictive management and orchestration for use in SDN and OAM.

WPO4.1 was addressed in [HEX-D41], where the state of the art, motivating challenges and targeted benefits were presented, including the role of data in 6G AI systems with data quality, quantity, availability, ownership, and monetisation. The challenges have been identified in the areas of AI-based air interface design and network supported AI functions.

WPO4.2 is addressed in Chapter 3 and Chapter 4 where AI/ML methods were implemented in block-by-block fashion or end-to-end approach. The AI/ML solutions for modular level functions include channel encoding/decoding, channel estimation, as well as beamforming solutions where performance improvements and complexity/overhead reductions are achieved by utilising the proposed AL/ML implementations. Furthermore, the low complexity end-to-end solutions are presented considering sub-THz systems and RIS-assisted communication systems which are applicable to new 6G use cases.

WPO4.3 is also addressed in Chapter 3 and Chapter 4 where attention is paid to hardware impairment mitigation and consideration of techniques for addressing channel variations with online learning methods. Specifically, the AI/ML-based PA non-linearity compensation approach in Chapter 3 is shown to improve throughput, extend the communication coverage, and to enhance energy efficiency. Adaptive and flexible AI/ML solutions focusing on generalizable channel estimation of unknown channel models and channel estimation in RIS-assisted systems under mobility, and flexible radio resource allocation in cell-free massive MIMO are also presented.

WPO4.4 is addressed by the concepts presented in Chapter 5. Native integration of AI computing capabilities in 6G networks enable new latency, compute, and energy efficiency optimization strategies for AI functions by exploiting network knowledge. New network services and data exposure capabilities are also described that will facilitate the efficient deployment of such AI applications.

WPO4.5 is addressed in Chapter 5 with a focus on AI as an efficient platform for 6G, in Chapter 6 with a focus on privacy, security & trust in AI-enabled 6G and Chapter 7 related to demonstration activities. Chapter 5 specifically introduces a federated ML model load balancing at the edge and a Frugal Federated Learning approach. Chapter 6 provides solutions for Security mechanism friendly privacy solutions for federated learning and Federated Learning of Explainable AI models. Chapter 7 finally introduced a Federated eXplainable AI (FED-XAI) demo.

WPO4.6 is addressed in section 3.2 with improvements in E2E network operation and management. Based on the gap analysis on the current M&O solution in 5G network, the technical enablers mainly based on the AI/ML techniques have been evaluated and described. These technical enablers focus on the predictive state of the network and the network function data plane (i.e., the User Plane Function (UPF)) using the data collected from the custom monitoring platform and NWDAF, to make more efficient usage of the allocated resources. For each of the different AI/ML techniques, evaluations and results are reported in the corresponding sections. Additionally, predictive orchestration AI/ML algorithms have been analysed and mapped to the Hexa-X M&O architecture to create a potential baseline for future predictive orchestration frameworks and implementations.

1.2 Contributions to the Hexa-X objective on Connecting intelligence towards 6G

WP4 is also contributing to the fulfilment of the Hexa-X objective on Connecting intelligence towards 6G with the following outputs, measurable and quantified results.

Introduce AI enablers as part of the B5G/6G air interface design, with emphasis on wireless transceiver and signal transmission/reception design and exposed performance benefits:

> Initial solutions for AI/ML-based enablers for B5G/6G air interface design were presented in D4.2 which provide radio access network performance improvements over classical design methods in terms of bit error rate, spectral efficiency, reliability, and resource utilisation. Specifically, AI/ML enablers for communication reliability improvements in D4.2 Section 2.1 proposed several AI/ML enablers for communication reliability improvements such as LiDAR-aided human blockage prediction, access point selection in cell-free massive MIMO for initial access and mobility management, and compressed sensing-based beam selection in distributed MIMO. AI/ML solutions for bit rate and spectral efficiency improvements proposed in the same section include design concepts and implementations for constellation shaping, channel

(de)coding for constrained devices, location-based beamforming, and beam management in initial access. In D4.3, these initial solutions are further improved along with new investigations in end-to-end learning of RIS-assisted communications, AI-native air interface design for sub-THz communications, and channel charting-based beamforming, which are presented in section 3.1. Furthermore, D4.2 section 4.1 and D4.3 section 4 proposed different channel charting and channel estimation solutions including channel estimation in RIS-assisted systems.

Deliver efficient (with respect to sustainability, security, privacy, and trust-awareness) management and governance of AI mechanisms, functions and agents, considering system-wide constraints of processing, storage, memory, data and networking resources:

Design concepts are provided for efficient global operation of distributed AI/ML mechanisms in 6G networks with focus on performance, sustainability, and trustworthiness. Requirements and solutions for 6G as an efficient AI platform are described in section 3 and 4 of D4.2 and section 5 of D4.3. It covers network services and interfaces for AI, efficient compute and communication resource usage for distributed AI inference and federated learning at the edge, and optimal allocation of AI workloads depending on communication load, power usage and trust levels. Section 5 of D4.2 and section 6 of D4.3 provides privacy, security and trust mechanisms for AI/ML: privacy solutions for federated learning, defence mechanisms against adversarial attacks, and explainable AI for mitigation of biased decisions in federated learning systems. AI-based management and orchestration solutions are provided in section 2.2 of D4.2 and section 3.2 of D4.3: solution for predictive scaling of system resources by introducing distributed deployment of AI agents at the network edge, and a framework for predictive orchestration.

Designs for data-driven wireless transceiver of low complexity, either "block-per-block", or, by means of "end-to-end" optimisation:

AI/ML approaches which enable low complexity transceivers by joint or end-to-end optimisation of the signal processing blocks is of great interest in AI-based air interface design. To this end, section 2.1 in D4.2 proposed a joint constellation and receiver learning mechanism for sub-6-GHz communications which enables pilotless transmissions, thereby reducing communication overhead and improving spectral efficiency. It is further extended to sub-THz communications in D4.3, where end-to-end learning of a sub-THz transmitter and receiver is presented in D4.3 section 3.1. Furthermore, an AI/ML-based end-to-end learning approach for RIS-assisted communications is proposed in D4.3 section 3.1, which provides BER gains and low computational complexity. In terms of AI solutions for optimising individual blocks in the wireless transceivers, several enablers are presented in section 2.1 and section 4.1 in D4.2, and section 3.1 and section 4 in D4.3 including channel (de)coding, beamforming and beam management, channel charting, low complexity channel estimation including channel estimation in RIS-assisted systems, and radio resource allocation in cell-free massive MIMO.

Frameworks for data-centric hardware impairment mitigation and adaptivity to the wireless environment:

The AI/ML-based hardware mitigation solutions presented in section 2.1 in D4.2 and section 3.1 in D4.3 include PA-induced out-of-band emissions reduction and PA non-linearity compensation. The former utilised a CNN at the transmitter to learn a waveform that produces less out-of-band emissions under a nonlinear PA and is accurately detectable at the receiver, while the latter enables energy efficient PA operation by compensating in-band distortions due to PA nonlinearities, at the receiver. The adaptive and flexible AI/ML solutions include generalisable channel estimation of unknown channel models, channel estimation in RIS-assisted systems under mobility, and flexible radio resource allocation in cell-free massive MIMO which are presented in D4.3 section 4.

Concepts and mechanisms to support and manage collaborative AI components across the network, also leveraging Federated Learning (FL) and deployment of eXplainable AI (XAI) models:

Seamless and pervasive in-network AI operation is supported by AIaaS concept targeting stringent availability and reliability requirements even in highly mobile environments (D4.1/section 2.5, D4.2/section 3.1, D4.3/section5.1). An algorithm is provided in D4.2 (section 3.3) for optimal placement of AI workloads including the factors of communication load, power usage and trust levels, with numerical evaluation in D4.3 (section 5.1). Solutions for efficient resource usage and resource allocation for cooperative inferencing at the (extreme) edge is addressed in D4.2 (section 3.2 and 3.3) and D4.3 (section 5.2) by goal-oriented communications, joint communication and compute orchestration, and by integration of AI inference control mechanisms with communication functions in resilient distributed AI architectures., Numerical evaluations on distributed AI scenarios derived from Hexa-X use cases. Concepts to reduce the massive amount of data required for distributed and specifically federated learning are described in D4.2 (section 3.2 and 4.2) and D4.3 (section 5.3). A general scheme for FL of Decision Trees and Rule-Based Systems has been reported in D4.2 (section 5.2 and 5.3) along with preliminary results for the task of QoS/QoE prediction with XAI models. In D4.3, the accuracy and interpretability of XAI models for regression tasks (Fuzzy Regression Trees and Takagi-Sugeno-Kang Fuzzy Rule-based Systems) have been further investigated. Results for AI/ML privacy and security are provided in D4.2 and D4.3. Although federated learning is a privacy-preserving collaborative learning, the model updates sent to the aggregator server may still leak private information. Methods to improve the privacy and security of data in federated learning is provided in D4.2 (section 5.1) using differential privacy and multi-hop communication with blind signature. In D4.3 (section 6.2), this method is enhanced where dropping the model updates by malicious client is prevented. The effect of adversarial attacks on an in-network AI use case was evaluated, for which a defence mechanism is provided in D4.3 (section 6.1) to decrease adversarial attack success rate and increase the robustness of the model.

Development and assessment of intelligent orchestration methods, such as predictive orchestration:

Architectural design and workflow definition among the main principal software components in the edge and cloud domains, aiming at the data collection and training of a distributed AI agent for scaling the UPF instances. Details are available in D4.2 (section 2.2.2). In D4.3 (section 3.2.1), the quantifiable measurement of inferencing and training latency and inferencing accuracy are reported. An in-depth view of the requirements of predictive orchestration for future 6G mobile networks is provided in D4.2 (section 2.21) and D4.3 (section 3.2.2) including advanced monitoring system and forecasting algorithms classification. Furthermore, a mapping between the WP6 D6.2 M&O Architecture blocks and the proposed Predictive Orchestration blocks is detailed.

The document, as the final deliverable of WP4, also summarizes the quantified results of the following targets defined for the Connecting intelligence towards 6G project objective in detail in Chapter 8:

- *(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*
  The target has been addressed by multiple technical enablers in section 2.1 and 4.1 in D4.2 and section 3.1 and 4 in D4.3, by improving BER/BLER via either optimising the individual tasks such as channel (de)coding, channel estimation, beamforming, hardware impairment compensation, or by end-to-end learning of the wireless transceivers. To summarise some of the quantitative results in section 3.1, the AI-based end-to-end learning for sub-THz is shown a BLER gain of 1-2 dB, while the AI-powered receiver for PA non-linearity compensation achieved approximately 1 dB gain at 10% of BLER for simulations with higher order modulations such as 64QAM and 256 QAM. The end-to-end learning system for RIS-assisted communications on the other hand resulted in a BER < 0.001 for BPSK equivalent configuration and BER < 0.01 for QPSK equivalent configuration for SNR > 15dB when full-CSI is assumed. Among the channel estimation solutions proposed in section 4, the ML-based generalisable low complexity channel estimation solution has achieved an NMSE within 3 dB

range of MMSE of the optimal solution, enabling improved BER and throughput by reducing channel estimation error. Furthermore, the channel estimation in RIS-assisted systems with mobility achieved 5-10 dB reduction in NMSE while also reducing the resource overhead. (Further details in section 8.1.)

- *(T2) Increased AI algorithm robustness to system parameter volatility, lower complexity and efficient resource utilisation and rate gain as compared to classical approaches*
  Several technical enablers are proposed in D4.2 section 2.1 and 4.1, and D4.3 section 3.1 and 4, focusing on AI/ML approaches to provide spectral efficiency improvements and efficient resource utilisation while reducing processing complexity and improving flexibility. From the results presented in section 3.1, the AI-based end-to-end learning for sub-THz has achieved a throughput gain of 10-20%, with reduced transmission overhead due to pilotless detection. The AI-powered receiver for PA non-linearity compensation achieved approximately 20% throughput gain. This also enables communication link coverage extension in the presence of PA non-linearity and has shown to improve energy efficiency (70% improvement of PA power-added efficiency) at the transmitter side. The AI-based compressed sensing for beam selection in D-MIMO in section 3.1 also has shown rate gain via efficient resource utilisation, achieved by reducing the beam scanning time to 5-20% of the baseline exhaustive scan. The ML-based low complexity resource allocation mechanism for cell-free massive MIMO proposed in D4.2 section 4.1 is more than 50 times less computationally complex compared to the optimisation-based baseline, while achieving above 95% spectral efficiency performance. It has also shown the flexibility and generalisability of the data-driven resource allocation approach by achieving at least 90% spectral efficiency compared to baseline when using the same trained-DNN model for different system configurations in the considered problem scenario. (Further details in section 8.2.)

- *(T3) Resilient communication and compute network services for distributed AI applications in large scales (e.g., applications with >1000 collaborating AI components)*
  The target has been addressed by multiple technical enablers in D4.3 section 5 and section 3.2. In a resilient distributed sensor sharing application, where a massive number of AI-enhanced sensor components provide information for the inference task, over-the-air communication was significantly reduced (by 80% compared to baseline), which enables dense deployment of these devices. Considering that the device density requirements for the targeted demanding 6G use cases is 5-10 times higher than 5G requirements, this difference is largely offset by the proposed joint compute and communication solution, which remains efficient at ~1000 collaborating AI components. Deployment and operation of AI components at large scale is further addressed by the concept of AIaaS in high mobility environments, an algorithm for AI workload placement, and in the context of automated UPF scaling in low-latency network slices. (Further details in section 8.3.)

- *(T4) The accuracy of an XAI model within (<10%) of "black box" solutions (e.g., Deep Neural Networks – DNNs)*
  The target has been achieved both in the context of classification and regression tasks. In D4.2, Sec. 5.2.1, the adoption of XAI models (fuzzy and classical Decision Trees (DTs)) has been investigated for the task of QoE classification, leveraging a publicly available benchmark dataset. Such inherently interpretable models are slightly outperformed (in the range [2%, 5%] and in any case within 10%, in terms of f-score) by a Random Forest classifier which, as an ensemble model, is generally considered opaque. In D4.3, the trade-off between accuracy and interpretability of XAI models for regression tasks (Fuzzy Regression Trees and Takagi-Sugeno-Kang Fuzzy Rule-Based Systems) has been investigated. The proposed TSK FRBS with enhanced interpretability is shown to be comparable to (or even better than) a less interpretable state of art TSK-FRBS, based on an empirical comparison on several benchmark datasets. Specifically, the gain of XAI model vs less interpretable state-of-art solution is in the range [-4%, +37%]. (Further details in section 8.4.)

- *(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 delegation*
  The target has been addressed by technical enablers in D4.2 and D4.3, focusing on the problem of energy consumption at devices and infrastructure side, including communication and computing. Simulation based assessment of energy consumption reduction at UE side is achieved by partial offloading of inference workloads through DNN splitting. Depending on channel models (e.g., path loss exponent) up to 90% reduction was shown in case of highly available edge computing resources. Further gains can be obtained through the goal-oriented approach, where 12% energy reduction was shown in the simulated scenario without losing performance, and even higher gains were obtained by trading off inference accuracy. At the infrastructure side, optimised workload placement has shown to reduce the power consumption by 23% compared to baseline strategies, and in the scenario under high availability of computing resources in the mobile edge 2x to 10x energy consumption reduction was obtained thanks to cooperative inference. (Further details in section 8.5.)

- *(T6) Increased trustworthiness of AI through privacy and security enhancing technologies; using differential privacy to evaluate privacy versus communication utility trade-off.*
  The target has been achieved by solutions described in D4.2 section 5.1 and D4.3section 6. The study on Differentially Private (DP) Federated Learning focused on DP as a metric to measure the privacy of a dataset or a machine learning model in a radio network use cases by quantifying the degree to which an individual's data can be inferred from the output. DP increased privacy with a trade-off on accuracy. Using multi-hop communication and blind-signature, the privacy of collaborative machine learning is enhanced because the server can no longer map participants with their model updates. The accuracy of the model does not change after applying the proposed method, and the resulting overhead is considered reasonable. From security aspect, the machine learning model must be robust and resistant to adversarial attacks. Vulnerability of an AI-based radio network use case against adversarial attacks was shown and a defence method was proposed to improve model resilience against adversarial attacks, with 37% reduction of attack success rate in the simulated scenario. (Further details in section 8.6)

1.3 Structure of the document

The document is structured in the following way:

Chapter 2 provides a general overview and of the technical enablers in a structured way with references to the detailed studies.

Chapter 3 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 4 discusses AI/ML as an enabler for 6G network sustainability by complexity reduction in radio access network functions.

Chapter 5 presents 6G networks as an efficient AI platform, describing technical solutions to execute AI functions and AI applications efficiently in 6G networks, considering seamless operations, communication and compute resource efficiency in both learning and inference tasks.

Chapter 6 investigates privacy, security & trust in AI-enabled 6G, elaborating on the topics of security and privacy of AI in 6G, and explainable AI (XAI) with design and implementation of federated XAI (Fed-XAI) algorithms.

Chapter 7 gives an outline about demonstration activities carried out in WP4 (Fed-XAI demo).

Chapter 8 summarizes how Hexa-X targets are addressed by the technical areas and their KPIs.
The document concludes and provides overall guidelines in the Conclusion chapter.

2 AI-driven communication and compute solutions

Future 6G network functions and use cases will be intertwined with various forms of learning and intelligence in many aspects: air interface design, data management, optimality of compute and processing functions, network automation, service availability, while it will also call for the implementation of suitable mechanisms for trustworthy operation. WP4 addresses all the above domains with technical enablers, algorithms, joint solution proposals for communication and computation to help fulfilling Hexa-X Connecting Intelligence research challenges [HEX-D12]. The technical solutions described in this report, therefore, contribute to various network domains. The mapping of these areas to the architectural blocks identified in [HEX-D13] are illustrated in Figure 2-1: and their breakdown to technical solutions in Figure 2-2.

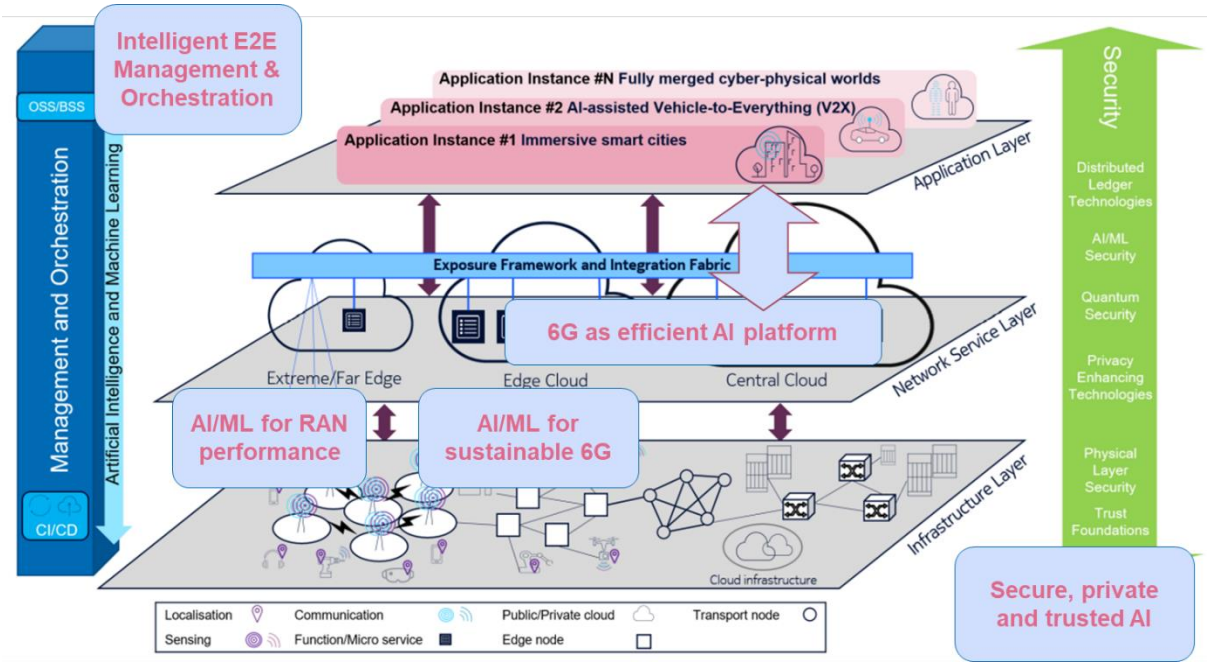


Figure 2-1: Contribution of WP4 technical enablers in network architectural blocks identified in D1.3.

AI/ML solutions for RAN performance enhancements and sustainable 6G are connected to both Infrastructure Layer and Network Service Layer, as the supported functions are targeting radio processing algorithms to improve communication performance, introduce energy efficient solutions, and supporting novel RAN concepts like Distributed MIMO (D-MIMO) and Reconfigurable Intelligent Surfaces (RIS). These AI/ML solutions can heavily rely on network-specific infrastructure elements or more generic AI accelerator hardware.

RAN performance improvements are addressed by multiple AI techniques in this document. In the transceiver chain, joint training of transmitter/encoder and receiver/decoder frameworks and algorithms lead to better spectral efficiency (in general or compared to similar complexity solutions), reduced level of pilot usage, overcome suboptimal solutions of modular designs, increased robustness against hardware impairments. Power amplifier non-linearities are also addressed by receiver-only solutions. Further improvements of beam selection overhead and latency are shown in massive D-MIMO deployments. The related technical enablers are the following:

- ML-based end to end learning of RIS-assisted communication systems (Section 3.1.1),
- NN/ML aided channel (de)coding for constrained devices (Section 3.1.2),
- AI based compressed sensing for beam selection in D-MIMO (Section 3.1.3),
- AI empowered receiver for PA non-linearity (Section 3.1.4),
- AI-Based Enhancements for Sub-THz (Section 3.1.5),
- Channel charting based beamforming (Section 3.1.6).

While RAN performance improvements are obvious goals for AI algorithms, the complexity and energy efficiency aspects must also be considered. This is a recurring concern as AI/ML based solutions can often yield large neural networks which have to be executed in real time. In some cases, the increased complexity from AI is justified by the benefits and gains seen in spectral efficiency or latency. But even in these cases, the problems may be solved by well-designed AI architectures, where reduced complexity is achieved by applying expert knowledge instead of a large set of fully connected dense layers or other general purpose NN layers, as demonstrated by the results below. Another area for AI/ML to improve energy efficiency is addressing large optimization problems that are practically intractable by conventional methods, like D-MIMO power allocation. In these cases, the general function approximation property of neural networks can be utilized. Further details are described in the following solutions:

- Low complexity radio resource allocation in cell-free massive MIMO (Section 4.1),
- ML-based channel estimation for RIS-assisted systems with mobility (Section 4.2),
- Generalizable low complexity channel estimation using neural networks (Section 4.3),
- Deep unfolding for efficient channel estimation (Section 4.4),
- Hybrid model for channel charting (Section 4.5).

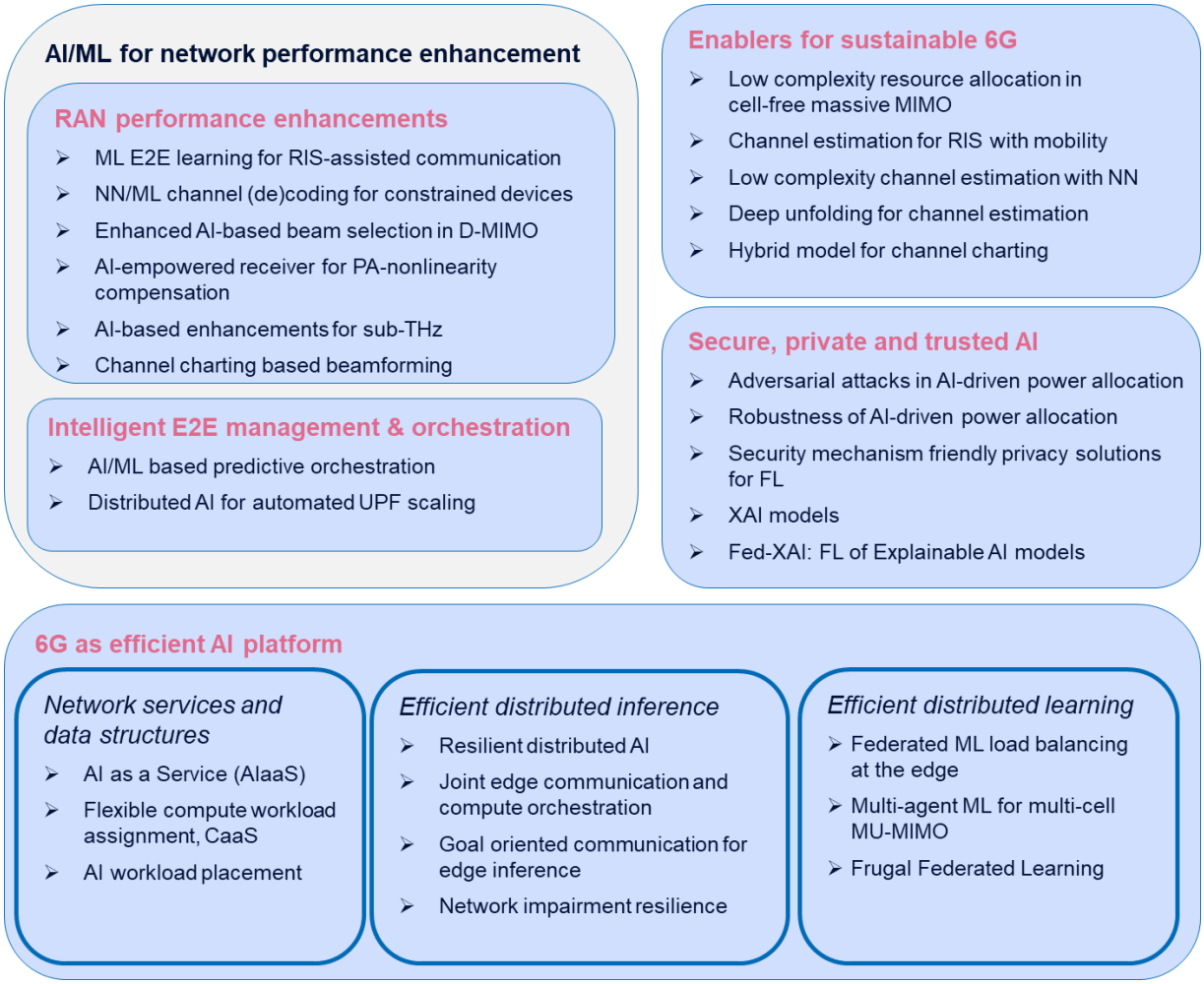


Figure 2-2: Summary of the AI-driven solutions of the document by their domain of application.

In the area of Intelligent E2E Management and Orchestration AI/ML solutions are presented as a way forward to enhancing the network communication in 6G systems, including predictive solutions regarding improvements on orchestration aspects, as well as a decentralised solution to improve the scaling of architectural components through a case study of efficient provisioning of User Plane Function (UPF), as described in the following sections:

- Distributed AI for automated UPF scaling in low-latency network slices (Section 3.2.1),
- AI/ML based predictive orchestration (Section 3.2.2).

The main goal of the enablers included in the block of "6G network as an efficient AI platform" is to enable and enhance the global operation of AI services over the network, considering computing capabilities as a native part of future networks. These functionalities primarily reside in network service layer, with strong interface and exposure requirements towards in-network AI functions and the application layer. The relevant technical enablers address multiple areas. Network services are proposed with open interfaces and data structures derived by the requirements of AI applications, fulfilling stringent availability and reliability requirements even in highly mobile environments. The specific requirements of AI applications in the form of data availability or trust levels are also considered during workload placement. See following sections for details:

- AIaaS - seamless exploitation of network knowledge (Section 5.1.1),
- Flexible compute workload assignment, CaaS (Section 5.1.2),
- AI workload placement for energy, knowledge sharing and trust optimization (Section 5.1.3).

Many of the AI-enabled 6G use cases require real-time critical functionalities, which would normally pose stringent requirements on the communication network bandwidth, latencies, device density and traffic volumes over wireless connections. However, the application level KPIs can also be achieved by joint design of application-level metrics and controls with supporting communication functions in 6G. This can be done by exploiting noise and loss tolerance of neural networks and adapt the operation to the real-time environment knowledge measured by the network. Concepts and solutions are described in the following enablers:

- Scalable and resilient deployment of distributed AI (Section 5.2.1),
- Joint communication and computation orchestration for edge inference (Section 5.2.2),
- Goal-oriented communication approach for edge inference (Section 5.2.3),
- Network impairment resilience of autonomous agents (Section 5.2.4).

Similar considerations apply to the model training tasks, where the major challenges are the huge data volume to be managed and shared between the AI agents. Data reduction techniques are provided for Federated Learning in general, and also for multi-agent ML by combining centralized training with decentralized execution, as studied in a MIMO network application:

- Centralized training and decentralized Execution for multi-cell MU-MIMO (Section 5.3.1),
- Federated ML model load balancing at the edge (Section 5.3.2),
- Frugal Federated Learning (Section 5.3.3).

This report also covers AI trustworthiness in three aspects: security, privacy and explainability, with the technical enablers below. The main goal of these enablers is to design and develop AI systems that are transparent, reliable, and fair and also ensure data privacy and security. Since AI systems are usually used to process and analyse data such as personal information, a security breach could result in significant harm. It is also important for AI systems to be transparent and explainable in decision-making process and be understandable and trustable by humans. Therefore, developing and deploying trusted AI systems is essential. The details of the following technical enablers and also the demo related to Federated Explainable AI can be found in Chapter 6 and 7 of the report:

- Adversarial evasion attacks in AI-driven power allocation (Section 6.1.1),
- Defence mechanisms to increase robustness of AI-driven power allocation (Section 6.1.2),
- Security mechanism friendly privacy solutions for federated learning (Section 6.2.1),
- XAI models: Fuzzy regression trees and TSK fuzzy rule based systems (Section 6.3.1),
- Fed-XAI: Federated Learning of Explainable AI models (Section 6.3.2).

3 Network performance enhancements using AI/ML in 6G

In this chapter the emphasis is on technical enablers for AI/ML-based solutions to enhance 6G network performance. The chapter contains two sub-chapters, introducing eight technical enablers. The first sub-chapter focus on radio access and link-level enhancements, comparing proposed solutions to classical design methods. The second sub-chapter presents AI/ML-based solutions for network management and orchestration.

3.1 Radio access network performance improvements over classical design methods

This sub-chapter presents six technical enablers for radio access and link-level enhancements. The first enabler targets system performance improvements in a reconfigurable intelligent surfaces (RIS)-assisted communication system by jointly optimizing the transmitter, receiver and RIS. The second enabler presents ML-based channel decoding for constrained devices. AI-based compressed sensing for beam selection, described in terms of a low latency solution, is the third enabler. This will be useful in high mobility beam tracking scenarios. The fourth enabler discusses a solution for compensation at the receiver for PA non-linearity at the transmitter. This is followed by the fifth enabler presenting learned constellations at the transmitter and AI-based receiver as part of an AI-native air interface, extending earlier work to sub-THz frequencies. The last enabler in the sub-chapter is channel charting based beam forming where the relative location of users is utilized. Performance gains are quantified in terms of bit error rate (BER) / block error rate (BLER), bit rate or spectral efficiency improvements, as well as computational complexity reduction in most of the above.

3.1.1 ML-based end-to-end learning of RIS-assisted communication systems

RIS is an emerging wireless technology for controlling the radio signals between a transmitter and a receiver using a planar surface that consists of a large number of passive reflecting elements. They can be used to improve the system performance by changing the phases of the incoming signals especially when the transmitter to receiver direct path is blocked. In an RIS-assisted system, there are additional signal processing tasks in the RIS such as the channel estimation and reflection coefficient optimisation (phase shift optimisation of the RIS reflection elements to maximise the transmission rate) in addition to the signal processing blocks in the transmitter and receiver in a conventional system. Recent works in RIS related research have proposed ML-based approaches for such different signal processing tasks in RIS-assisted [GSR+21, GZC+20, KKH+19]. On the other hand, several works have proposed end-to-end learning of a communication system, where the transmitter and receiver are jointly optimised using the autoencoder concept [FCD+18, WWL+19]. Such an ML-based end-to-end communication system is also much simpler and straightforward than the complex algorithms involved in each sub-task of the signal processing chain. Motivated by this fact, in this work we investigate the potential of ML-based end-to-end learning in an RIS-assisted communication system. We exploit the learning capability of ML-based autoencoders for joint optimisation of the transmitter, RIS, and the receiver in an RIS-assisted communication system to improve its bit error rate (BER) performance.

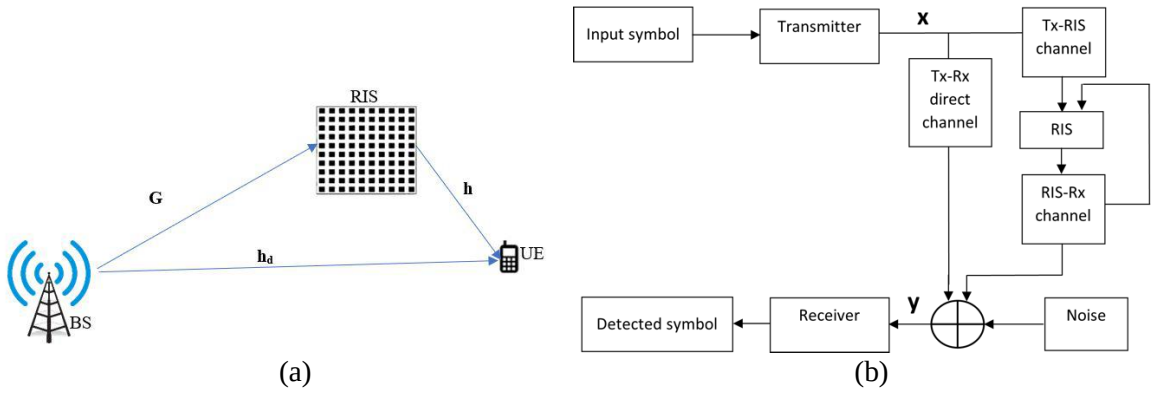


Figure 3-1: RIS-assisted communication system model and system architecture.

Downlink transmission of a single-user RIS-assisted system is considered, where an RIS is placed in between the base station (BS) and the user equipment (UE). Single-antenna user is assumed and the RIS has N elements. The downlink transmission happens from the BS to the UE through the direct channel and cascade channel as shown in Figure 3-1/a. The system architecture illustrated in Figure 3-1/b is implemented as a ML-based end-to-end system: the transmitter and receiver are implemented as two convolutional neural networks (CNNs) and the channel block (consisting of the RIS block and noise layer) is implemented as a fully connected deep neural network (DNN) block. The RIS block is pre-trained to predict the optimal phase shifts to maximize the SNR of the transmitter-to-RIS-to-receiver cascade link using either perfect or imperfect channel state information (CSI) of both direct and cascade channels. With perfect CSI, the RIS block is trained with random channels (the direct links from the transmitter to the receiver and the transmitter-to-RIS-to-receiver links), which are assumed to be known to the RIS. For the imperfect CSI scenario, pilot transmission with a predefined reflection coefficient matrix for the RIS is considered and least square channel estimations are obtained and feedbacked to the RIS to find the optimal reflection coefficients for the data transmission. This pre-trained RIS block is then used in the end-to-end model. The full transmitter-channel-receiver model is then trained in an end-to-end manner to minimise the symbol detection error. We denote this CNN-based end-to-end as a CNN-autoencoder (CNN-AE).

Figure 3-2 shows the comparison of the BER performance of the CNN-AE for the RIS-assisted system when the number of RIS elements varies for BPSK modulation scheme. The results are also compared against the theoretical BER performance of an equivalent system. The theoretical BER values of the RIS-assisted system is calculated using equations in [BDD+19] and only the transmitter-to-RIS-to-receiver channel is considered there. Furthermore, for the theoretical BER calculations, it is assumed that the channel values are known and the optimized phase shift values that maximize the SNR are obtained through equations. The CNN-AE BER performance on the other hand are obtained for both instances when the transmitter-to-receiver direct channel is included and excluded. Figure 3-2 shows that when the number of elements in the RIS are increased the BER of the CNN-AE system has decreased. The CNN-AE system has a lower BER compared to the theoretical in the high SNR range, and the performance improves when the number of RIS elements is increasing. In these simulations we have trained the CNN-AE for a given SNR in each RIS setting and evaluated the BER across the considered SNR range using the trained model. While this shows the flexibility of the CNN-AE, it should be noted that the training SNR considerably effect the performance of the CNN-AE and therefore needs to be carefully selected to achieve the expected performance.

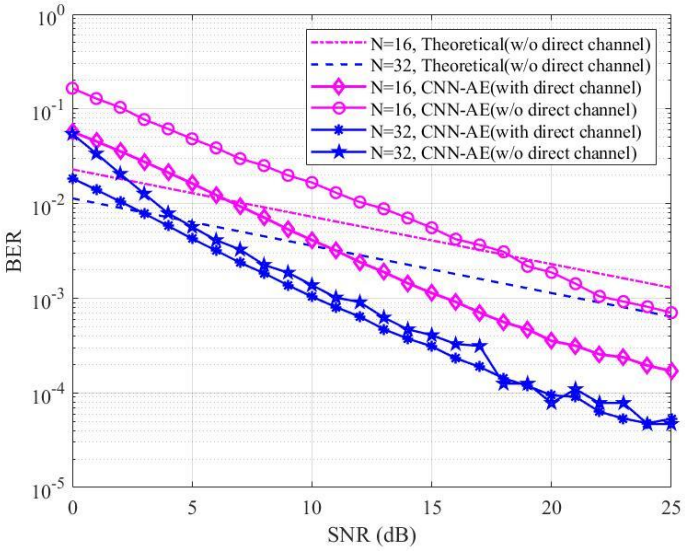


Figure 3-2: BER performance (BPSK) of the proposed CNN-AE for RIS-assisted communication vs theoretical [BDD+19] for different RIS sizes.

Furthermore, the BER performance of the CNN-AE when the RIS-assisted system is configured in different communication rates which is equivalent to applying different modulation schemes in a conventional communication system is shown in Figure 3-3. The number of RIS elements is selected to be 16 for these simulations. Therefore, this shows the compatibility of the proposed system for different modulation schemes.

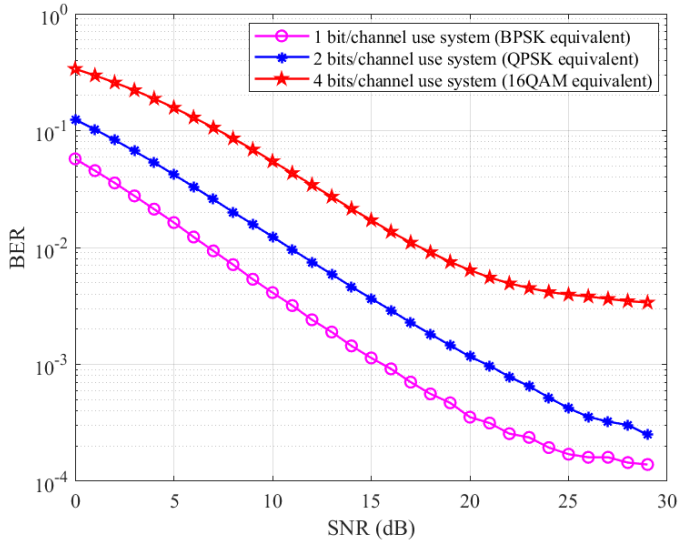


Figure 3-3: BER performance of the CNN-AE for RIS-assisted system for higher communication rates.

Finally, the comparison of BER performance for perfect CSI and imperfect CSI for QPSK modulation scheme when number of elements are 16 is shown in Figure 3-4. The pilot-aided channel estimation mechanism explained earlier is implemented in the imperfect CSI scenario. It shows that the BER performance is reduced with imperfect CSI compared to perfect CSI.

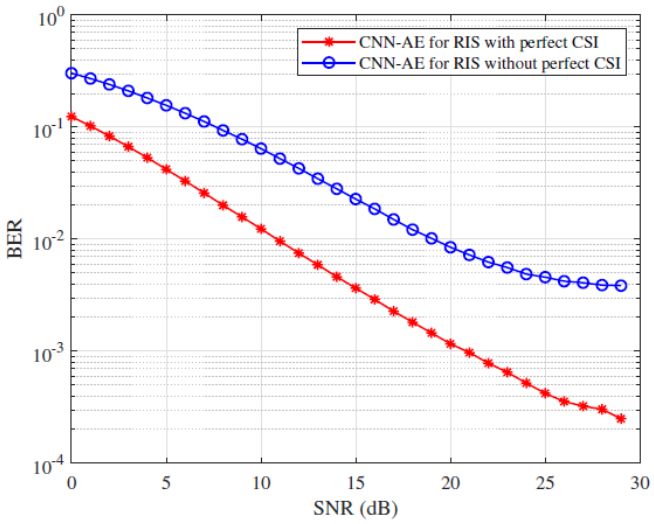


Figure 3-4: BER performance of the CNN-AE system with and without perfect CSI.

The numerical results show the potential of a CNN-AE system for an RIS-assisted communication system which is capable of learning transmit signals and the optimal reflection coefficients for the RIS in an end-to-end manner to minimize the BER. Such a system would be relevant and beneficial in “Interactive and cooperative mobile robots" use case where new RIS-assisted communication architectures could be used to manage a cluster of drones over a 6G network for improved end-to-end performance [HEX-D41]. The relevant target for this enabler is T1 - Improved end-to-end BER/BLER, which we have used to evaluate the performance of the proposed CNN-AE system.

3.1.2 NN/ML aided channel (de)coding for constrained devices

Efficient Forward Error Correction (FEC) schemes are a key enabler for Ultra-Reliable Low Latency Communications (URLLC) and/or Internet of Things (IoT) scenarios. Such use-cases usually imply the use of short packets (datagrams of a tens bits to a few hundred bits) and low-complexity coding/decoding procedures, in line with the conflicting constraints of latency, energy consumption, hardware cost, available computational power, etc.

Existing FEC codes, such as Bose Chaudhuri and Hocquenghem (BCH), Tail-Biting Convolutional Code (TB-CC), Turbo, Polar or Low-Density Parity Check (LDPC) codes and their respective decoders strike different trade-offs which, further leading to different FEC choices for different use-cases. As an illustration of this, the 3GPP standardized the use of Polar codes for the control channels and LDPC for the data channels of the 5G New Radio (5G-NR) interface for enhanced Mobile Broad-Band (eMBB) applications [3GPP16]. The rationale is that LDPC codes offer near capacity performance and efficient decoding [FMI99], [CDE05], in the large code block regime, but their performance is subpar for shorter codes. Conversely, polar codes provide excellent performance at a reasonable decoding complexity for shorter block lengths and their complexity becomes a limiting factor for larger code lengths.

In this current setup, devices are required to support two different FEC schemes along with their respective decoders, adding to the complexity and cost of the devices.

In this work, we investigate the use of Linear Block Codes (LBC) for short block lengths, typically in the range of few tens up to hundreds of bits, using Belief Propagation (BP) decoders, as used for LDPC codes. The goal is to provide a unified decoder architecture that would encompass both extrema in block regimes, and therefore suit both data and control planes requirements. Such an architecture would be of great interest for next generation communication networks such as 6th Generation (6G) networks.

This work relies on machine learning to discover efficient short block codes under BP decoding and to optimize the decoder under complexity constraints, i.e., fixed number of iterations.

Proposed approach

The overall model used in this work is an auto-encoder, encompassing both the encoding of messages and the decoding of channel codes, as illustrated in Figure 3-5:.

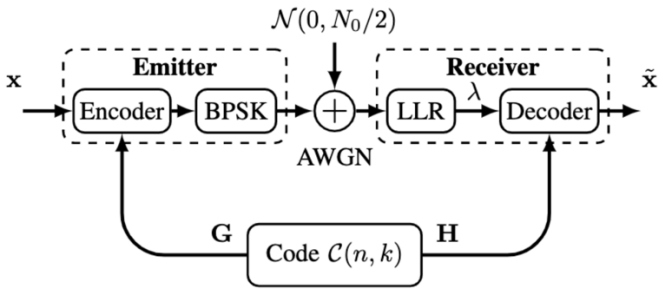


Figure 3-5: Model architecture.

This model consists of two main building blocks, namely the encoder and decoder, and shared parameters: the Code $\mathcal{C}(n,k)$. The encoder, as the name suggests, carries out the encoding of messages and modulation, while the decoder performs the decoding of channel codes, more specifically of the evaluated Log-Likelihood Ratios (LLRs). The training process evaluates the output of the decoder with respect to the input of the encoder using a loss function $\ell(\tilde{x}, x)$, here the Binary Cross Entropy (BCE). The Code parameters, i.e., the parity equations, are optimized using a Stochastic Gradient Descent (SGD) variant.

Encoder:
The encoder side is a differentiable implementation of a linear block coder, where input messages are multiplied by a generator matrix G and further modulated using a Binary Phase Shift Keying (BPSK) constellation. More specifically, the proposed encoder block describes linear combination in $\mathbb{F}_2$ by representing the XOR function as a differentiable product of bipolar symbols and follows the logic described in Figure 3-6:

- Binary words of size k are converted into bipolar form (i.e. {0;1} are mapped to {+1; −1}).
- Inputs are broadcasted and multiplied by trainable external binary weights, i.e., the code's generator matrix G, thus selecting the bits participating in each of the n encoding equations.
- For the variables that were not selected the neutral element of the product (+1) is added.
- Finally, a product reduction is performed for each of the n encoding equations.

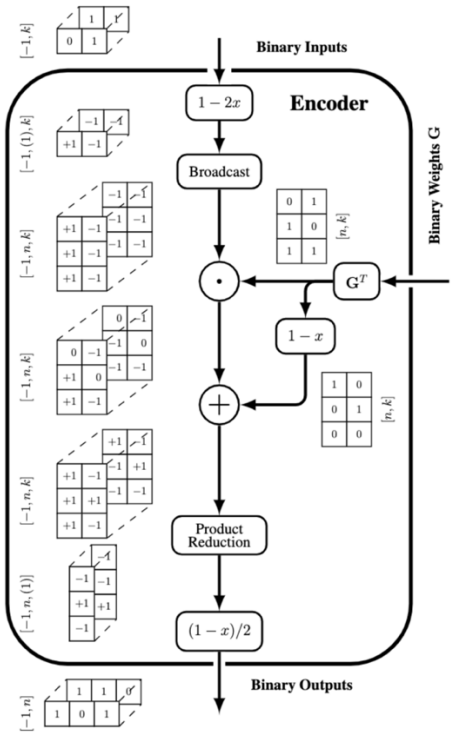


Figure 3-6: Encoder model - execution graph.

Channel:
A simulated 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 w.r.t the encoder parameters.

Decoder:
The decoder part of the auto-encoder supports mainly 3 functions:

1. Demodulation: Log-Likelihood Ratios (LLRs) of the received noisy symbols are computed according to the type of modulation used (BPSK) and current noise level. A trainable normalization is used to find the best LLR range to be provided to the forthcoming decoder.
2. Decoding: The LLRs are provided to a Neural Network Belief Propagation decoder which is described below. This trainable decoder has parametric weights on the received LLRs to improve the decoding performance when compared to standard BP decoders.
3. The results of the different decoding iterations are recombined to form the final decoding result using a trainable weighted sum and thus be able to only select the iterations outputs of the decoder that provide the more reliable results.

Code, shared weights:
The parameters of parity-check and generator matrices (H and G) of the code are regrouped in a place-holder model. To reduce training time, a systematic form is employed to allow for a direct conversion between G and H. Using this approach, we ensure that encoder and decoder are always matched). To ensure the differentiability of the architecture with respect to the code model parameters, G and H are derived from real scalar weights W using a differentiable step function (DSF) defined as:

$$f(x) = step(x)$$

$$\frac{df(x)}{dx} = \sigma(x)\sigma(1-x)$$

The Neural Network Belief Propagation decoder

Arguably the most significant contribution of this work, the NN-NBP decoder is based on the Tanner Graph (TG) representation of codes and the Sum-Product Algorithm (SPA). In essence (a detailed explanation of those notions is provided in [LDL+22]), the SPA is an iterative decoding algorithm, based on message passing between variables nodes and parity check nodes of a Tanner Graph.

The NN-NBP implements the SPA as a Recurrent Neural Network Cell (RNN-cell) and is illustrated in Figure 3-7:.

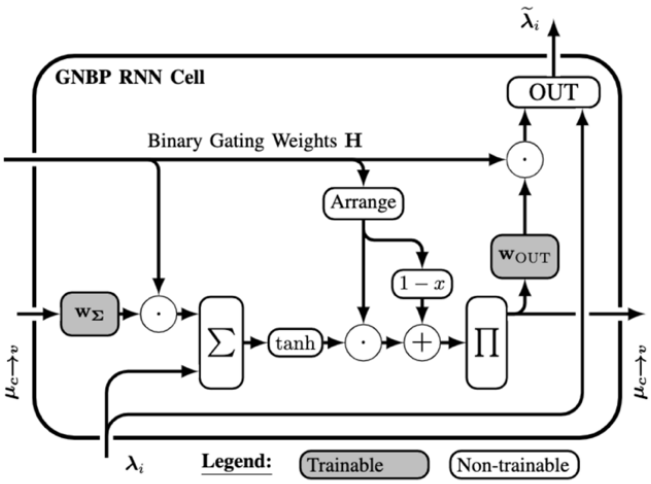


Figure 3-7: RNN-cell executing the Sum-Product Algorithm.

At each decoding iteration, the cell receives the normalized LLRs λ_i. Input LLRs are summed with weighted messages from parity check nodes to variable nodes $\mu_{c\to v}$ coming from previous iterations, effectively computing update messages from variable nodes to check nodes $\mu_{v\to c}$, as defined in the "sum" part of the sum-product algorithm:

$$\mu_{v_i\to c_j} = \lambda_i + \sum_{l\neq j} w_{\Sigma\,l,j}\ \ \mu_{c_l\to v_i}\ .$$

The cell's graph execution then computes updated messages from check nodes to variable nodes, i.e. the "product" part:

$$\mu_{c_j\to v_i} = 2arctanh\left(\prod_{l\neq i} tanh\left(\frac{1}{2}\mu_{c_l\to c_j}\right)\right).$$

Finally, the a posteriori LLRs are computed:

$$\widetilde{\lambda_l} = \lambda_i + \sum_l \mu_{c_l\to v_i}\ .$$

The overall architecture builds up around this RNN-cell as shown in Figure 3-8:.

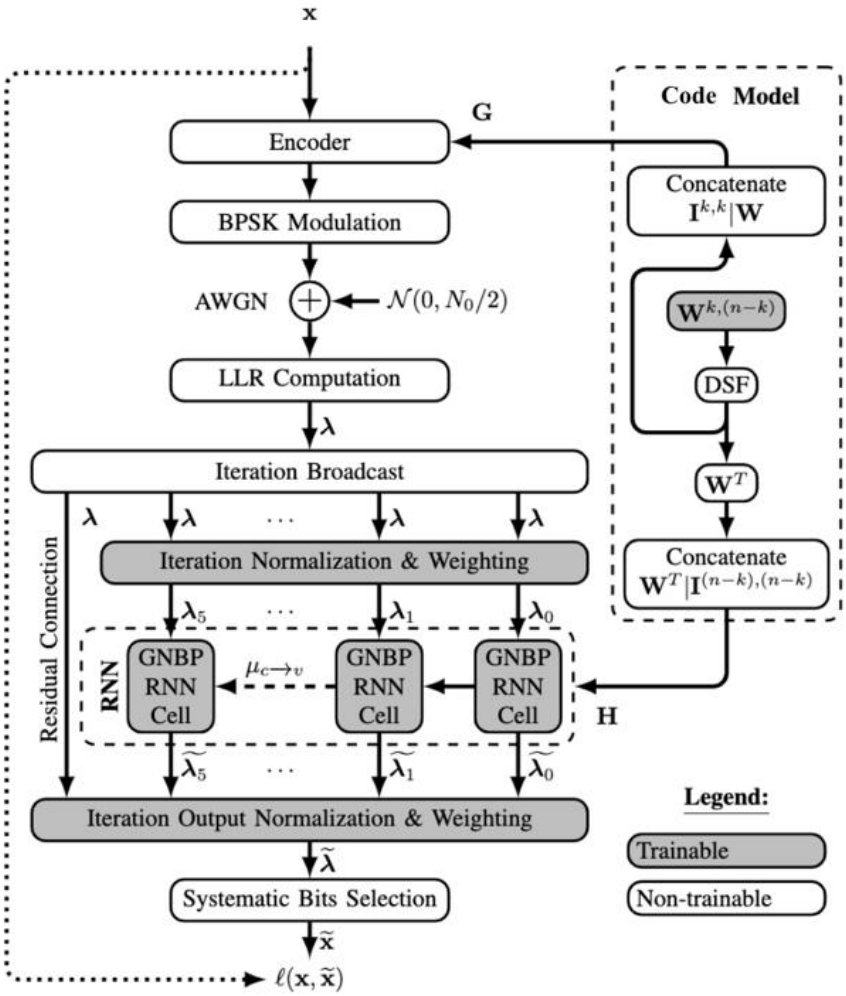


Figure 3-8: Complete model architecture.

Training procedure and datasets

As mentioned above, the parameters of the Code are trained using an SGD based on the Binary Cross Entropy loss function. All datasets used in the proposal are synthesized.

In a naive approach, learning a code of size (n, k) would require a training dataset including all the 2^k words. This number grows exponentially with the code size and can become prohibitively large. In the case of LBC and under symmetric assumption on the channel and the decoder (implying in the case of NN based decoders the use of symmetric activations, absence of biased units, etc.), it is common to train or evaluate a decoder only with the zero code-word, while guaranteeing performance on the complete code [RU07]. Nevertheless, the training of the encoder is also considered in this work, thus requiring the use of different words. The proposed encoder ensures that all code-words are linear combinations of the basis of the vector subspace of the code. Hence, the model can be trained using only the basis vectors of the words thus reducing the training data-set size from an exhaustive dataset of 2^k words to only k words.

Results

An extensive performance analysis of the codes learned using this proposal is available in [LDL+22]. More specifically, we provide results and structure interpretation for very short block lengths codes,

e.g. (8,4); we investigate the repeatability and convergence of code training instances on short block lengths codes, e.g. $(n = 31, k = 16)$ and we provide extensive results for codes of length of interest$(n = 64, k = 18,36,45)$and $(n = 128, k = 64)$.

Here, we provide key results that compares learned codes to state-of-the-art LDPCs and Polar Codes.

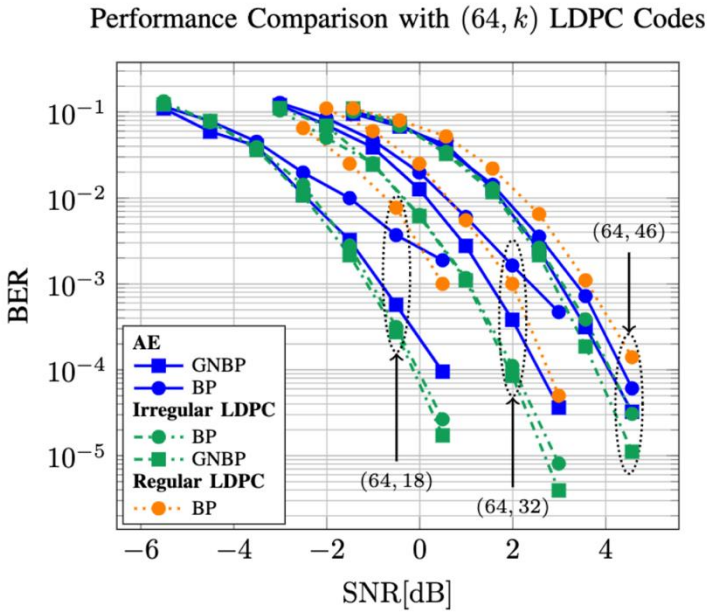


Figure 3-9: Performance comparison, n=64.

Figure 3-9: compares learned codes of size $n = 64$ to both regular and irregular state-of-the-art LDPC codes designed for this block regime using the Progressive Edge Growth (PEG) algorithm, decoded with both a standard BP decoder and an improved BP decoder using our learning proposal (fixed code weights). Despite learned codes being in standard form, their performance is significantly better than that of regular LDPC codes and in the same ballpark as irregular LDPC codes (no more than 0.5dB difference).

Figure 3-10: compares the performance of learned codes of size $n = 128$ with state-of-the-art LDPCs, polar codes and Tail-Biting convolutional codes. The take-away from this graph is that learned codes perform close to state-of-the art LDPC codes designed for this code size (again, no more than 0.5dB difference) for the same number of decoding iterations, while retaining the standard form property, and therefore allowing to skip the decoding altogether for high SNR conditions.

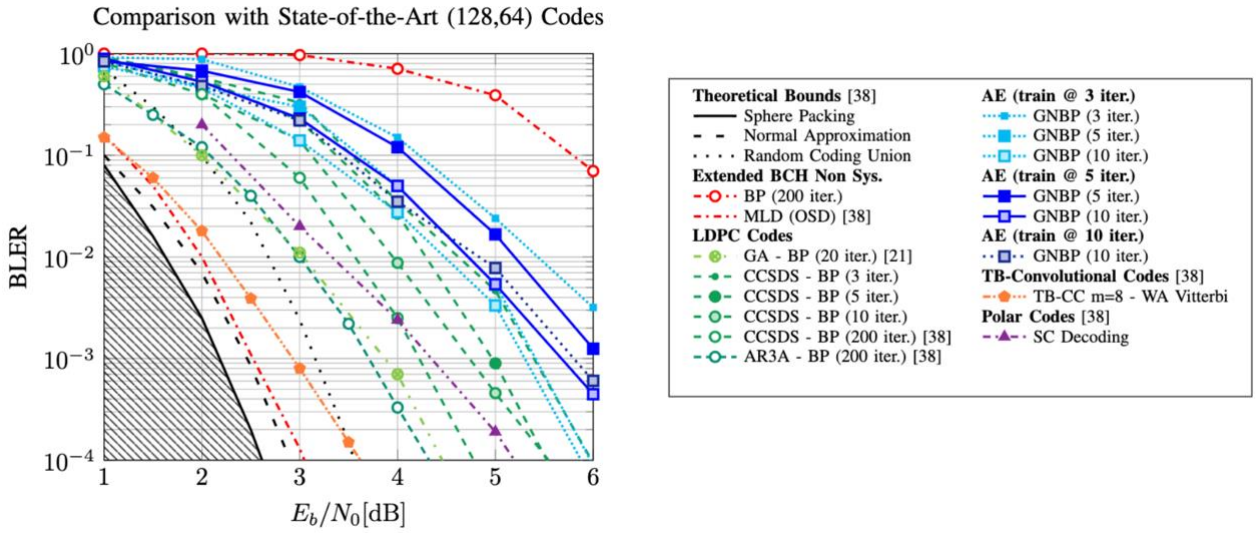


Figure 3-10: Performance comparison, n=128, k=64.

Obviously, there are codes that perform significantly better than proposed codes, but their complexity is up to 40 times higher (200 to 5 iterations).

These results highlight the value of machine learning for code discovery for scenarios that need to satisfy composite requirements, here a trade-off between decoding complexity and BLER. The proposed learning approach has indeed allowed to learn codes with low decoding complexity that approach state-of-the-art performances while guaranteeing the explicability of the models and the compatibility with legacy decoders, i.e. the generating and parity matrices can be extracted and reused in a standard BP decoder.

In addition, the learned codes are particularly suitable for small datagrams and allow direct access to the messages (the codes are made up of the message to be encoded to which redundancy bits are postfixed) which avoids a costly decoding prior to any exploitation of the data, e.g., CRC check. Thus, these results contribute to the achievement of the objectives of the use case "Immersive smart cities and integrated micro-networks for smart cities", in particular the related applications requiring the transport of small datagrams under constraints of energy efficiency and use of communication resources and contribute to the achievement of the target KPIs of WP4 concerning the performances in terms of BLER and energy efficiency.

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

Beamforming is a necessary technique to provide reliable coverage at higher frequencies in the mmWave range. Using narrow beams has the advantage of higher beamforming gains, capacity and coverage, but it also poses a significant challenge to the beam selection process in the form of increased overhead. High beam selection delays have especially high impact on connection reliability in applications with high user mobility or in deployments with many blockages. Several of the envisioned 6G use cases, like interactive and cooperating mobile robots or high-speed vehicular communication will require reliable connections even in the mmWave range. The problem can become even more severe in Distributed massive MIMO (D-MIMO) settings, where there is a large number of APs (typically more than UEs), each with multiple transmit antennas and possible beam directions.

Compressed Sensing (CS) based beam identification is an efficient way of reducing the delay of best beam selection by providing algorithms using significantly fewer measurements than the number of beams. This is done by exploiting the channel sparsity in the angular domain ([CSD+17]). We consider a MIMO system with *n* transmit antennas and one receive antenna at the UE. The reference signal for beam selection is transmitted over *m* time slots. Assuming a narrowband propagation model, when sending a reference signal $\boldsymbol{p}$, the beamforming system is described as

$$\boldsymbol{y} = \boldsymbol{DBhp} + \boldsymbol{n},$$

where $\boldsymbol{y}^{(m\times 1)}$ is the received signal vector, $\boldsymbol{h}^{(n\times 1)}$ is the channel vector for *n* transmit antennas, $\boldsymbol{B}^{(n\times n)}$ is a DFT-based codebook forming *n* equally spaced beams, $\boldsymbol{D}^{(\boldsymbol{m}\times\boldsymbol{n})}$ is a matrix for linear combination of beam weights (called dictionary or sensing matrix), and $\boldsymbol{n}$ is additive Gaussian noise. In this system the channel vector in the angular domain $\boldsymbol{x} = \boldsymbol{Bh}$ is a sparse vector due to multipath sparsity, for which the system equation is transformed into a standard sparse decoding problem of $\boldsymbol{Y} = \boldsymbol{Dx}+\boldsymbol{n}$, which can be solved even if $m \ll n$.

Although these CS-based techniques present reduced overhead with certain random dictionaries, local environment statistics can also be utilized to further optimize the dictionary. The distribution of beam patterns received by UEs is not uniform and statistics on typical joint beam patterns can be exploited. Initial investigations indicated that the number of required measurements can indeed be further reduced if we apply AI/ML techniques to learn the dictionary for a given deployment [HEX-D42].

This dictionary training can be performed with the help of an autoencoder architecture (Figure 3-11:), where input is a representative set of beam channel vectors *X* sampled from potential UE locations. The set of channel vector samples may be collected offline with dedicated measurements (drive test) or online with regular updates from the mobile devices. The encoder is a dense layer (matrix multiplication) with the dictionary elements set as weights ($\boldsymbol{D}$), which results in the measurements Y_{meas}. The decoder block must implement a sparse decoder algorithm (using the encoder dictionary $\boldsymbol{D}$) and it must also be differentiable so that gradient descent can be applied to optimize $\boldsymbol{D}$. Several neural sparse decoder architectures were investigated, including the Learned Iterative Shrinkage and Thresholding Algorithm family of LISTA [GL10], LISTA-CP [CLW+18] and Ada-LISTA [AGE21]. All the above algorithms are based on the Iterative Shrinkage and Thresholding Algorithm (ISTA), but their models have a significant difference in the number of trainable parameters. While LISTA scales with the square of the beam number (*n*), LISTA-CP scales with the size of the dictionary $n \times m$, whereas Ada-LISTA can be trained to be independent of the dictionary and the number of trainable variables scales with $m \times m$.

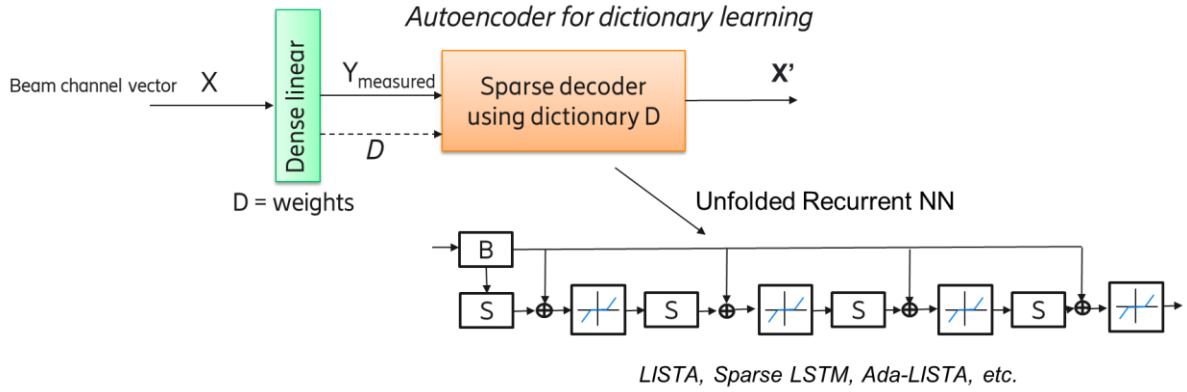


Figure 3-11: Autoencoder neural network for optimizing dictionary and sparse decoding.

The main KPI to compare the performance of different solutions is the best beam match ratio within *1dB* (decoded beam is accepted as best beam if its gain with path loss is within *1dB* compared to the best). Note that even if the dictionary and the neural sparse decoder are trained jointly, *D* can still be used with any other traditional sparse decoder algorithms. Although using a learned dictionary itself shows the expected benefits, significant additional gain can be realized by using the decoder model trained with the given dictionary [HEX-D42]. The models with higher complexity and more parameters

are expected to be better in this additional optimization, as it can be seen in Figure 3-12:. This analysis has been performed based on the open DeepMIMO dataset [ALK19], 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.

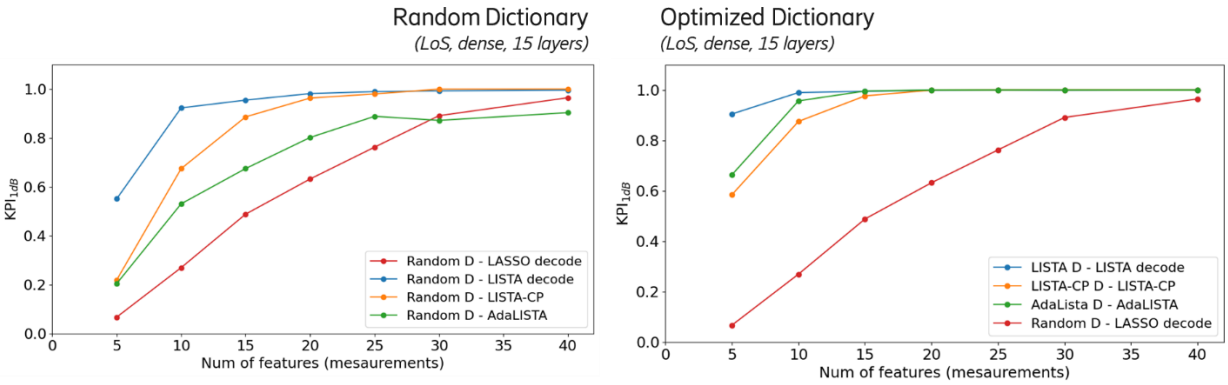


Figure 3-12: Comparison of performance of neural sparse decoders with random and optimized dictionary.

The impact of the training data quality has also been investigated. Since collecting perfect training dataset is hardly possible, it is important to see if the training is robust enough for less frequent sampling. Figure 3-13: shows that more fine-grained sampling leads to better model quality, even a sampling distance of *1m* results in close to optimal performance, both in line of sight (LoS) and non-LoS scenarios. This observation suggests that the model with sparse neural decoder generalizes well even for a coarser training data.

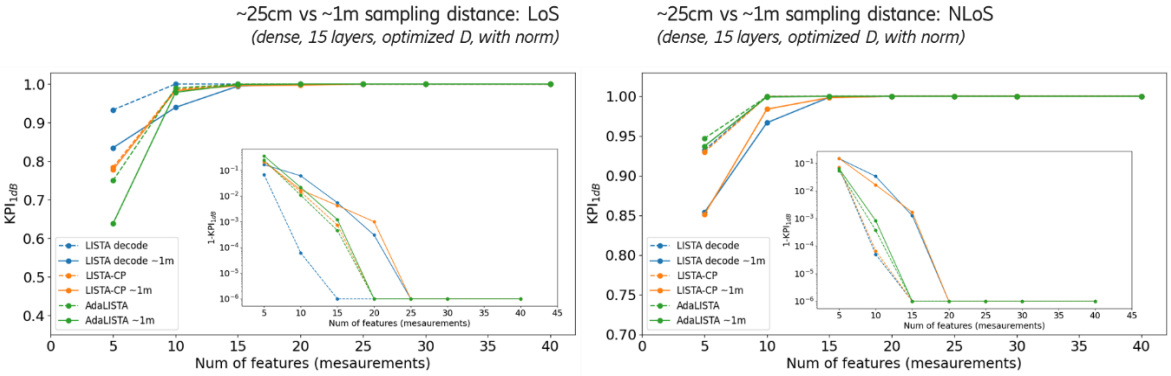


Figure 3-13: The training performs well in both LoS and more complex NLoS scenarios. The model also generalizes well for small training data.

Decreasing the beam scanning time is especially beneficial in high mobility beam tracking situations when fast beam changes are required to prevent losing connection for latency critical communication. The improvements demonstrated by this technical enabler may reduce the scanning time by an order of magnitude, leading to both lower ratio of connection drops as well as faster recovery times. This will contribute to the extreme reliability KPIs required by use cases like “Interacting and cooperative mobile robots” or Telepresence and Massive twinning use case families.

3.1.4 AI empowered receiver for PA non-linearity compensation

Introduction: Power amplifier (PA) non-linearity causes throughput degradation in wireless communication systems. The classical methods compensate PA non-linearity at the transmitter-side, e.g. by applying power back-off or performing digital-pre-distortion (DPD). However, applying PA power back-off leads to lower energy efficiency, and lower output power, and hence reduces coverage;

and performing DPD results in higher complexity of the transmitters. We propose an alternative approach to compensate the impact of PA non-linearity at the receiver side using a neural network-based demapper [FHS23] in operating regimes in which out of band emission requirements are fulfilled. We evaluated the performance of the proposed method and compare it against legacy receiver using KPIs including un-coded bit error rate (BER), block error rate (BLER), throughput, and power added efficiency (PAE).

AI-empowered receiver: The designed receiver architecture for Discrete Fourier transform-spread orthogonal frequency-division multiplexing (DFT-s-OFDM) transmitted signals is shown in Figure 3-14:. The receiver performs channel estimation and equalization of the received signal using legacy methods. A neural network-based demapper computes soft bits based on the equalized symbols and SNR estimate. The soft bits are used as inputs to an LDPC decoder.

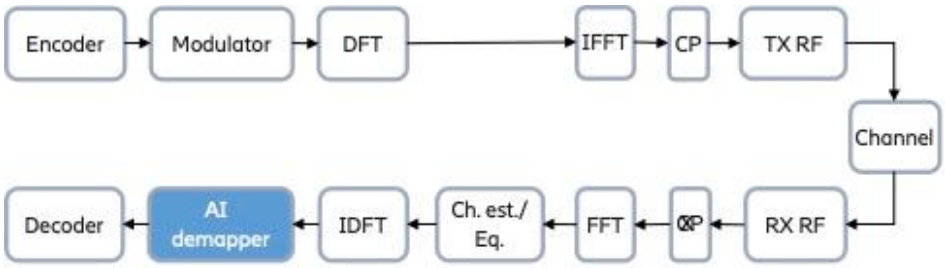


Figure 3-14: The architecture of DFT-s-OFDM signal transmitter with AI-empowered receiver for PA non-linearity compensation.

Neural network architecture: A fully connected neural network (FCNN) is used for demapping of the received symbols to soft bits. The inputs of the FCNN are the real and imaginary part of the equalized received symbol and the SNR estimate, and the outputs are soft bits. The input layer is composed of three neurons, there are five hidden layers each composed of 64 neurons, and the number of neurons in the output layer is equal to the number of bits per symbol. The hidden layers have a ReLU activation function and the output layer has a linear activation function. The neural network is trained using synthetic data generated from link simulators, and binary cross entropy is used as loss function for training the model. The training is performed over collected training data for different channel realizations and a range of SNR values.

Simulation results: Simulation assumptions for performance evaluations are summarized in Table 3-1.

Parameter	Setting
Waveform	DFT-s-OFDM
PA model	Memory-less polynomial model
PA back-off	4dB
Carrier frequency	3.5 GHz
Sub-carrier spacing	15 KHz
Bandwidth	10 MHz
Channel model	TDL-A
UE speed	3 km/h
RMS delay spread	100 nsec
HARQ re-transmissions	3
MCS Table	Table 5.1.3.1-2. in [TS38]

Table 3-1: Simulation assumptions.

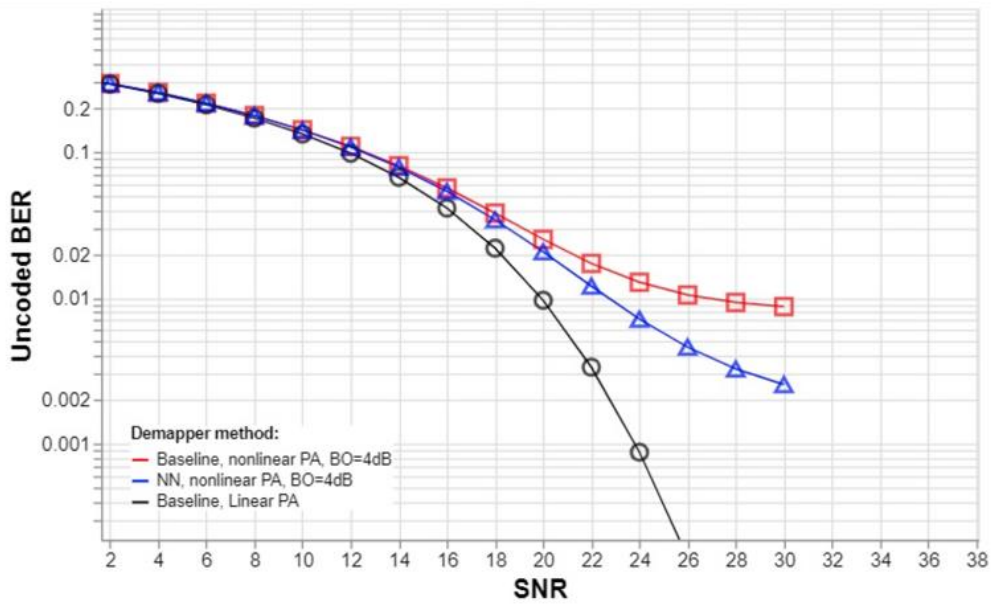


Figure 3-15: Uncoded BER performance for 64QAM modulated signals.

Figure 3-15: shows raw BER performance for 64QAM modulated signals. The performance of a legacy receiver in the presence of a linear PA can be considered as a lower bound on performance. The NN-demapper outperforms the legacy demapper and the performance gain is larger at higher SNR regime, where PA non-linearity has more severe impact on receiver's performance degradation.

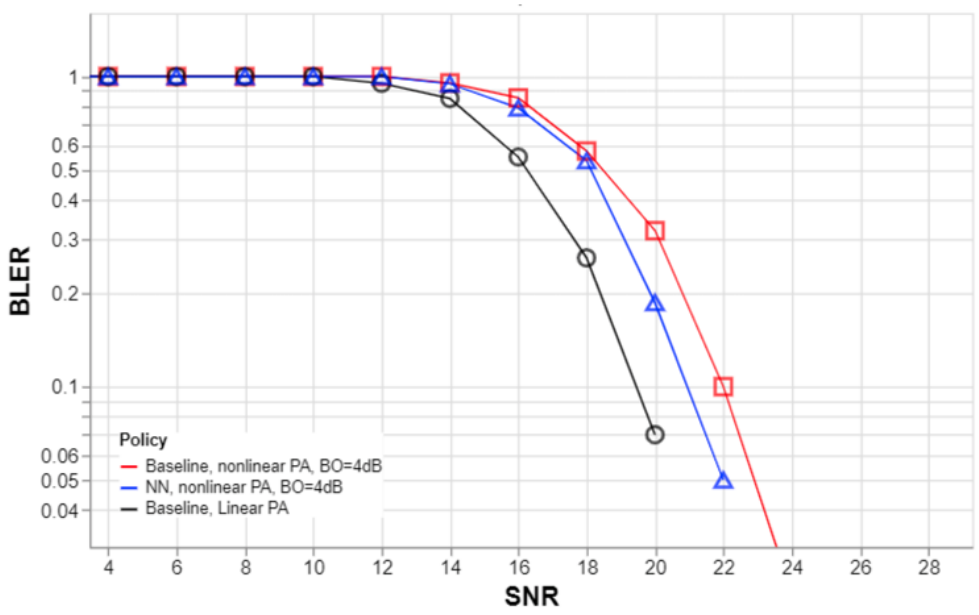


Figure 3-16: BLER performance for 64QAM modulated signals with MCS=19.

Figure 3-16: shows block error rate (BLER) performance of the considered methods for 64QAM modulated signals with MCS index 19. Comparing the BLER performance of the legacy demapper in the presence of PA non-linearity, and that of the proposed receiver with NN-demapper, it can be seen that, at 10% BLER, which is a reasonable operating point, the proposed method reaches roughly 1dB performance gain.

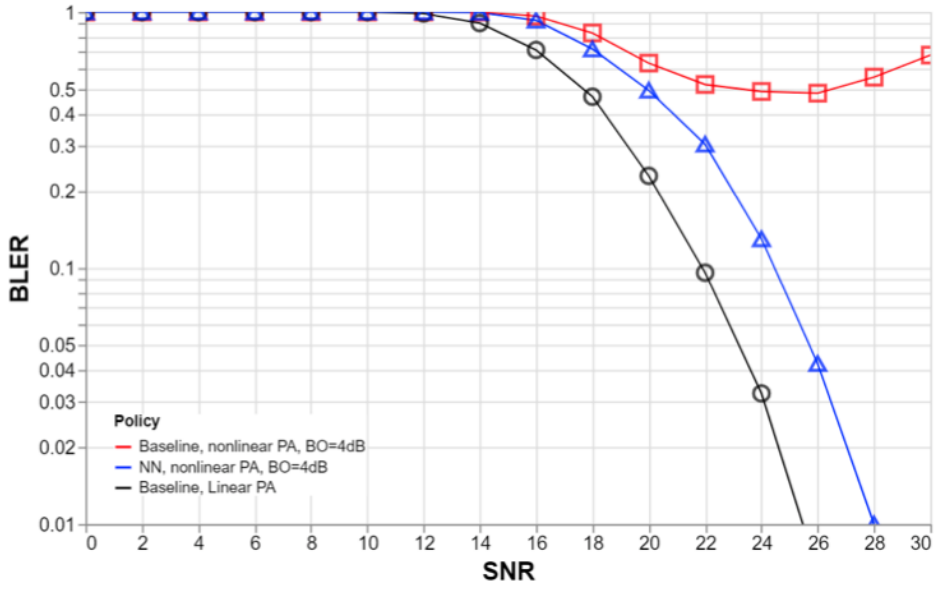


Figure 3-17: BLER performance for 256QAM modulated signals with MCS=20.

Figure 3-17: shows the BLER performance of the studied methods for 256QAM modulated signals. Evaluations are performed for MCS index 20, corresponding to the lowest code rate for this modulation order. For this modulation scheme even for the lowest code rate defined, the legacy receiver cannot successfully detect the signals and fails to reach 10% BLER. However, the proposed receiver with NN-demapper can provide satisfactory results without any visible error-floor. This confirms that the proposed receiver can enable higher order modulations while providing signal reception with desired reliability.

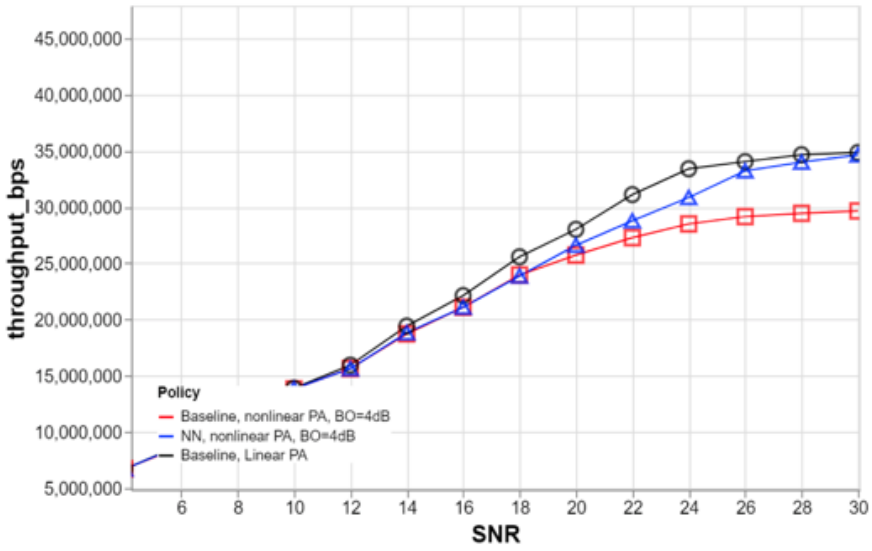


Figure 3-18: Throughput performance of the NN-empowered receiver and legacy receiver with link adaptation.

The achievable throughput of the considered methods in the presence of link adaptation is shown in Figure 3-18:. The highest modulation order that was considered in the simulations with link adaptation was 64QAM. The legacy receiver in the presence of linear PA provides an upper bound on the achievable throughput. In the presence of PA non-linearity, the legacy receiver achieves lower throughput compared to the upper bound, and the performance gap become larger at higher SNR values. At high SNR regime, the throughput saturates at a lower level compared to the one that can be achieved when there is no PA non-linearity. The proposed receiver with NN-demapper achieves higher throughput compared with legacy receiver. The performance gain is larger at higher SNR values and the achieved throughput saturates at the same level as the one for the upper bound on throughput.

Conclusions: The simulation results confirm that the proposed method can increase throughput (20% improvement) and/or extend the coverage of a communication link in the presence of PA non-linearity, and to enable enhancing energy efficiency (70% improvement of PA power-added efficiency) at the transmitter side. The proposed method can be effective in compensating the impact of in-band distortion, however, for the out-of-band emission, the transmitter's parameters need to be set so that the resulting transmitter's signal fulfil out-of-band emission's requirements. Therefore, the proposed method would be suitable for use cases in which the inband distortion is a limiting factor (e.g. high throughput transmission using high-order modulation schemes). This method can be also used for compensating other hardware impairments such as phase noise at high frequency bands as shown in [FS20]. The proposed method can be used in uplink or downlink scenarios and can enable using higher order modulation for PAs operating in non-linear energy efficient operating regimes. The proposed method requires no extra processing at the transmitter side but requires more complex processing at the receiver side compared to the legacy receiver methods. In uplink scenarios which are usually coverage limited, this technique enables the user equipment (UE) to increase its output power, and enhance coverage, while more capable receivers using this method can perform signal detection in the presence of distortions at the base station.

The proposed method contributes to Hexa-X connecting intelligence targets on BER and BLER gains with respect to legacy methods and resilience against hardware imperfections (T1), resource utilization enhancement by improving spectral efficiency and throughput gain (T2), and energy efficiency enhancement (T5). This enabler could be used in wide range of 6G uses cases that require extreme performance, e.g., "fully-merged cyber-physical worlds"; extreme reliability, e.g. "interacting and

cooperating mobile robots"; extreme energy efficiency, e.g. "network trade-off for minimized environmental impact"; or coverage enhancement, e.g. "earth monitor".

3.1.5 AI-Based Enhancements for Sub-THz

AI-native air interface has a high potential in improving the spectral efficiency, flexibility, and resilience against hardware impairments, as shown by various earlier works [KHH+22, PKH+21]. In particular, the existing solutions have demonstrated that learning the transmitter and/or receiver processing can result in reduced overhead [KHH+22], lower out-of-band emissions [KHH+22], and higher detection accuracy under severe inband distortion [PKH+21]. However, most of these works have assumed sub-6-GHz scenarios, where propagation models and hardware impairments are quite different from the higher frequencies.

In this section, we demonstrate how to extend the analysis of ML-based physical layer design to sub-THz frequencies, where especially the hardware impairments cause considerable challenges for algorithm design. Our results indicate that joint learning of sub-THz transmitter and receiver processing can lead to higher spectral efficiency and high resilience against hardware impairments. This extends our original findings reported in [HEX-D42], where similar results were obtained for a sub-6-GHz center frequency.

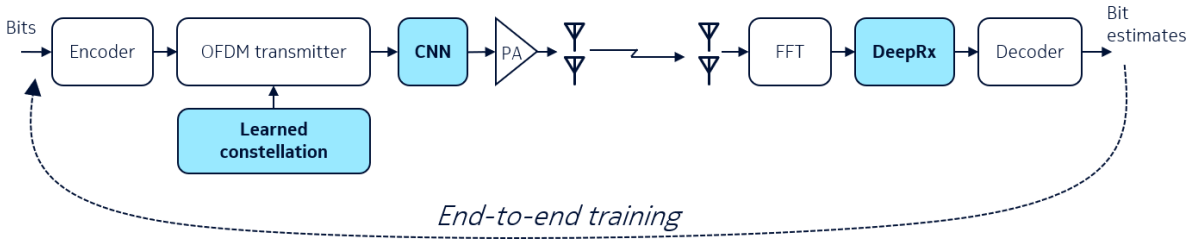
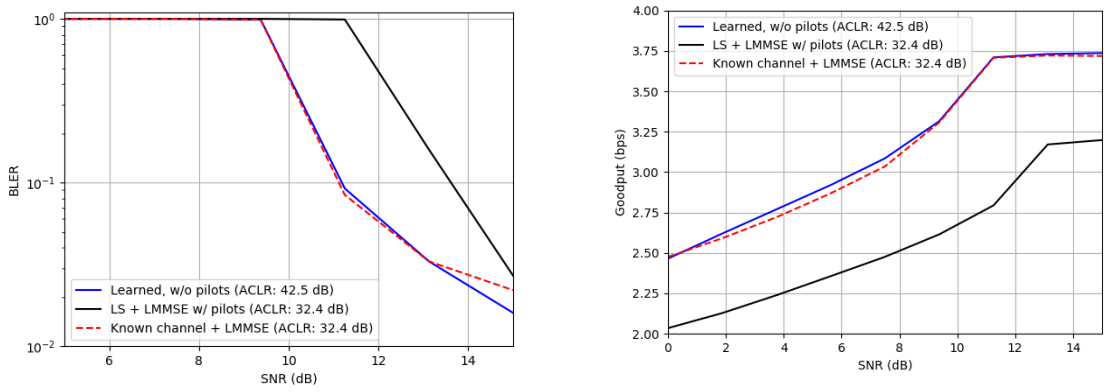


Figure 3-19: System model for end-to-end learned sub-THz link.

Here, ML-based processing is utilized both at the transmitter and receiver side, as depicted in Figure 3-19. In the transmitter, the constellation shape is being learned to facilitate pilotless detection [KHH+22], which greatly reduces transmission overhead. Namely, with such a learned constellation shape, all resource elements (REs) can be used for data transmission. In addition, the transmitter utilizes a small convolutional neural network (CNN) to make the transmit waveform better suited for a nonlinear power amplifier. In the receiver side, the complete receiver is implemented as a deep convolutional ResNet (DeepRx), similar to [HKH21].

A key aspect of the proposed approach is to learn the transmitter and receiver elements jointly. This approach is referred to as end-to-end learning, and it ensures that the system learns to utilize the air interface as efficiently as possible. In practice, such end-to-end learning is achieved by implementing the complete transmitter and receiver link as a single differentiable model. This way, the system can be interpreted as an autoencoder, where the transmit bit stream is the input, and the bit likelihoods estimated by the receiver are the output. The effective model consists of a combination of trainable and non-trainable layers, where the latter include elements such as the channel model and noise source. The former includes the constellation shape, transmitter-side CNN, and the learned receiver (DeepRx). With such formulation, the training can be carried out with supervised learning, where the original bit sequence can be used as the labels of a binary cross-entropy loss function.

To introduce a realistic sub-THz channel model, the measurements carried out in Hexa-X WP2 have been utilized [HEX-D23]. In particular, the measurement-based channel model for a center-frequency of 140 GHz reported therein has been implemented as a TensorFlow layer in order to facilitate the end-to-end learning as described above. In addition, a measurement-based power amplifier (PA) model has been included in the transmitter, which is pushed well within its nonlinear operation region to represent a nonideal sub-THz PA. To evaluate the performance limits of the proposed approach, a CP-OFDM waveform is utilized, which has a rather high peak-to-average-power ratio (PAPR). In this study, a subcarrier spacing of 1 MHz is assumed.



(a) (b)

Figure 3-20: (a) BLER and (b) goodput of the learned system, compared against baseline solutions.

The performance results are shown in Figure 3-20 in terms of the BLER and goodput. The latter is calculated similar to [KHH+22], and it reflects the useful bit rate after accounting for the overhead reserved for the transmission of pilots and the cyclic prefix. Firstly, it can be observed that the proposed approach achieves essentially the same BLER as the genie-aided baseline relying on perfect channel knowledge. With a BLER of 10%, the gain over the practical baseline is over 2 dB.

When considering the goodput, the gain over the practical baseline is equally evident. Depending on the SNR, the goodput gain is in the order of 20-30%. Note that the goodput of the genie-aided baseline is calculated without any pilots, meaning that the learned scheme can only outperform that by suppressing the nonlinearities produced by the transmitter PA. In this particular example, the learned scheme is mainly targeting higher ACLR, meaning that the throughput is on par with the genie-aided baseline, while the out-of-band emissions are 10 dB lower.

Altogether, this solution addresses the target T1 by providing a BLER improvement, and target T2 by improving the throughput. The BLER gain is approximately 2 dB, while the throughput improvement is in the order of 20-30%.

3.1.6 Channel charting based beamforming

Channel charting (CC) is the task of locating users relative to each other in an unsupervised way and can be viewed as a way to discover a low-dimensional latent space charting the channel manifold. A more thorough description of CC is given in section 4.5. In this work, we propose to leverage a learned chart together with the recently proposed location-based beamforming (LBB) method [LYP+22a] and presented in [HEX-D42] to show that CC can be used for mapping channels in space or frequency.

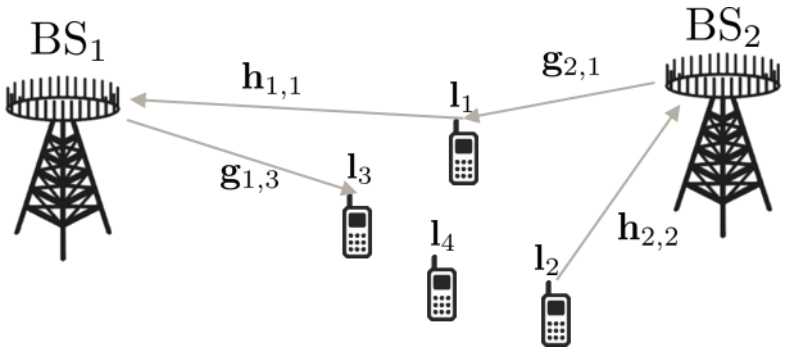


Figure 3-21: Setting where two BSs communicate with users. h is an uplink channel and g is a downlink channel.

LBB, as the name suggests, traditionally relies on a user's location to predict its associated optimal precoder. However, users' locations are not always available and usually require a separate system (e.g., GNSS) to provide them. On the other hand, CC only requires channel measurements to produce a chart of pseudo-locations coherent enough with the real locations. Such a chart appears to be the ideal candidate to replace the latter for unsupervised deployments. This would effectively allow a base station to choose appropriate precoders using the LBB technique but relying on chart locations instead of spatial locations. Ultimately, the goal of the proposed contribution is to map channels in space or frequency, which amounts to determine a downlink precoder $\boldsymbol{w_{k,j}}$ that should be highly correlated with the downlink (target) channel $\boldsymbol{g_{k,j}}$, based on the knowledge of the uplink (origin) channel $\boldsymbol{h_{i,j}}$, as illustrated on Figure 3-21. The origin and target channels are not necessarily on the same band nor even on the same base station (k and i may be different).

The inference phase of the proposed method comprises three steps (depicted on Figure 3-22:):

1. At first, D latent variables $\boldsymbol{z_{i,j}} \in \mathbb{R}^D$ are computed from the origin channel $\boldsymbol{h}_{i,j}$ using channel charting methods [SMG+18] (with $D \ll$ channels' dimension) at the ith BS. In particular, the efficient channel charting method of [LEM21] is used in this paper.
2. Then, the latent variables $\boldsymbol{z_{i,j}}$, that can be seen as a compressed version of the channel are sent from the ith BS to the kth BS.
3. Finally, the kth BS performs location based beamforming using the latent variables as inputs, so as to output a precoder.

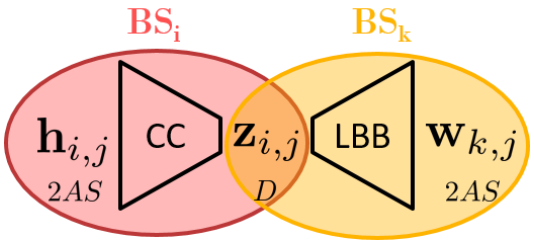


Figure 3-22: Schematic view of the proposed method (the real dimension being shown below each variable).

In order to calibrate the LBB step, a training phase is required. The method proposed in [LPY+22a] is used, with the different that here chart locations are taken as input instead of the true spatial locations. A database of chart locations and associated target channels is first built using classical channel estimation methods. Then a neural network comprising a random Fourier features layers is trained to minimize the misalignment of the predicted precoders with respect to the target channels according to the following cost function:

$$\mathrm{CF}_k \triangleq 1 - \frac{1}{N}\sum_{n=1}^{N} \frac{\left|\boldsymbol{w}_{k,n}^H \boldsymbol{g}_{k,n}\right|}{\left\|\boldsymbol{g}_{k,n}\right\|_2^2}.$$

The proposed method can be applied to several tasks, depending on the source and target channels. For example, if CC and LBB are done in the same base station ($i = k$) but the frequency bands are different for the uplink (source) and downlink (target) channels, then channel mapping in frequency is carried out. In the more general case where CC and LBB are done in two distinct base stations ($i \neq k$), the proposed method allows to determine a precoder at one BS relying on a channel estimated at another BS (potentially in another frequency band), so that no channel estimation is required at the precoding BS. This is particularly interesting if the CC done at one BS is used to perform LBB at several other BSs (say $B - 1$), since in that case channel estimation would be needed at only one BS to determine precoders at B BSs. The overhead due to channel estimation would then be reduced by a factor of B. Moreover, the backhaul requirements are kept reasonable, since only D real numbers (dimension of the channel chart which is much smaller than the dimension of the channel) have to be shared for each user.

The model is evaluated on the DeepMIMO [ALK19] dataset on the 'O1' outdoor scenario where two BSs on opposite sides of a street communicate with UEs spread across the map. Its performance is evaluated in terms of the normalized correlation between the precoder and the target channel. It is expressed as

$$\eta = \frac{|w^H g|^2}{\|g\|_2^2}.$$

The method is compared to the pure LBB one [LYP+22a] where the precoder is predicted based on the knowledge of the true locations. Figure 3-23: show the cumulative distribution function (CDF) of the correlations η–the closer to a Dirac at $x = 1$ the better. The curves are almost identical meaning that both methods achieve similar performance. This indicates that, although lacking positional information about the UEs, the presented approach is capable of producing precoders of good quality.

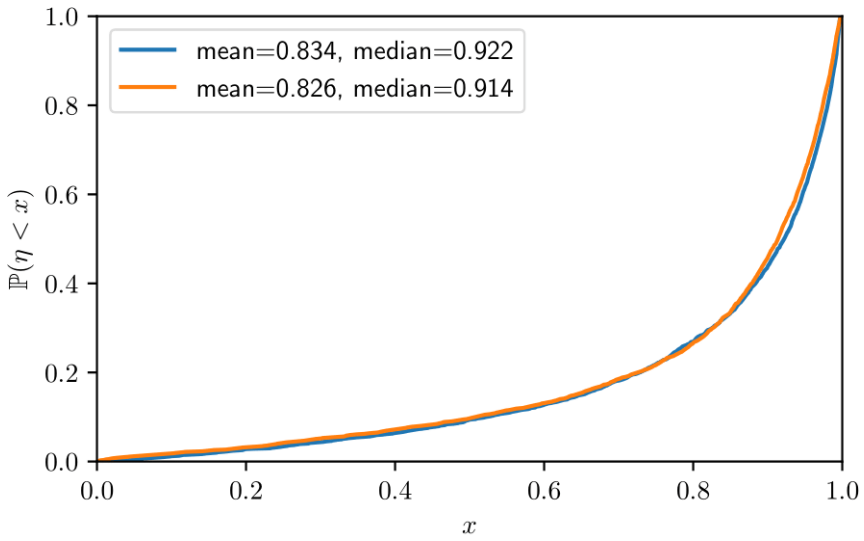


Figure 3-23: CDF of the correlations. Blue curve corresponds to the original LBB at BS1. Orange curve corresponds to CC at BS1 and LBB at BS2 at different frequencies.

More details on this work are given in [LYP+22a].

3.2 Improvements in E2E network operation & management

It is beyond doubt that the inclusion of AI/ML techniques in management and orchestration operations for B5G and 6G networks will be a key enabler in order to support the new requirements associated to these networks i.e., network complexity, network heterogeneity, extreme-edge domain integration, etc. as described in [HEX-D42] and [HEX-D62].

This sub-chapter presents two technical enablers for AI/ML-based solutions for network Management and Orchestration (M&O). The first enabler addresses scheduling of finite User Plane Function (UPF) resources in support of low latency communication services. This is done by introducing distributed AI/ML agents at the network edge. The agents are tasked to forecast traffic patterns, serving as input for decisions on adding or reducing local UPF instances. The second enabler presents a framework for predictive orchestration to enable M&O operations in terms of flexibility, dynamicity, resource placement etc. Performance gains are quantified in terms of inferencing accuracy, latency, network energy efficiency.

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

As explained in [HEX-D42] low-latency requirements relying on edge limited computational power not always be satisfied, because of the number finite resources provided by the data plane function, i.e., the UPF. One challenge addressed by the inclusion of AI in network management and orchestration is to avoid degradations in service caused by finite User Plane Function (UPF) resources while ensuring security, privacy and reducing data flow demands. The UPF has critical role in the data transfer within the 5G network, processing, routing and forwarding the packets from the UEs to the Internet. In this way, an edge AI agent serves for the optimal auto-scaling of local UPFs placed at the network edge, in support of low latency communication services. This solution is strongly applicable to all 6G use cases which depend on reliable low latencies including those under the umbrella of Massive Twinning and Robot to Cobot use case families described in [HEX-D12]. The agent is responsible for inferring traffic patterns associated with local UPFs and using this information to foresee opportunities for optimising resource allocations. The resources and scaling to be considered include additional or reduced local UPFs instances servicing a geographical area in a given time period. The logic behind the optimisation relies on AI functions, tuned for the capabilities of the servicing local UPFs. The pre-emptive scaling decisions are relayed to a Management and Orchestration (M&O) block responsible for the resource scaling.

The architecture of the solution, shown in Figure 3-25: takes into account the security a privacy concerns associated with sensitive data transmission by keeping the monitoring data local at the edge. This results in no data transmission external to the edge at runtime. The data used for training/retraining and testing of the time forecasting ML models is transmitted to a cloud server only on an as-needed basis. Further precautions can be readily included in the solution, such as data encryption or pseudonymization.

As the collection of UPF network data is in its early stages, limited availability, limited coverage, and limited historical readings curb the evaluation of appropriate time forecasting models under realistic runtime scenarios. For this reason, the proposed solution architecture has been designed to be suitable for a variety of time forecasting models. The model serving is done through the Tensorflow Serving framework TFS23] allowing the seamless adaptation of data transformations and dimensionality reductions required by different models. As a PoC , a Long Short-Term Memory (LSTM) flavour model was used, consisting of a graph convolution layer followed by LSTM and dense layers. The model used 2 metrics available from the Monitoring Platform (a set of software components for retrieving, managing and storing information about the UPF computational and networking usage), available in 5 minute timesteps:

- ‘host:upd_pkts packets_sent’ - Total number of UDP packets sent

- ‘host:upd_pkts packets_rcvd’ - Total number of UDP packets received

UDP traffic was chosen because of its role in time-sensitive communications, i.e industrial XR applications. The model was tuned to predict the system in the next consecutive 30 minutes given as input the previous hour’s measurements.

Due to the lack of available data for the PoC, a simulation was performed to produce synthetic metric data over a span of 6 weeks to use for tuning, training, and testing the models, using as a baseline the known behaviour of the system inferred from UPF stress tests, as well as patterns in urban mobility taken from available UE datasets [NCS+19].

The UPF stress tests consisted of the monitoring platform developed in WP6, UPF and 5GC Network functions provided by a third party, and the Spirent landslide tool. The number of UEs, the time window and the traffic amount are examples of the parameters that were configured from the landslide tool during the tests. The behaviour of the UPF under different input traffic inputs was used to simulate the metrics used for training.

The simulation incorporated daily and hour fluctuations in the baseline signal, as well as noise, to better represent a real-life scenario. A sample week is shown in Figure 3-24:. Because the 2 signals are very similar to the naked eye, for visual clarity, the packets received, and the ratio of packets received over packets sent is shown.

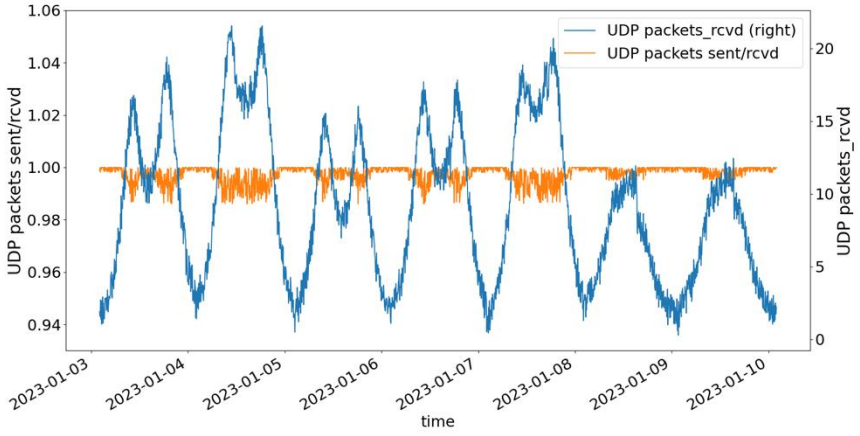


Figure 3-24: Synthetic network traffic in one UPF for 1 week.

The M&O block responsible for the UPF scaling also orchestrates the deployment of the edge AI agent, as well as the update of the active ML models employed by the individual agents. A third component, the edge AI monitoring block, allows for continuous feedback on the validity of the inferred traffic patterns, and can trigger a retraining or model update through the M&O block.

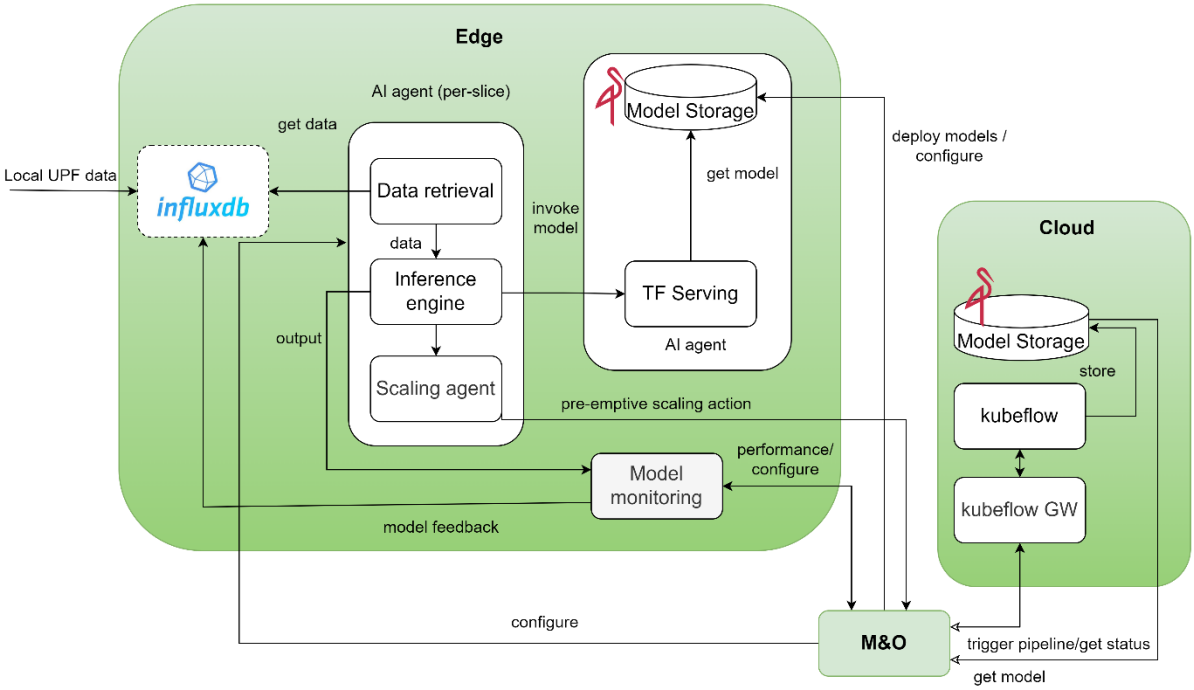


Figure 3-25: Automated UPF scaling using AI block diagram.

Runtime

During runtime, the local UPF data is collected and stored in an instance of InfluxDB at the edge. The Data retrieval block queries for available data which it validates and passes to the Inference engine. The Inference engine requests an inference from the TensorFlow Serving block, which is then passed to the Scaling agent and Model monitoring blocks. The Scaling agent contains an AI-based decision algorithm for triggering an action request to the M&O for the scaling of local UPF resources. The Model monitoring block compares the previous inference to the current network traffic status, stores this information in a database for online viewing, and if the performance is below threshold, the Model monitoring triggers a retraining/model review to the M&O.

Training

The model training, achievable performance evaluation, tuning, and development are done on cloud, due to the high demand of resources these activities involve. When a new model, or a retrained model, are available, the M&O reconfigures Tensorflow Serving to use the appropriate model for a given instance of edge AI agent.

For the PoC, the simulated input data described above was split by 50% train, 20% validation, 30% test samples. Normalization, and a sliding window average to increase the signal to noise ratio, were used as data pre-processing. The trained model achieved a Mean Absolute Error (MAE) of 0.003 (naive MAE of 0.051) and a systematic Mean Absolute Percentage Error (sMAPE) of 0.13 on test data, however the evaluation of the trained model for this scenario is directly tied to the functional requirements of the AI agent system which has a high tolerance due to the nature of the input data and the desired functionality of the agent using the model.

Target Evaluation

The distributed AI for automated UPF scaling in low-latency network slices technological enabler defined 3 KPIs related to fulfilment of the quantifiable target T3:

- Inferencing latency with target of less than the half the timestep of the input data (< 30 seconds);
- Training latency with target of time from AI M&O request of training to the instantiation of the ML training pipeline < 1 minute;

- Inferencing accuracy with target of LSTM 89% accuracy on training data, 83% accuracy on runtime data.

The inferencing latency is defined as the time from when the inferencing agent requests an inference from the deployment of TF Serving to when the agent receives the inference, not considering the effects of network latency. To achieve this, both the agent and TF Serving were deployed on the same virtual machine. A sample size of 1000 requests was used with the resulting median time for latency being 6.8 ms, with a distribution shown in Figure 3-26:. This is well within the target of half the size of the timestep of the input data (5 minutes for the PoC).

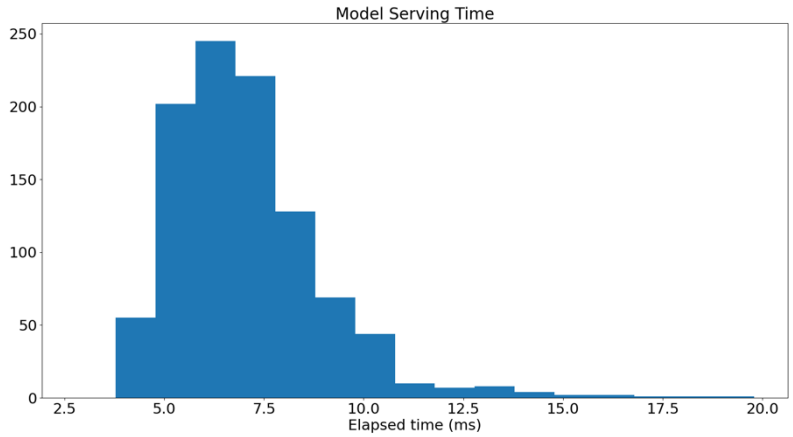


Figure 3-26: Inferencing latency for the distributed AI for automated UPF scaling in low-latency network slices technological enabler.

The training latency is defined as time from the AI M&O request of training to the instantiation of the ML training pipeline. The target of this latency is less than one minute, which is comparable to the time it would take to do the operation in a manual fashion. To measure this latency, 16 requests were sent to pipeline orchestrator, Kubeflow, and the time from the request to the instantiation of the first pod of the training pipeline was measured. The resulting latency was found to be less than one second, well below the one-minute target.

The inferencing accuracy KPI has 2 targets. The first is the training accuracy, which for the PoC was measured as 83% with a validation accuracy at 81%. This falls around the target of 89% accuracy defined before the design of the model. The second, the runtime accuracy, looks at the performance of the model in the context of the overall solution. To perform this test, the deployed PoC was used, and the average accuracy over a 5-hour period was measured. The resulting performance of the PoC LSTM model in the solution was evaluated using the systematic Mean Absolute Percentage Error (sMAPE). The achieved accuracy was determined to be 86%, in-line with the 83% accuracy target.

3.2.2 AI/ML-based predictive orchestration

It is beyond doubt that the inclusion of AI/ML techniques in management and orchestration operations for B5G and 6G networks will be a key enabler in order to support the new requirements associated to these networks i.e., network complexity, network heterogeneity, extreme-edge domain integration, etc. as described in [HEX-D42] and [HEX-D62]. The majority of the SDOs, since 5G, have already started introducing AI/ML components or frameworks as part of their Management and Orchestration (M&O) stack, as a sort of "natural choice", for being able to cope with the increasing complexity of automating and managing the network. One of the most commonly known efforts, in this regard, comes from the 3GPP with the incorporation of two AI/ML-based modules to its standardized architecture: (i) the Network Data Analytics Function (NWDAF) [23.288]; and (ii) the Management Data Analytics

Function (MDAF) [28.533]. Many other global SDOs and industry alliances, such as the ISO/IEC, ITU-T, IEEE, GSMA or the O-RAN alliance have a clear focus on involving AI/ML for future networks as reflected in [ETSI-34]. The majority of these proposals are evolving from the early stage where high-level key building blocks were described to in-depth network data analytic frameworks. Therefore, once more, this remarks the paramount role that AI/ML will play in M&O procedures for efficient and optimal decision-making operations.

AI/ML-driven orchestration offers a wide range of approaches in order to ease the M&O-related operations within the network [HEX-D62], being *Predictive Orchestration* one of those approaches. This method aims at foreseeing future states of the network, using prediction/forecast AI/ML-techniques mainly based on time-series forecasting, to be able to maximize the output of M&O operations in terms of flexibility, dynamicity, real-time, or resource placement, among others. Predictive Orchestration might use legacy time series techniques such as Exponential Smoothing or Auto Regressive Integrated Moving Average (ARIMA) [SEV18] as they have already probed their effectiveness as time-series predictors. Besides, it is important to consider the volume of the dataset to be used and the forecast expected time because, in some cases, these legacy techniques may outperform AI/ML-based methods (e.g., Long-Short Term Memory (LSTM) and Recurrent Neural Networks (RNNs) [TSP19]), but although legacy time-series predictive techniques could be quickly implemented, they are sometimes not accurate nor flexible enough to adapt to complex datasets. On the other hand, AI/ML-based predictive techniques, such as LSTM, are able to face long and complex datasets achieving greater accuracy (although at the cost of being computationally expensive and requiring large amounts of data and having long training times). However, these techniques might be used to support existing M&O modules in every layer of the M&O architecture (e.g., service, network, infrastructure and design), as envisioned in [HEX-D62].

Predictive Orchestration requires a process performing data collection in a similar fashion to the capabilities given by the 3GPP NWDAF, i.e., monitoring, data collection and storage. Hexa-X aims at integrating an advanced monitoring system which is able to provide monitoring and telemetry from all network segments, from infrastructure to applications. This requirement, allowing the overall monitoring system to collect and aggregate data from multiple domains and/or different architectural layers, is considered a key enabler to provide data-driven orchestration functionalities (i.e., advanced AI/ML functions such as cross-layer/cross-domain self-adaptation and self-optimization decisions required for Predictive orchestration) [HEX-D62]. Finally, as it can be extracted from the previous statements, it is critical to remark the importance of enabling the monitoring functions to exchange information between layers and external domains to achieve real E2E Predictive Orchestration.

As suggested in [DMR+20], three categories of forecasting algorithms should be implemented for performing proper Predictive Orchestration: *(i) Long-term Forecasting (*LTF*),* operating on the larger timescale (e.g., hours) and in charge of predicting aggregated high-level KPI metrics across layers and domains, e.g., involving Network Slice migration from one operator‘s domain to other operator’s domain due to an expected peak on the Network Slice services aggregated load; *(ii) Mid-term Forecasting (MTF),* operating on the order of tens of minutes, and issued to make predictions across layers and domains for M&O operations requiring faster actions, e.g., scaling across the infrastructure layers, or update the weights of a RL model running within an operator’s domain, among others; *(iii) Short-term Forecasting (STF),* operating in the shortest scale (e.g., seconds or milliseconds) and carrying out local predictions within a layer or domain which may be sent as an input to the LTF and MTF algorithms, or for other M&O functions to suggest further actions, such as traffic load prediction within a VM in the Infrastructure layer, potential security breach on a Containerized Network Funciton (CNF), etc. These three algorithms can be encompassed within the *Data-Driven M&O processes* defined in [HEX-D62], more specifically, within the *AI-Driven Orchestration M&O processes.* Thereupon, they should be able to effectively perform: *(i) Basic Orchestration Actions,* comprise the baseline for more complex M&O processes, they can be understood as “atomic” M&O operations i.e., instantiating, scaling, updating, upgrading/downgrading, terminating, etc.; *(ii) Orchestration processes,* considered as complex M&O actions that are composed of two or more Basic Orchestration actions i.e., E2E, seamless integration processes, programmable processes, automation processes, etc.

Figure 3-27 depicts a potential mapping between the aforementioned Predictive Orchestration forecasting algorithms and the Hexa-X M&O architecture. It is important to remark that this is just an example of all the possible implementations that could be deployed aligned with this architecture. As it can be seen, the forecasting algorithms have been split between the *Management Functions* block and the *AI/ML Functions* block at the Network Layer. Although the three algorithms are AI/ML components, this implementation has been selected as an example to demonstrate that they can be allocated outside of the *AI/ML Functions* block if they help the respective M&O block to perform its M&O action. The *Management Functions* block enables the execution of basic management capabilities: *(i) fulfilment capabilities,* collection of capabilities that allow provisioning instances of any given resource; *(ii) assurance capabilities,* collection of capabilities that enable network monitoring and failure prediction; *(iii) artifact management capabilities,* set of assets that aid operators in their fulfilment and assurance capabilities (i.e., catalogues, inventories, etc.). Therefore, the LTF forecasting algorithm is allocated inside this block to ease all the operations related to *fulfilment capabilities* and give high-level support for those related to *assurance capabilities.* Besides, as it is depicted on Figure 3-27, the *Management Functions* block uses (i.e., ingests data from, requests information to, etc.) the *Monitoring Functions* block and the *AI/ML Functions* block as complementary functions that give support to its basic functionalities. From the point of view of predictive orchestration, the *Monitoring Functions* block will provide the required capabilities for collecting and storing all the data-sets (e.g., time series) required to perform predictions and then, be consumed by the required elements, in this case by the three forecasting algorithms. Paraphrasing [HEX-D62] *AI/ML Functions* description:

"AI/ML functions are intended to provide the mechanisms to build out the knowledge and the intelligence for controlling, managing and optimising the services deployed on the network, and to take decisions about the actions to be performed at the various architectural layers (…). Certain AI/ML functions would be specifically designed to support the activity of management functions (within the scope of M&O), while others could be deployed for other purposes"

It can be concluded that the main idea behind this functions' block is to give support to the *Managing Functions* block. Therefore, the MTF and STF algorithms have been allocated within the *AI/ML Functions* block. Once more, it is important to remark that, although for the sake of simplicity these elements have been represented inside the *Network Layer*, they should act upon all the layers of the M&O architecture.

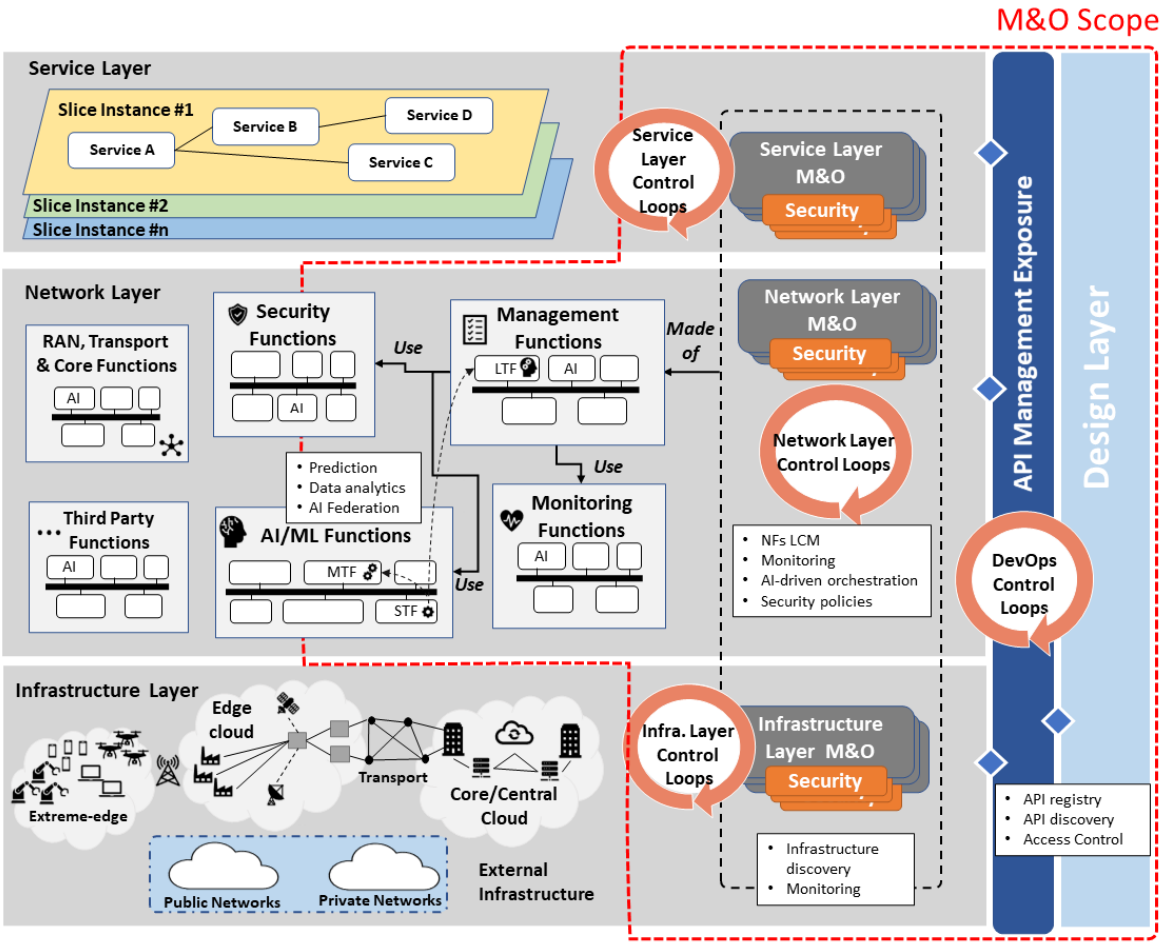


Figure 3-27: Predictive orchestration components mapping to Hexa-X M&O architecture.

4 AI/ML as enabler for 6G network sustainability

AI/ML can serve in multiple ways to improve the sustainability of 6G networks, mostly in terms of energy efficiency. This chapter is dedicated to three main categories of ideas, in which we describe in the following. In the first application, the AI/ML universal functional approximation property is utilised to learn the relationship between the input condition and optimal values of a complex optimisation problem in the network. The optimisation problem is chosen such that heuristic methods are not close to optimal, while the optimal solution is very complex to implement in real time. Specifically, we investigate the optimisation problems that are impossible to solve in polynomial time (mixed-integer problems such as radio resource allocation). The second category of applications is to apply AI/ML well-known statistical problem, e.g., channel estimation, by using training on the relevant data. This allows to move large part of the computational complexity to the training phase, while keeping the inference computationally affordable. Since the inference phase is executed orders of magnitude more than training, the total computational complexity is reduced, thus the total energy footprint of the system. The last application that we investigate in this chapter is the use of AI/ML to acquire more meaningful data from the available network data, in order to facilitate further functions in the network to save energy. For instance, we study channel charting, where the output of the AI/ML operations could be used to enhance multiple network functionalities such as beamform tracking.

Relevant KPIs applicable for these technical enablers can be summarised as throughput enhancements, reduced channel estimation error, and complexity gain. The complexity gain of the AI/ML algorithms

is evaluated by the reduced processing time/number of mathematical operations in the algorithm to produce the outputs while achieving other problem-specific metrices. The proposed AI/ML-based channel estimation algorithms also achieve improved channel estimation accuracy while reducing the resource overhead compared to the conventional approaches. Furthermore, the solutions are flexible and generalisable due to the inherent learning capability of AI/ML, which can thus quickly adapt to different system configurations and environments without much degradation in performance. Therefore, these low complexity, energy efficient, flexible and adaptable AI/ML-based solutions enable achieving the KVIs towards 6G sustainability and flexibility targets.

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

This enabler is an extension of the work in [HEX-D42] and the objective of the study is to investigate the potential of ML-based resource allocation algorithms for cell-free massive MIMO in overcoming algorithmic deficiencies associated with conventional resource allocation algorithms. Radio resource management (RRM) in communication networks enables improving the system performance by efficient utilisation of available resources. RRM is often done to achieve a desired objective (such as maximising the system sum rate or minimum user rate etc.) where optimisation-based or heuristic-based techniques are used to solve the optimisation problems. These conventional RRM methods face several challenges with the increased network and parameter complexity in modern communication networks, 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). Recently, ML approaches are being used for resource allocation tasks to overcome the above-mentioned challenges using their data-driven learning capability. Specifically, there are several studies in literature for deep learning-based power control in cellular and cell-free massive MIMO systems [AZB+19, CCB+20, ZNG20]. In our previous work in [RSR+21] we proposed an unsupervised learning-based power control algorithm to maximise the minimum user rate in an uplink cell-free massive MIMO network.

In [HEX-D42], we have shown that a similar ML-based approach could be used for the joint optimisation of user power and fronthaul capacity allocation for CSI and data in a cell-free network to maximise the system sum rate. There, we assume that the fronthaul links between the APs and the CPU have a limited capacity and it is needed to properly utilise those limited capacity fronthaul links to transmit CSI and actual uplink data signals to the CPU from the APs in order to improve the system sum rate performance. Furthermore, uplink power control is done in order to further improve the system sum rate. As it is explained in detail in [HEX-D42], a DNN (which is denoted as PowerNet in Figure 4-1 and Figure 4-2 below is trained to directly optimise over the sum rate objective as a custom loss function to produce uplink user power allocations and fronthaul capacity allocations for CSI and data transmission between the APs and the CPU. There we presented simulation results showing that the ML-based RRM algorithm has a similar sum rate performance and a lower computational complexity compared to the optimisation-based algorithm. Here we present further results to show the flexibility of the ML-based approach which can provide expected performance under different system configurations. Figure 4-1 shows the average sum rate performance when different total fronthaul capacity values are allowed. There, we have used the model trained for total fronthaul capacity C = 1 bits/s/Hz scenario to generate the results of other instances. From the plots, we can see that PowerNet produces similar results to the baseline solution and online learning slightly improves the average sum rate across the whole fronthaul capacity range considered. Furthermore, Figure 4-2 shows the cumulative distribution of the sum rate for when there are 5 users in the network where the pre-trained model with 10 users is used to obtain the user power allocations and fronthaul capacity allocations. This flexibility of the ML-based approach overcomes the challenge associated with conventional optimisation techniques for resource allocation where it is needed to reformulate and solve the problem when the network parameters or system parameters change.

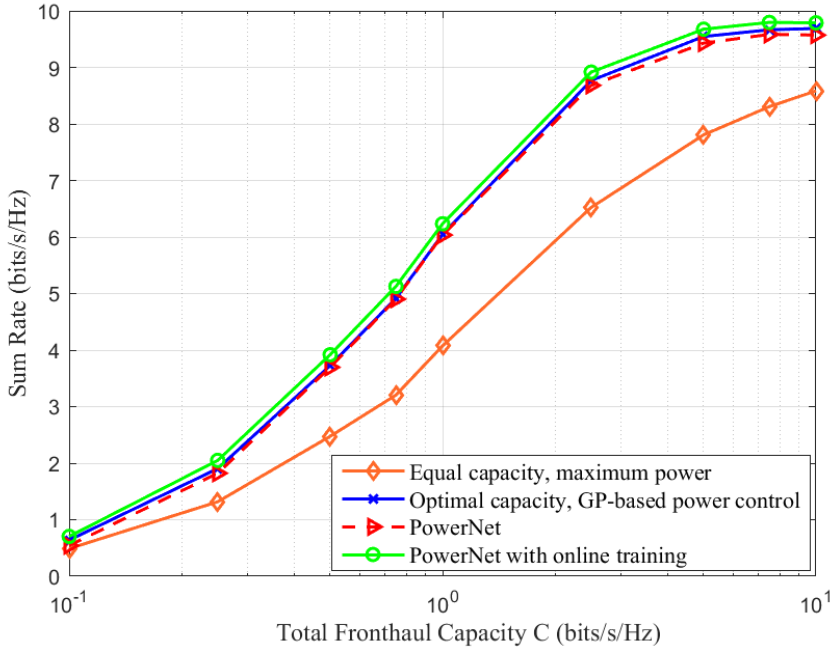


Figure 4-1: Average sum rate performance with different power control and capacity allocation algorithms for 50 APs and 10 users in the cell-free massive MIMO network. The DNN results are obtained using the model trained with total capacity C = 1 bits/s/Hz.

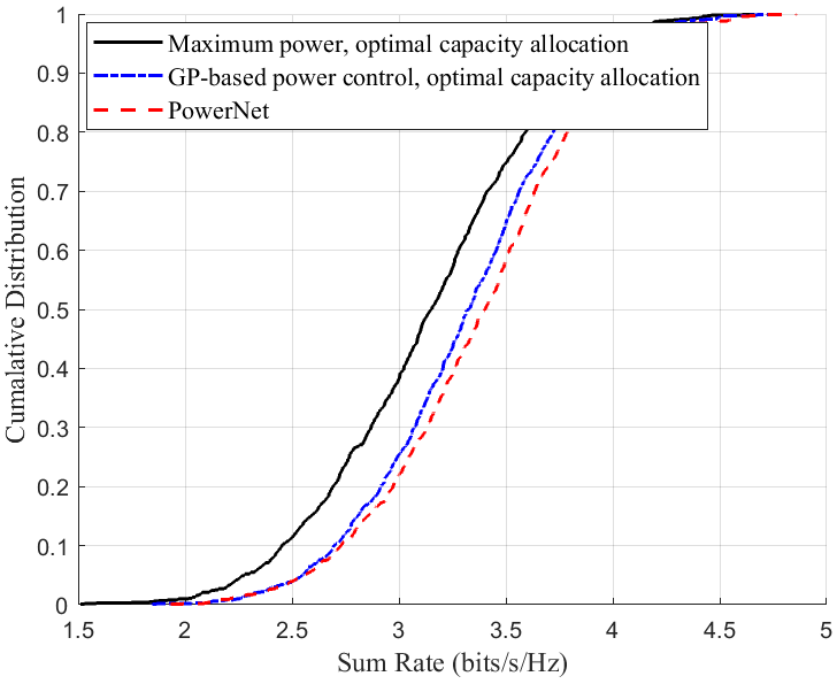


Figure 4-2: Cumulative distribution of the sum rate for 50 APs and 5 users in the cell-free massive MIMO network., obtained using the model trained with 10 users.

Furthermore, in this work, we consider a different RRM problem relevant for cell-free massive MIMO networks. We extend the unsupervised learning concept in [RSR+21] for the pilot and data power control problem in uplink of a cell-free massive MIMO network. In cell-free massive MIMO, pilot

contamination resulting from using non-orthogonal pilots by the users in the network degrades the system performance, which could be reduced by proper pilot assignment and pilot power control. The system performance could be further increased by proper power control for data transmission. Therefore, in this work, we consider the joint pilot and data power control problem to maximise system sum rate, to which finding optimal solutions is difficult due to the non-convexity. We utilise a ML approach to solve the joint optimisation problem in a less computationally complex data-driven manner. A DNN is trained to directly optimise over the sum rate objective as a custom loss function producing pilot and data power outputs. The aim is to reduce the computational complexity of the resource allocation task, while achieving comparable performance with respect to the optimisation-based algorithm. The proposed unsupervised learning approach is illustrated in Figure 4-3. Then, Figure 4-4 shows the results of the proposed method in comparison to fixed power allocations for pilots and uplink data transmission. It is visible that power control improves the system sum rate compared to the fixed power allocation.

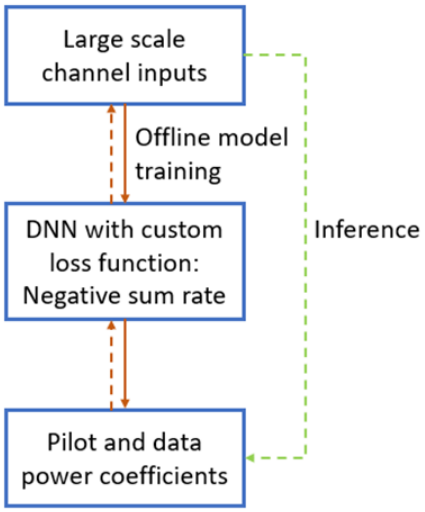


Figure 4-3: Unsupervised learning approach for solving the resource allocation problem.

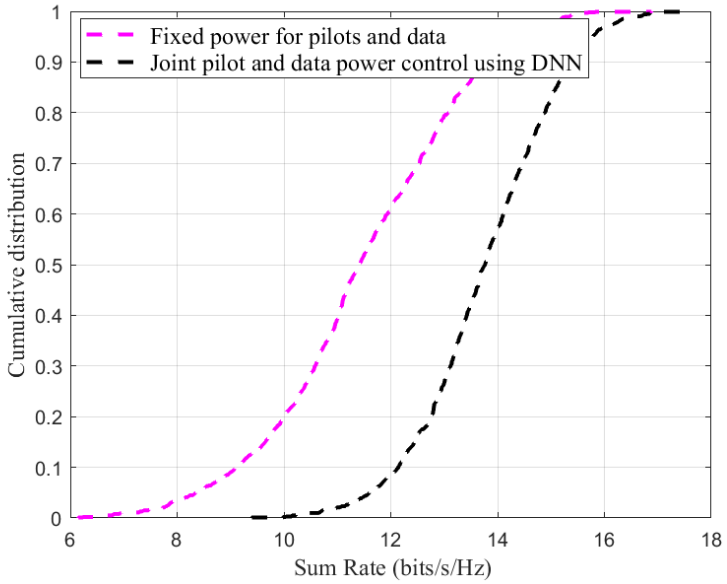


Figure 4-4: Sum rate performance comparison for fixed power transmission and pilot and data power control using the proposed unsupervised learning approach.

The proposed ML-based resource allocation method would be applicable to "Interactive and cooperative mobile robots" use case where new cell-free massive MIMO architectures could be used to manage a

cluster of drones over a 6G network along with novel ML/AI-based resource management, link adaptation and AP selection algorithms for improved performance. The relevant target for this enabler is T2 - complexity gain which targets on reducing the processing time/number of operations in the relevant problem scenario where other problem-specific metrics (spectral efficiency/ BER etc.) are achieved.

4.2 ML-based channel estimation for RIS-assisted systems with mobility

AI-assisted V2X is a service discussed in D1.3, where focus is to enhance automotive services provided by future 6G networks. It can facilitate many applications such as autonomous driving and improvements of safety and comfort applications. All these applications depend on wireless communication technologies to provide connectivity between vehicles and various other devices. However, the connectivity is challenged by the inherent randomness in the wireless propagation environment. Recently, reconfigurable intelligent surfaces (RISs) have been introduced into the wireless communication landscape to control the wireless propagation environment with software-controlled reflections [WZ20]. However, there are many challenges that we need to overcome to facilitate an RIS aided vehicular network (here we focus on an urban vehicular environment). One of the challenges is the channel estimation in an RIS aided network. Since RIS is a passive device, the channel estimation is performed using pilot signals received at the base station (BS), which requires more resources as an RIS consists of a large number of reflecting elements. The fast-changing wireless channel due to mobility requires frequent channel estimation, which results in prohibitive overhead for RIS aided systems. In Section 2.1.1, end-to-end optimisation of an RIS assisted system was considered, whereas here the focus is specifically on improving the channel estimation accuracy under mobility.

In this work we propose a ML-based channel estimation scheme for RIS-aided systems, while considering mobility. We consider the uplink channel estimation of a mmWave vehicular network consisting of a BS with M antennas and a single antenna vehicular user, assisted by an RIS with N reflecting elements. Reflecting elements are arranged in G uniform groups to reduce the pilot overhead [DMR+21]. As illustrated in Figure 4-5, the channel consists of a direct link which is non-line-of-sight (NLoS), and a reflected link through RIS which is assumed to be consisting of line-of-sight (LoS) components. The sparsity of the mmWave scattering channel is used to represent the system in a compact notation which consists of the angle of arrivals (AoA) and Doppler shifts for different paths as parameters. The direct channel and the cascaded channel through RIS can be calculated separately using the estimated parameters.

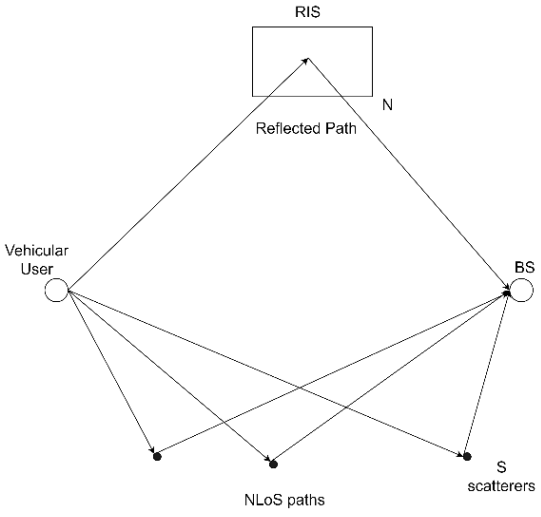


Figure 4-5: Scattering channel model of the RIS-aided system.

In order to estimate channel parameters, first the AoAs are estimated using a neural network, followed by complex path gain and Doppler shift estimation. The NN used for prediction of AoAs is shown in Figure 4-6, where received, pilot symbols are used as an input to the AoA prediction network, with real and imaginary parts are stacked together. At the output, two output layers with *sigmoid* and *tanh* activations are used. Output layer with sigmoid activation predicts the discrete grid point in the AoA grid, and the output layer with tanh activation predicts the residual error. Once the AoA are calculated, the problem reduces to complex path gain and Doppler shifts estimation, and no longer sparse since inactive paths can be just dropped.

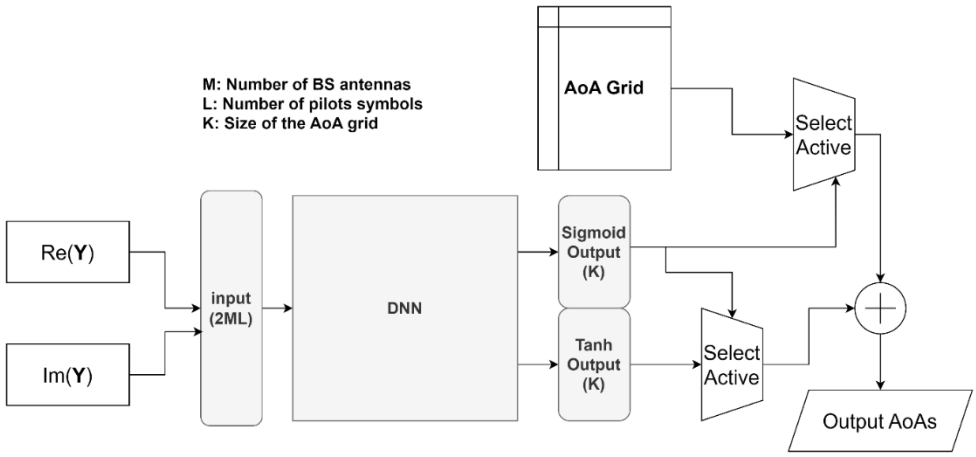


Figure 4-6: Neural network for predicting AoAs.

Training loss is shown in Figure 4-7. We can see that the AoA prediction improves with training epochs, while overfitting is avoided by stopping the training at a suitably early point.

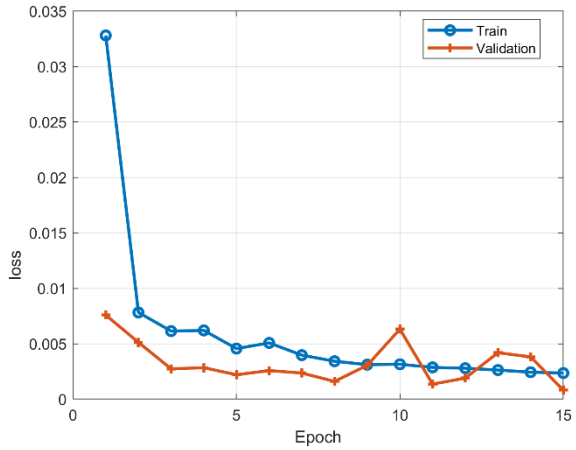


Figure 4-7: Training and validation accuracy of AoA prediction.

Preliminary results for the channel estimation procedure are shown in Figure 4-8, where the channel estimation accuracy (i.e., NMSE, normalized mean square error) is compared for known and predicted AoA cases, for both direct and RIS channels.

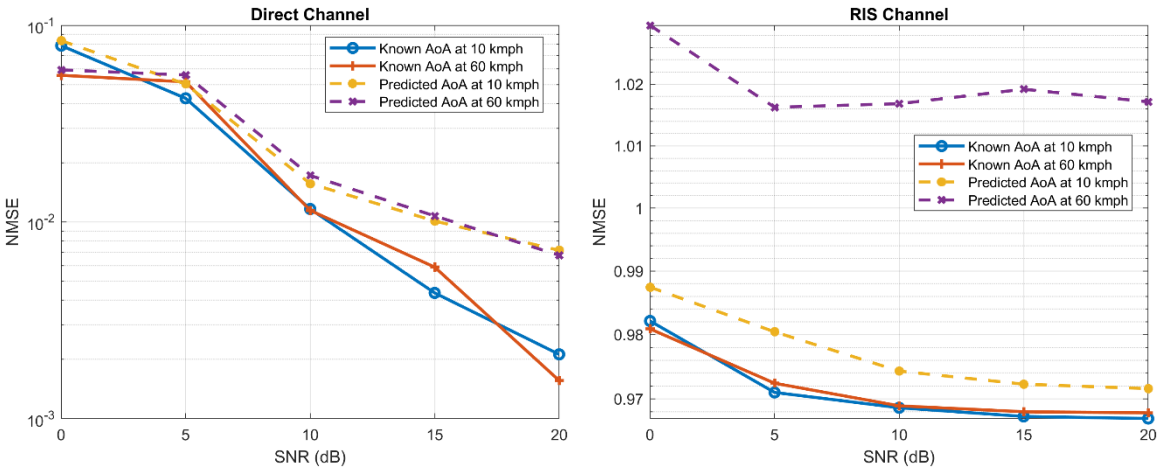


Figure 4-8: Variation of NMSE with SNR (dB) for direct and RIS channels.

4.3 Generalizable low complexity channel estimation using neural networks

A neural network (NN) based solution requires solid validation and proof before it is considered for as an option to replace signal processing solutions. In the case of channel estimation, generalizability with respect to different channel models is critical. Therefore, we dedicate the following part of our research on investigating the behaviour of our solution with respect to unknown channel models.

As a brief overview, our solution is inspired by the MMSE estimator for the linear problem of noisy pilot observations: $y = h + n$. We propose an NN, which its formulation resembles the MMSE estimator, i.e. $\widehat{\boldsymbol{h}} = (C + \Sigma)^{-1} y$, where $\boldsymbol{C}, \Sigma$ are the channel and noise covariance matrix. For wideband channel estimation, such as Sounding Reference Signal (SRS), $\boldsymbol{y}$ has multiple dimensions with different physical meaning: frequency, time, spatial (number of antenna elements). This can increase the dimensions of $\boldsymbol{y}$ and thus the resulting NN grows exponentially with it**.** We can use the Kronecker approximation model for the covariance matrix, which is

$$C \approx C_f \otimes C_t \otimes C_s,$$

where, $\boldsymbol{C_f}, \boldsymbol{C_t}, and\ \boldsymbol{C_s}$ are the frequency, time, and spatial domain covariance matrices. Using this approximation, we can treat the problem as three smaller problems and further even decompose the spatial domain as vertical and horizontal domain. This allows for smaller NNs that can be even trained separately, which roughly translates to smaller training sample complexity. This process is depicted in Figure 4-9.

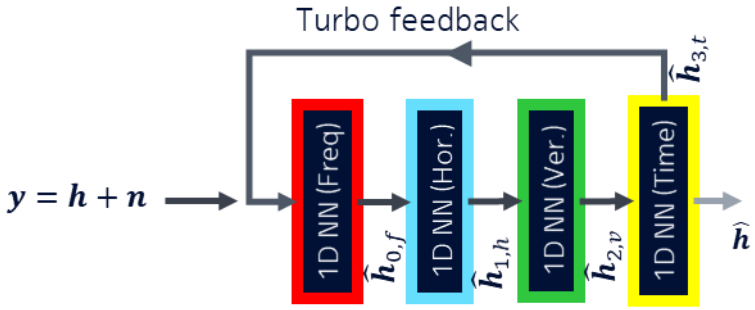


Figure 4-9: Turbo-AI procedure. It is consisting of 4 different smaller independently trainable NNs and a feedback loop.

We define generalizability of an NN as its performance on a dataset sample from a different probability distribution that the one which is used for training and testing of the NN. We train the NN with cluster delay line models, CDL-A, CDL-D, and CDL-E and test it over dataset produced from CDL-B and CDL-C model. This experiment allows us to understand the frequency in which a deployed NN requires retraining.

In Figure 4-10 we demonstrate the generalizability of Turbo-AI [CMW+21a] on the trained channel distributions. We train the Turbo-AI on samples generated according to 3GPP numerology of channel models CDL-A, CDL-D and CDL-E. Then we test the performance by generating test datasets according to the same channel models. As it is observed in Figure 4-10 the performance is within ~3dB of the statistically optimal solution, i.e. MMSE.

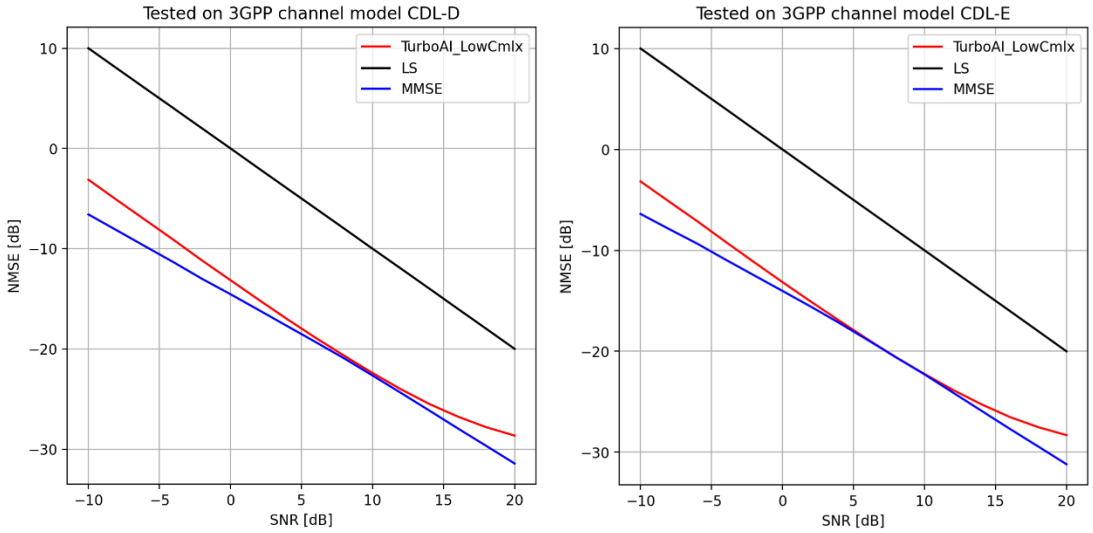


Figure 4-10: The comparison of Turbo-AI, Least squares (LS) , and MMSE are presented for the channel models CDL-D and CDL-E. The training data for Turbo-AI is also produced from mixed channel models of CDL-D & -E.

In practice, the training dataset and the onsite test pilots differ in distribution. This model gap mismatch

can be negligible if we retrain our model using the onsite data. However, this is rather cumbersome due to many issues including online label production and the complexity of online training. To measure how well our channel estimator is prepared for channel model probability deviation, we test our model on

the dataset generated from distributions that are fundamentally different from our training dataset. This is beyond the common definition "generalizability". To this end, we use the same model (trained merely

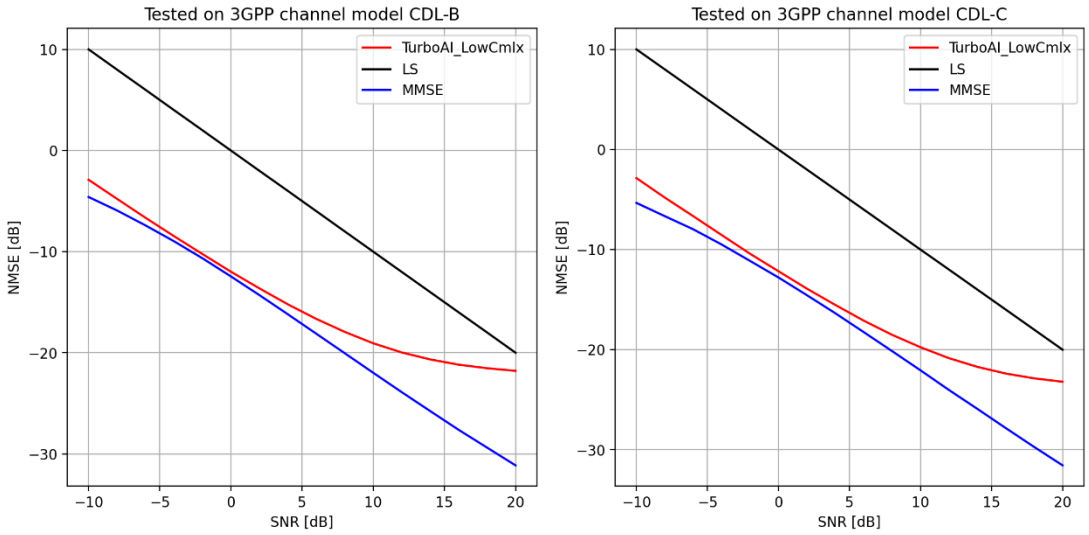


Figure 4-11: The NN is trained on CDL-A,-D & -E, while tested on dataset generated according to CDL-B and CDL-C, to show TurboAI's generalizability beyond the definition. Obviously, the performance degradation appears when compared to the previous case in Figure 4-10.

on CDL-A,D, & E) to test on datasets generated according to CDL-B and CDL-C standards. The results are illustrated in Figure 4-11. The performance degradation at higher SNRs could be due to model distribution mismatch.

To compare the complexity of Turbo-AI [CMW+21b] with the state of the art [NWU18], we assume that the input channel tensor is of $\mathbf{y} \in \mathbb{C}^{N \times M \times T \times F}$, where N and M are dimensions of the antenna panel, T is the number of time samples and F is the number of subcarriers. In [NWU18] the computational complexity becomes $O((MNTF)^4)$. The 4D Turbo-AI's complexity for a similar performance is $O(NTFM^4 + MTFN^4 + MNTF^4 + MNFT^4)$.

4.4 Deep unfolding for efficient channel estimation

Channel estimation is a key step in any communication chain. It is also a challenging one, especially with the advent of massive MIMO systems and their induced complexity. In [HEX-D42], a deep unfolding approach for channel estimation [YL22] is explored in the context of MIMO channels with a single subcarrier. Building on that, we explore the use of the same technique for SISO-OFDM channels with multiple subcarriers.

In particular, the proposed approach focuses on physical prior information about the system. In real case scenarios, this knowledge is often imperfect due to calibration errors and hardware imperfections. In the case of SISO-OFDM systems, these errors could result from the frequency generation/acquisition step and give rise to Carrier Frequency Offset (CFO) and/or Sampling Clock Offset (SCO) phenomena. With this in mind, the proposed approach tries to correct these errors and relies on deep unfolding [MLE21] to do that. Deep unfolding is a technique which considers an iterative algorithm as a NN by effectively "unfolding" it, meaning that each iteration is transposed into a layer with trainable weights. The depth of the resulting NN thus corresponds to the number of iterations of the original algorithm. Its parameters are then 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.

Similarly to the work presented in [HEX-D42], the proposed NN, called mpNet and adapted for SISO-OFDM channels, takes as input noisy channel vectors denoted $\boldsymbol{x}$ resulting from the least-squares channel estimation. The weight matrix $\boldsymbol{W}$ of the network is initialized with a set of frequency response vectors (called a dictionary) according to the available physical model. The depth of the NN corresponds to the number of matching pursuit iterations that are unfolded and is allowed to vary to be SNR-adaptive. Finally, the network is trained online in an unsupervised way, with a cost function of the form $\frac{1}{2}\left\|\boldsymbol{x}-\widehat{\boldsymbol{h}}\right\|_2^2$.

To reduce the complexity of the model, two novel ideas are introduced, namely constrained dictionary and hierarchical search. In a nutshell, a constrained dictionary is a dictionary of frequency response vectors where only the parameters producing each vector (i.e., complex antenna gains and SCO frequency offset) are allowed to be learned, as opposed to learning every entry of the dictionary. On the other hand, hierarchical search is a way of finding the most correlated atom in the dictionary in a hierarchical way instead of the classical exhaustive way, reducing the number of operations to be carried out. More details about both ideas are given in [CLR22].

All in all, this work aims at providing a frugal channel estimation algorithm with reduced computational complexity when compared to generic deep learning-based approaches. In order to evaluate the performance of the model, we rely on the normalized channel estimation error (NMSE) as the main KPI and defined as

$$\mathrm{NMSE}=\frac{\left\|\widehat{\boldsymbol{h}}-\boldsymbol{h}\right\|_2^2}{\|\boldsymbol{h}\|_2^2}.$$

We consider the "O1" outdoor scenario of the DeepMIMO dataset [ALK19]. Transmission occurs over channels with a central frequency of 3.4GHz, a bandwidth of 50MHz and 256 subcarriers. The results are shown on Figure 4-12. The blue curve corresponds to mpNet initialized with a dictionary based on an imperfect knowledge of the underlying physical parameters (nominal dictionary), the orange one corresponds to using least squares estimation, the green one corresponds to using a physical model with the same imperfect knowledge as mpNet, and lastly the red one corresponds to using a physical model with a perfect knowledge of the physical parameters (ideal dictionary). Observe how mpNet quickly learns and perfects itself, and goes from performance equal to what the imperfect physical model achieves, to performance very close to what could be achieved had a perfect dictionary been available. Training is very fast as only approximately 400 channels are needed for the model to converge.

More details can be found on the associated paper [CLR22].

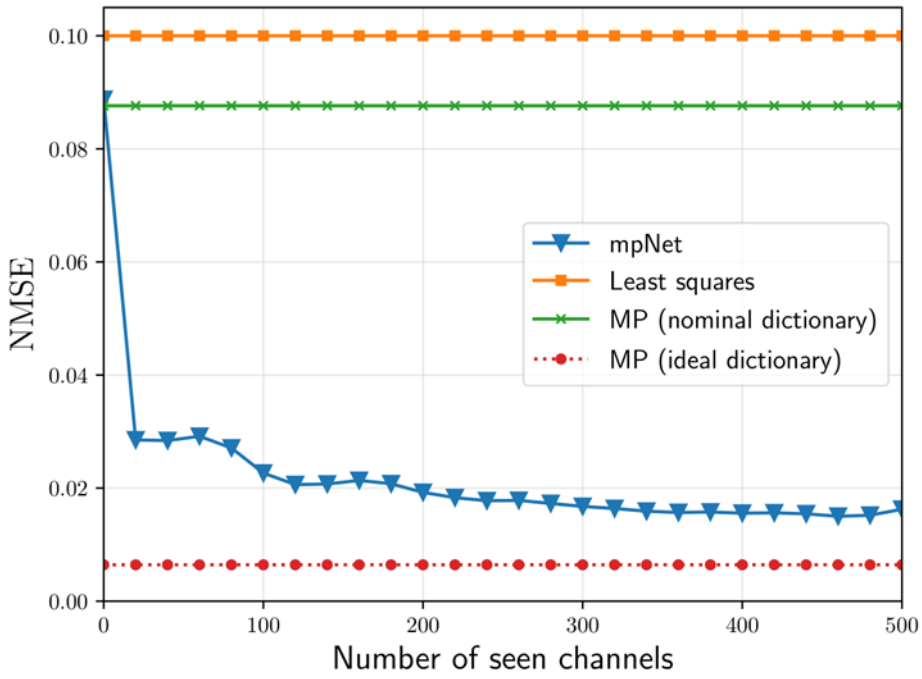


Figure 4-12: Comparison of mpNet to baselines throughout its training.

4.5 Hybrid model for channel charting

Channel charting is an unsupervised learning method that aims at mapping wireless channels to a so-called chart, preserving as much as possible spatial neighborhoods. It falls within the realm of machine learning in general, and dimensionality reduction in particular. It aims at projecting high-dimensional channel observations into a low-dimensional space, typically of 2 or 3 dimensions, in order to learn a channel chart. In fact, physical channel models indicate that channel observations are subject to the manifold hypothesis, meaning that although their original space is of high dimension, they are, in reality, governed by a small set of parameters. Those parameters are directly related to the spatial locations, where the corresponding signals originate from. In this sense, a successfully learned chart is a map between channel measurements and low-dimensional representations that preserves the local geometry of the original transmit locations.

In order to conceive a model capable of learning the charting function, we rely on deep learning methods for what they offer in terms of capacity and fast adaptation to data. In particular, we adopt the philosophy of model-based deep learning, where neural network structures are guided by physical principles yielding hybrid models more capable than generic multilayer perceptrons (MLPs) for a given task.

The model relies on 3 components:

- **Distance measure:**

  The first step in our approach is to compute the distance matrix of the collected channel vectors. The adopted distance measure needs to correctly reflect the local spatial neighborhoods of channel observations. The traditional Euclidean distance is inadequate to the task because of its

unusual behavior in high-dimensional spaces. A more adequate distance measure is the one introduced in [LEM21]. It is defined as follows:

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

Hybrid encoder:

We begin by selecting a small subset of N_{init} collected channel observations that we organize in a matrix $\boldsymbol{D} \in \mathbb{C}^{M \times N_{\text{init}}}$ as column vectors. The more representative it is of the data distribution the better. We then compute its distance matrix using and feed it to Isomap [TDL00] - a non-linear dimensionality reduction algorithm - to produce an initial channel chart $\boldsymbol{Z}$ according to the method proposed in [LEM21]. For a given new observation $\boldsymbol{h}$ that we would like to project, we compute the modulus of its correlation to each one of the channel vectors in $\boldsymbol{D}$. We keep the k largest elements of the obtained vector using the hard thresholding nonlinear operator denoted $\text{HT}_k(.)$. We normalize the result using an l_1-norm so that the vector elements sum up to 1. Finally, we multiply the normalized vector by the matrix $\boldsymbol{Z}$ which amounts to performing a weighted average of the projections of the k most correlated channel vectors to the input vector $\boldsymbol{h}$.

This strategy makes it possible to project individual channel observations independently of the rest of the dataset. Besides being initialized from a small dataset in conjunction with Isomap, it

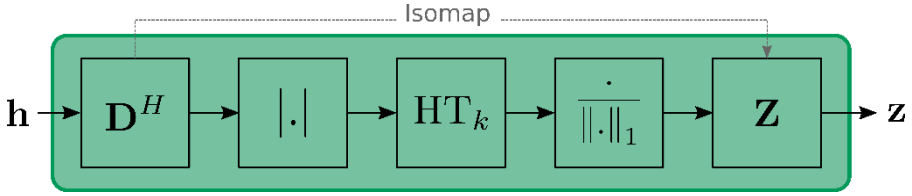


Figure 4-13: Hybrid encoder.

comprises two matrix multiplications separated by nonlinear operations, which makes it easily laid out as a neural network as depicted in Figure 4-13:.

- **Triplet loss:**

The main advantage of using neural networks is the ability to train and improve their performance through gradient descent with backpropagation. The main idea is to calculate a loss function that quantifies the error of the network's output and to iteratively "move" its weights in the opposite direction of the gradient of this loss function as to minimize it. In a supervised learning setting, this loss function takes as input both the output of the model and the target value that is ought to be achieved, hence the need of a training dataset of target values. However, in the context of channel charting, the objective is to rely solely on the channel observations themselves. As a consequence, it is necessary to make use of unsupervised learning methods. In particular, contrastive learning has been used extensively for various tasks. Simply put, it aims at teaching the network which samples are similar and which are different with the help of a specifically designed loss function L . Triplet networks (Figure 4-14:) are an implementation of this framework, where triplets of three samples each are constructed and fed to a neural network. A single triplet comprises an anchor sample $\boldsymbol{h}$, a close sample $\boldsymbol{h}^+$ and a far sample $\boldsymbol{h}^-$. The network produces their corresponding projections into the channel chart $\boldsymbol{z}$, $\boldsymbol{z}^+$ and $\boldsymbol{z}^-$ respectively. The loss function is defined as

$$L(\boldsymbol{z}, \boldsymbol{z}^+, \boldsymbol{z}^-) = \max(0, d^+ - d^- + m)$$

where $d^+ = \|\boldsymbol{z} - \boldsymbol{z}^+\|_2$, $d^- = \|\boldsymbol{z} - \boldsymbol{z}^-\|_2$ and m is a margin parameter. In essence, minimizing L amounts to maximizing the difference between d^+ and d^- by pulling the close sample closer and pushing the far sample farther until the difference is greater that m and therefore $L = 0$. Instead of using a generic MLP, we propose to use the hybrid encoder presented in above initialized with a small subset of channel observations following the proposed strategy. The parameters (i.e. weights) of the network $\boldsymbol{\theta} = \boldsymbol{D}, \boldsymbol{Z}$ are then optimized through the training of the encoder in the triplet configuration.

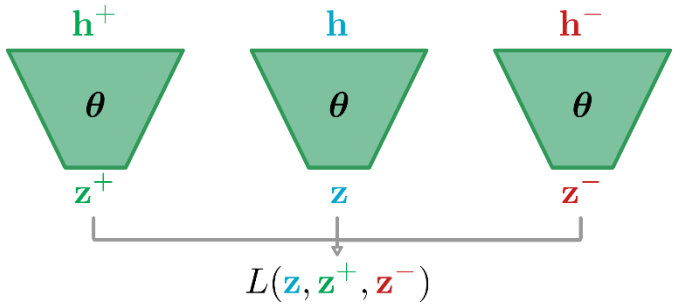


Figure 4-14: Triplet network structure. $\boldsymbol{\theta}$ is the set of parameters (i.e. weights) that the neural network learns throughout training.

It is important to note that constructing triplets obviously requires knowledge of positive and negative samples for a given anchor sample. In the unsupervised setting, this knowledge has to come from the dataset of inputs itself. In fact, the time-correlated nature of channel observations can be exploited. Indeed, considering two observations close in time to correspond, with high probability, to transmit locations spatially close to each other is a reasonable assumption. Since each channel's timestamp is readily available, it is straightforward to determine which channels are close together. From this knowledge, constructing a dataset remains a matter of determining, for each channel observation, a close (positive) sample and a far (negative) sample based on a temporal threshold.

A channel chart on its own doesn't serve much purpose, but it could be leveraged as a starting point to improve many downstream tasks where location knowledge is key, such as user localization, beam management, resource allocation, etc. For example, in section 3.1.6, a learned chart is used as the input to a location based beamforming model in order to predict the optimal precoder for a given user. Consequently, the quality of the chart heavily affects the performance of the subsequent task. However, channel charting being an unsupervised task, it is hard to assess the quality of a learned chart on its own. A combination of metrics, such as trustworthiness (TW) and continuity (CT) is thus used to gain insight on the produced chart. In short, TW measures whether nearby projections on the chart correspond to spatially close users, while CT measures whether nearby users correspond to close projections on the chart. They are measured for a given neighbourhood size K (given as % of the total number of data points N in Figure 4-15) and range from 0 (worst) to 1 (best). See [LEM21] and references therein for a formal definition.

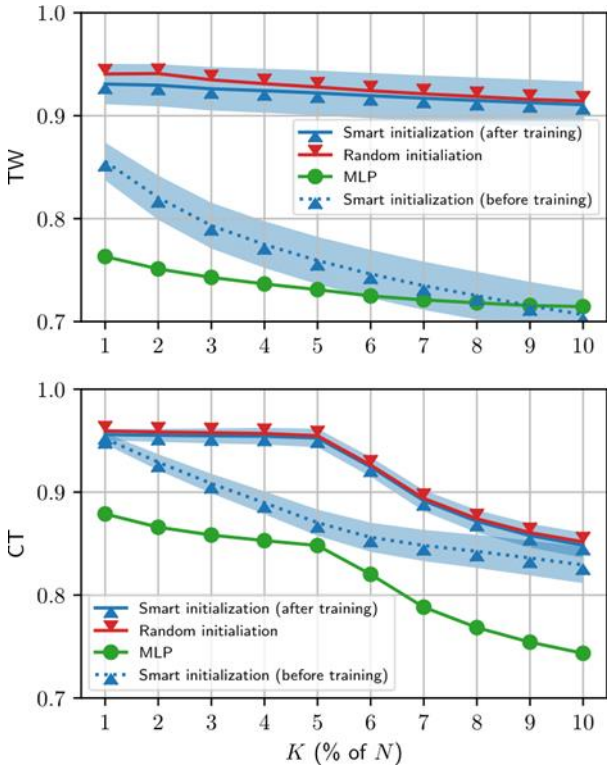


Figure 4-15: Comparison of the models in terms of TW and CT.

We rely on the DeepMIMO dataset ('O1' scenario) [ALK19] to assess the performance of the proposed method. We simulate the trajectory of a pedestrian by collecting samples along a path on the map at a constant speed. We evaluate the performance of the model with regards to three aspects: *initialization*, *structure* and *training*. We compare the contribution of each aspect to the construction of the chart. Results in terms of TW and CT for different neighbourhood sizes K are shown on Figure 4-15.

5 6G network as an efficient AI platform

It is well known and widely accepted that AI and ML will have a twofold role in future networks, both as enablers of flexible system optimisation (AI/ML for networks) and as a service efficiently enabled by 6G (networks for AI/ML). This chapter focuses on the second paradigm, with contributions exploring the tight integration of computing capabilities in wireless networks, thus calling for new optimisation strategies that involve heterogeneous resources (e.g., communication and computing). The integration of communication and computing capabilities at the edge opens the way for enabling several use cases, which require the transmission of data, and the management of workload in edge computing resources. Among the ones identified by the Hexa-X vision, the technical solutions proposed in this section cover several aspects. First, a use case of interest is AI-assisted vehicle-to-everything, with KPIs including AI agent availability, AI agent reliability, and mobility support, as well as energy consumption. Additionally, focusing on network impairment resilience, massive twinning and from robots to cobots use cases (including AI partners) are considered, with KPIs including latency, safety, maintainability, and security. Going beyond, and covering distributed scenarios, load balancing in federated learning settings is foreseen to further target KPIs in terms of latency, AI agent availability, and energy efficiency, thus also covering enabling sustainability use cases. Further technical solutions

on frugal federated learning are proposed to customise FL training to the available resources of the different devices, which can be very different in general. In this case, still envisioning use cases such as the interacting and cooperative mobile robots one, inference accuracy, latency, and E2E energy efficiency are the main target KPIs. Also focusing on distributed settings, scalable and resilient deployment of distributed AI is envisioned to address interacting and cooperative mobile robot use cases, in particular for real-time decision making through resource efficient data sharing. The latter also covers hyperconnected resilient network infrastructure, with KPIs including privacy and complexity. Furthermore, the chapter focuses on the Compute-as-a-Service concept, with flexible workload assignment, targeting low latency. Finally, when optimizing workload assignment, it is also fundamental to consider the joint optimisation of wireless and computing resources. To this end, and with possible implications on cooperative mobile robots, a method for DNN splitting at the edge with end device energy frugality and controlled latency is proposed, while the concept of communication-computation co-design is further extended to the arising paradigm of goal-oriented communications for edge inference, thus considering inference accuracy as additional KPI. However, going further, optimal workload placement is also exploited to minimise energy and traffic, and to maximise trust level, thus again addressing sustainability targets. Overall, the solutions proposed in this chapter can contribute to the sustainability key value, with key value indicators including energy consumption during operation, for both communication and computing, and overhead.

5.1 Network services and data structures for AI applications

Edge AI/ML workload management is the first key technical enabler for the Communication to Learn paradigm, which targets the KPIs of energy efficiency and E2E application delay, by accounting for both communication and computing components in the processing chain. These performance metrics contribute to the KVIs of Sustainability. AI agent availability and inferencing accuracy are also shown to be supported in high-mobility environments involving safety-critical communications. In AI workload placement, multiple KPIs are considered, including AI agent availability, network energy efficiency (by targeting reduced energy consumption), as well as trustworthiness (by prioritizing trustworthy physical nodes).

In Section 5.1.1, we describe our protocol framework for AI-as-a-Service (AIaaS) by providing a set of data structures to support scenarios calling for frequent inferencing-based decisions. In Section 5.1.2, we describe the Compute-as-a-Service (CaaS) concept that allows to make a multitude of nodes with heterogeneous computational capabilities available to other users. Finally, in Section 5.1.3, we address the close to optimal placement of AI workloads to the various physical network nodes by minimising the energy consumption of the overall network towards sustainability, minimising traffic, and maximising the trust level.

5.1.1 AIaaS - seamless exploitation of network knowledge

In this section, following the overall initial solution proposal documented in [Section 3.1.1, HEX-D42] and a proposed protocol framework for AI-as-a-Service (AIaaS) in [Section 6.3.2, HEX-D51], we provide more details and propose a set of data structures to support a proposed method, applicable to scenarios calling for frequent inferencing-based decisions, that allows a User Equipment (UE, or other equipment/ machine) to: (i) define and communicate a specific problem statement (e.g., inferencing task) and its performance requirements to the network exploiting the AI Service (AIS) and accompanied Learning Application Programming Interface (API); (ii) receive the connection endpoint details of the available and relevant (with regards to the focused inferencing task) AI agents it can attach to - such connectivity will abide by performance requirements set by the AIS consumer (e.g., the requesting UE) and, (iii) either receive model update notifications from the AI agent it is subscribed to in order to then conduct inferencing locally at the device (e.g., in case the inferencing input data set is large) or post new data to the data management entity of the AI agent the UE is subscribed to. Consequently, by subscribing to the appropriate in-network AI agent(s), the UE (or other equipment) is able to fully

exploit the knowledge of a large part of the network, without individually communicating with each and every AI agent, as this is a functionality of the AIS service.

A typical example relates to prediction of detailed communication parameters in a mobility scenario – far exceeding the information provided by existing navigation tools. Let us assume that a vehicle plans to move from location A (e.g., Munich, Germany) to location B (e.g., Stuttgart, Germany). The vehicle trajectory and journey starting time are a-priori known and considered as input features for inferencing. As an example, the vehicle intends to use the Highway "A8" in Germany as illustrated in Figure 5-1.

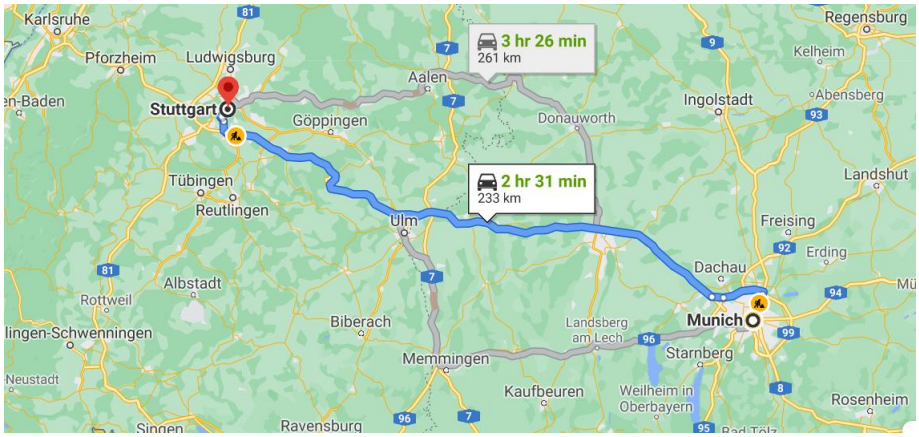


Figure 5-1: Planned vehicle trajectory - client input to an AI agent for issuing journey-relevant recommendations (source: Google maps).

Let us, furthermore, assume that the concerned vehicle requires a local AI/ML model (e.g., a Neural Network - NN, a regression model for classification, a Support Vector Machine - SVM, etc.) providing recommendations on the following tasks en route to the final destination: (i) optimised switch of access technologies (e.g., when to switch from Frequency Range 1 - FR1 to Frequency Range 2 - FR2 access anticipating loss of coverage, when to switch from 3GPP to public WiFi, etc.), (ii) optimised choice of initial communication parameter values when switching access technologies (e.g., optimised Modulation & Coding Scheme - MCS selection, anticipation of antenna beam selection, etc.), (iii) detection of dangerous traffic situations (e.g., obstacles on the road, etc.). Note that this scenario allows for different approaches to managing AI agents: A network based AI agent may provide a specific AI/ML model to a local UE, the network based AI agent may be duplicated in the target UE or a network based AI agent may interact with a UE level AI agent.

5.1.1.1 Relevant data structures for AIS-assisted inferencing

The concerned vehicle (e.g., through a Multi-access Edge Computing - MEC application instantiated at the network's edge corresponding to a client application instantiated at the vehicle) communicates the task description/ problem statement to the AIS with the ultimate goal to obtain a fully trained ML model, as instantiated at one or more available AI agents in the network and customised to address the specific inferencing task with high accuracy. When an AIS consumer requests to the AIS to receive an ML model configuration by an available AI agent, a data structure is contained in the message body of the request (assuming the AI API is a Representational State Transfer - RESTful API) that may include the following information elements (considered as *filtering criteria* of the AIS consumer to be implemented for AI agent selection):

a. Description of a Use Case. Example: We indicate that a user is moving from point A to point B in a geographic area (e.g., by car on a road, by bike on a bike track, walking anywhere, etc.) and we need the ML model to be optimised for this specific trajectory in this specific geographic area.
b. Required input features to the trained (NN, etc.) ML model in the UE. Example: The UE may indicate to the network the input data attributes, e.g., available sensors, e.g., Global Navigation Satellite System (GNSS), video, etc. After obtaining the trained model, the UE will apply these

inferencing inputs to the NN in order to obtain an action recommendation (e.g., change of MCS scheme etc.).

c. Required output features to the (NN, etc.) ML model in the UE. Example: The UE may indicate to the network the required output features, e.g., warning about obstacles, predictive configuration of the modem (best configuration), etc.
d. Required characteristics of the (NN, etc.) ML model in the UE. Example: 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.

Then, the AIS provides a data structure to the AIS consumer (e.g., UE), as part of its response containing the following information elements:

a. (for direct communication between UE and AI agent - need to both follow the same application layer protocol): the AIS provides the connection information of the selected AI agent satisfying the selection filtering criteria provided by the AIS consumer (e.g., the UE). Then, the AIS consumer can subscribe to that specific AI agent. After subscription, the AI agent can provide a data structure consisting of the following exemplary attributes:
    - Type of model, e.g., NN, Tree based estimation, Bayesian estimator, etc.;
    - characteristics of the model, e.g., number of layers of the NN, number of nodes per layer, etc.;
    - number of inputs (plus bits per input data point), number of outputs (plus bits per output);
    - maximum latency to obtain inferencing result, etc.;
    - requirements on inferencing (prediction, estimation etc.) accuracy, trustworthiness etc.
b. (for indirect communication between the UE and the AI agent via the AIS): The AIS provides the latest update of the selected ML model satisfying the AIS consumer's AI agent filtering criteria to the UE. The UE will thus be able to implement a model containing a broader knowledge of the network without exchanging raw data and without reaching out to all available AI agents individually.

5.1.1.2 Data structures relevant to AIS discovery to be used for service interoperability

In terms of AIS discovery by potential consumers (i.e., UE, machine or other) and for interoperability purposes, the network (i.e., a Radio Access Network - RAN, overlaid by a MEC deployment) will advertise the offered AIS to any attached equipment. Since the AIS is assumed aware of the registered AI agents' characteristics (e.g., focused task, I/O inferencing features etc.), as part of service advertisement, it will communicate to potential service consumers the following (empty) data structures:

A data structure describing valid geographic target areas and user movement as well as applicable limitations, for example:

- Eventual limitations of the geographic validity area, for example to a country or parts of a country (for example to the areas where a network operator has deployed its network and thus has knowledge to be shared);
- aaccepted ways of describing the geographic validity area, for example start-point A, end-point B GNSS information, a rectangular shape as a validity area, etc.;
- description of valid trajectories, for example it may be indicated whether certain public highways or other official roads are being used (typically by vehicles); other possibilities include the indication of cycling tracks, pedestrian walkways, hiking trails, off-road tracks, etc. As an extreme case, it may be indicated that the user may move randomly in the assigned geographic area (i.e., the user may not use any official road/path/trail).

A data structure describing the anticipated time period for the movement, for example:

- Exact day and time, for example is it a week-end day, a public holiday, a day during vacation time, a normal working day, etc.;

- concerning the time, it may be indicated a specific period (e.g., anticipated start time is xx:xx hours and anticipated arrival time is yy:yy hours) or a rough indication of the period, e.g., morning, noon, afternoon, evening, night, morning/evening rushhour time, etc.

A data structure describing the inferencing task input attributes, data points of which are locally available to the UE, for example:

- GNSS based positioning information;
- video information (vehicle cameras)
- Light Detection and Ranging (LiDAR) information (vehicle LiDAR);
- radar information (vehicle Radar);
- information obtained wirelessly from neighboring vehicles (co-operative perception based data sharing);
- accelerometer information;
- network based equipment tracking information (e.g., triangulation-based positioning).

A data structure describing the inferencing output features to be locally used by the UE, for example

- "handover" information (e.g., when to switch from 3GPP to public WiFi or vice versa, etc.)
- anticipated best configuration for network switch (e.g., anticipated MCS, anticipated best frequency band (e.g., FR1 or FR2), anticipated best antenna beam configuration, etc.)
- anticipated danger ahead (e.g., obstacle on the road, etc.).

It should be noted that inferencing requests related to new applications' tasks, so far unknown to the network (the AIS), may create the need to specify new data structures. However, in general, communication of invalid data structures by the AIS consumer (or incorrect parameters of known data structures) will trigger a "400 Bad Request" response message in case the AI API is structured as a RESTful API based on Hypertext Transfer Protocol Secure (HTTPS) requests.

The proposed AIaaS concept is envisioned to apply to high-mobility environments involving safety-critical communications; an exemplary use case is the one of AI-assisted Vehicle-to-Everything (V2X) [Section 2.1.4, HEX-D13]. 6G KPIs of relevance are the ones of AI agent availability, AI agent reliability and mobility support. The proposed solution is expected to enhance system robustness to mobility events and concurrently reduce energy consumption as compared to peer-to-peer AI agent discovery.

Overall, the AIaaS paradigm is addressing multiple objectives of future 6G systems including i) it leads to an increased availability by improved mobility solutions, ii) an increased reliability by accounting for low-quality connections, iii) mitigating latency due to radio handovers, association to different AI agents in mobility events, iv) AI models can rely on a substantial information pool, possibly the entire knowledge of the NW. This substantially improves inferencing accuracy, v) a model is requested to be derived by the NW which is substantially reducing power consumption in the UE and vi) the knowledge of the NW can be exploited through providing AI models to the UE without exposing the actual underlying information. This protects the information source.

5.1.2 Flexible compute workload assignment, CaaS

A future generation communications system is expected to be comprised of a multitude of nodes with heterogeneous computational capabilities, including computational elements such as general purpose microprocessors, digital signal processors, field programmable gate arrays, etc. The Compute-as-a-Service (CaaS) concept allows to make such computational elements available to other users. A related API is proposed to provide the following services, as it was originally discussed in [HEX-D42]:

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. Typically, one of the following code types is applied in this case:

- Executable Code: The code is optimised and compiled for a specific target platform. For any additional target platform, additional optimisation steps may need to be performed.
- Source Code: The source code is provided to the operator of the target platform and needs to be processed further. In case of a general-purpose compute platform, the compilation may be straightforward. In case of a heterogeneous platform, typically consisting of different computational element of different kind (such as general-purpose microprocessors, digital signal processors, field programmable gate arrays, etc.), typically the Virtual Machine (VM) based approach is preferred. Such a platform requires a suitable allocation of code components and corresponding optimisation steps.

2. 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. Thus, code developed for such a unified architectureis then mapped onto the target platform by the SP.

Depending on the target scenario and inherent platform characteristics (such as highly heterogeneous platforms employing distinct types of computational elements versus homogeneous platforms relying on a single family of microprocessors), the most suitable Code type should be selected. We now consider the optimum code type selection for the Hexa-X (updated) Use Case Families (UCFs) and individual use cases thereof as summarised in [Figure 2-2, HEX-D13]. An analysis for suitable mapping of Hexa-X UCFs to specific types of code is given in Table 5-1 below:

Hexa-X Use Case Family	Proposed suitable Code type	Comments
Telepresence	Executable Code	Telepresence products are expected to be highly optimized mass market solutions. Also, the variety of products (by a specific vendor) is expected to be limited. It thus seems preferable to provide highly optimized executable code.
Robots to Cobots	Source Code or ConfigCode	In the context of Roborts, Cobots and Industrial IoT in general, performance targets may change over time and for different locations a robot/ cobot may visit as part of its task (e.g., packet pick-and-place). Therefore, it is expected that a multitude of distinct platforms will be used during the lifecycle of a robot task calling for workload offloading. Also, various platforms typically consist of very specific designs providing diverse components (general purpose microprocessors, Field-Programmable Gate Arrays - FPGAs, Digital Signal Processors - DSPs, etc.). It is, therefore, preferable to use highly portable processing task instruction code as provided by Source Code or ConfigCodes optimised for a VM approach. Regarding this UCF, we would highlight the "Situation-aware device reconfiguration" use case calling for platform flexibility in workload offloading, as a device may repurpose itself when located at different deployment instances (e.g., on the road, then inside a factory floor etc.).
Massive Twinning	Source Code or Executable Code	Massive Twinning applications relying on a digital representation of a real-world environment (the recent, so-called Digital Twin or Metaverse concepts) are expected to be executed on large scale platforms, such as data centers or similar. Generic Source Code (typically not relying on some specific hardware) may thus be a preferred Code type that can be compiled for the target compute platform (typically relying on General Purpose Microprocessors as used in Data Centers). In case that some specialized hardware is being used, optimized executable code (or ConfigCodes) can be used, depending on the case.
Trusted embedded networks & Hyperconnected resilient network infrastructures	Source Code or ConfigCode or Executable Code	There is a broad variety of products to be deployed in local trust zones for human and machines, ranging from low-cost sensors to localised (secure) networks and similar. In this field, the specific product family needs to considered and the most efficient Code type is selected for the specific target.
Enabling sustainability	Source Code or ConfigCode or Executable Code	Sustainable development consists of a multitude of specific Use Cases, including E-Health for all, Earth monitor, etc. which all rely on platforms serving very distinct needs. The most efficient Code type is selected for the specific target.

Table 5-1: Preferred Code Types for Hexa-X Use Cases.

Typically, it is the system manufacturer, integrator or any other suitable organization is finally coordinating that the code is transported to the target platform through a so-called “Radio Application Package (RAP)”:

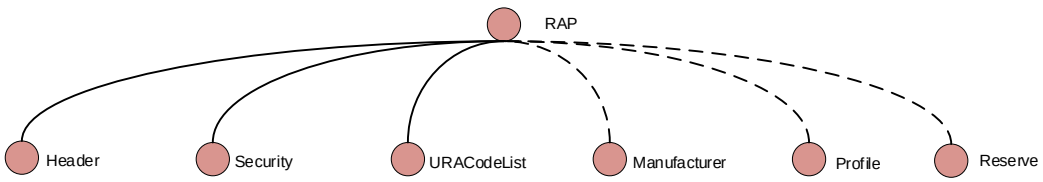


Figure 5-2: Top Level tree structure as defined by ETSI TS 103 850 [103 850]

As further detailed in Annex A, we propose an extension of the “Reserve” field (as introduced in Figure 5-2) in order to accommodate for two additional requirements in the context of Hexa-X applications:

1) It is proposed to indicate the suitability of the RAP for usage in specific Hexa-X use cases. There are indeed different requirements depending on the target application – for example, code

may require specific conformity checks prior to usage in an industrial environment where any malfunctioning may cause harm to persons.

2) We furthermore consider the specific case that the RAP contains code the is considered to enable a “High Risk” application as defined by the draft AI Act [AIA]. We introduce the propose of RAP containing a specific bit which indicates that the manufacturers guarantees compliance for specific AI “High Risk” categories. Only if the corresponding bit is set, the software component may be used in this specific context.

The proposed framework on flexible compute workload assignment aims to enable intelligent and flexible workload delegation, factoring in 6G use case's requirements as well as end user/ developer capabilities. The proposed solution may be relevant to all Hexa-X 6G UCFs, as described in [HEX-D13].

The CaaS approach thus address multiple objectives of future 6G networks including i) latency can be reduced by choosing computational resources such that delays for forwarding of information are minimized, for example locally co-located computational resources and ii) Flexible compute workload assignment (CaaS) accounting for network and device energy consumption.

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

Several modern information and communication technology systems are gradually including AI enablers, which can potentially have great benefits on economies and society by supporting a more fair, inclusive, and safe decision-making. This decision-making must be operated by nodes in a trustworthy and sustainable manner, at the same time. Therefore, managing the AI operations is crucial, especially in decentralised scenarios. This can be accomplished by designing a novel AI management system, applicable in B5G/6G architectures. This section describes the close to optimal placement of AI workloads to the various network’s physical nodes by minimising the energy consumption of the overall network towards sustainability, minimising traffic, and maximising the trust level. The AI workloads are assumed to be mostly inferencing related. It is assumed that CPUs/GPUs/NPUs account for most to energy consumption of nodes compared to memory, disk storage and bandwidth. It is also assumed for now that each physical node is characterised by a given trust level index and finally for this problem it is taken into consideration transmission and processing delay aiming to minimise their sum.

The input to the optimisation algorithm is the set of AI workloads characterised by their computational requirements (CPU/GPU, RAM, etc.), the data size to be processed by each AI workload, and the location of the data, i.e., the network node where data is produced. Additional input to the algorithm is the set of physical nodes with their capabilities (available CPU/GPU, RAM, etc.), their power consumption when idle and when fully loaded, their trust level index and the network topology graph with the maximum bandwidth capacity of each link.

The objective is the allocation of AI workloads to the available physical nodes by minimising the objective function that satisfies a set of performance constraints. The objective function is the following:

$$\min_{x,z}\left(a_1 \sum_{j=1}^{m}\left((W_{max}^{j} - W_{idle}^{j}) \frac{\sum_{i=1}^{n} cpuAI_i x_{i,j}}{cpuPN_j} + W_{idle}^{j} \right) - a_2 \sum_{i=1}^{n}\sum_{j=1}^{m} trustPN_j x_{i,j} + a_3 \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{j'}^{m} z_{i,j,j'} \left(\frac{ds_i}{cpuPN_{j'}} + \frac{ds_i}{B_{j,j'}} \right) \right)$$

The notation can be found in Table 5-2. The first term of the function is related to power consumption where it is assumed that CPU utilisation rate accounts for most compared to memory, disk storage and bandwidth as was proposed in [AAR22]. CPU is referred for the sake of simplicity, the computing capacity on node can be a function of CPU, GPU, and NPU. The second term is related to trust level

and the last one refers to processing and transmission delay (no processing queues assumed). All terms are normalised and weighted depending on the use case.

The set of constraints utilised are:

- $\sum_{j=1}^{m} x_{i,j} = 1 \; \forall \; i \in \{1, \; \dots, \; n\}$, each AI workload is allocated to only one physical node,
- $\sum_{i=1}^{n} x_{i,j} \; cpuAI_i \leq cpuPN_j \; \forall \; j \in \{1, \; \dots, \; m\}$ and $\sum_{i=1}^{n} x_{i,j} \; memAI_i \leq memPN_j \; \forall \; j \in \{1, \; \dots, \; m\}$, the maximum computational load of each physical node is respected,

$z_{i,j,j'} \leq w_{i,j}, \; z_{i,j,j'} \leq x_{i,j'}, \; z_{i,j,j'} \geq w_{i,j} + x_{i,j'} - 1, \quad \forall \; i \in \{1, \; \dots, \; n\}, \; j, j' \in \{1, \; \dots, \; m\}$ the communication is among physical nodes where at least one has the data needed for the AI workload.

Notation	Definition	Notation	Definition
$n, \; m$	Total number of AI workloads and physical nodes, respectively	$W_{max}^{j}, \; W_{idle}^{j}$	Power consumption when physical node j is fully loaded and idle, respectively
i	Index of AI workloads	$w_{i,j}$	Binary constant (0,1) showing where each AI workload's data is generated
$j, \; j'$	Indexes of physical nodes	$trustPN_j$	Trust level index of physical node j
$cpuAI_i, \; memAI_i$	CPU and memory requirements of AI workloads	$a_1, \; a_2, \; a_3$	Weights to prioritise the energy consumption, trust level or E2E latency depending on the use case
$cpuPN_j, \; memPN_j$	CPU and memory resources of physical nodes	$x_{i,j}$	Binary decision variable depending on whether AI workload i is (is not) assigned to physical node j
ds_i	Data size of AI workload i	$z_{i,j,j'}$	Binary decision variable depending on whether AI workload i is assigned to physical node j' taking the data produced on physical node j
$B_{j,j'}$	Maximum capacity of the links among physical nodes j and j'		

Table 5-2: AI workload placement notations.

The above problem was initially solved with the use of a Mixed Integer Programming (MIP) python solver called Python MIP [ST20]. MIP solvers are known to provide the optimal solution but are computationally intractable, especially for large experimentation. For this reason, a meta-heuristic algorithm was developed building upon the Genetic Algorithm paradigm [MBS+21].

In general, genetic algorithm considers a population of individuals, known as "chromosomes", to encode a solution of a problem each. The "chromosome", in turn, is a series of predetermined number of "genes", and each "gene" stands for a parameter that defines the solution for that individual. In this problem, each "chromosome" is a series of physical nodes, where each one represents the "proposed" physical node for each AI workload and the length of each "chromosome" equals to the number of AI workloads. The algorithm firstly generates randomly a population of "chromosomes" and then a fitness function is applied to each "chromosome". The fitness function corresponds to the objective function described above and provides a fitness score. Over the course of a defined number of generations, a population of chromosomes evolves, and some operators (parent selection, crossover, mutation) are used to improve the population's overall fitness.

Parent selection operator is the process of selecting chromosomes in one generation to pass them to the next generation, these chromosomes are known as "parents". In this case the tournament selection was utilised where each "parent" is the fittest out of a predetermined number of randomly chosen chromosomes of the population. Crossover operator is used for creating two "children" candidate solutions (new solutions) from two "parents". In this solution one random split point is utilised. Finally, mutation is the procedure of a random change in a single gene of a "child's" chromosome or in a group of genes for exploring new areas of the solution space. The number of generations was determined by

a dynamic stopping criterion in order to achieve a satisfactory execution time-convergence trade-off. Crossover and mutation operators occur with a predefined probability (crossover and mutation rate). Additional steps were introduced related to efficient initialization of "chromosomes" among others. Figure 5-3 shows the flowchart of the algorithm.

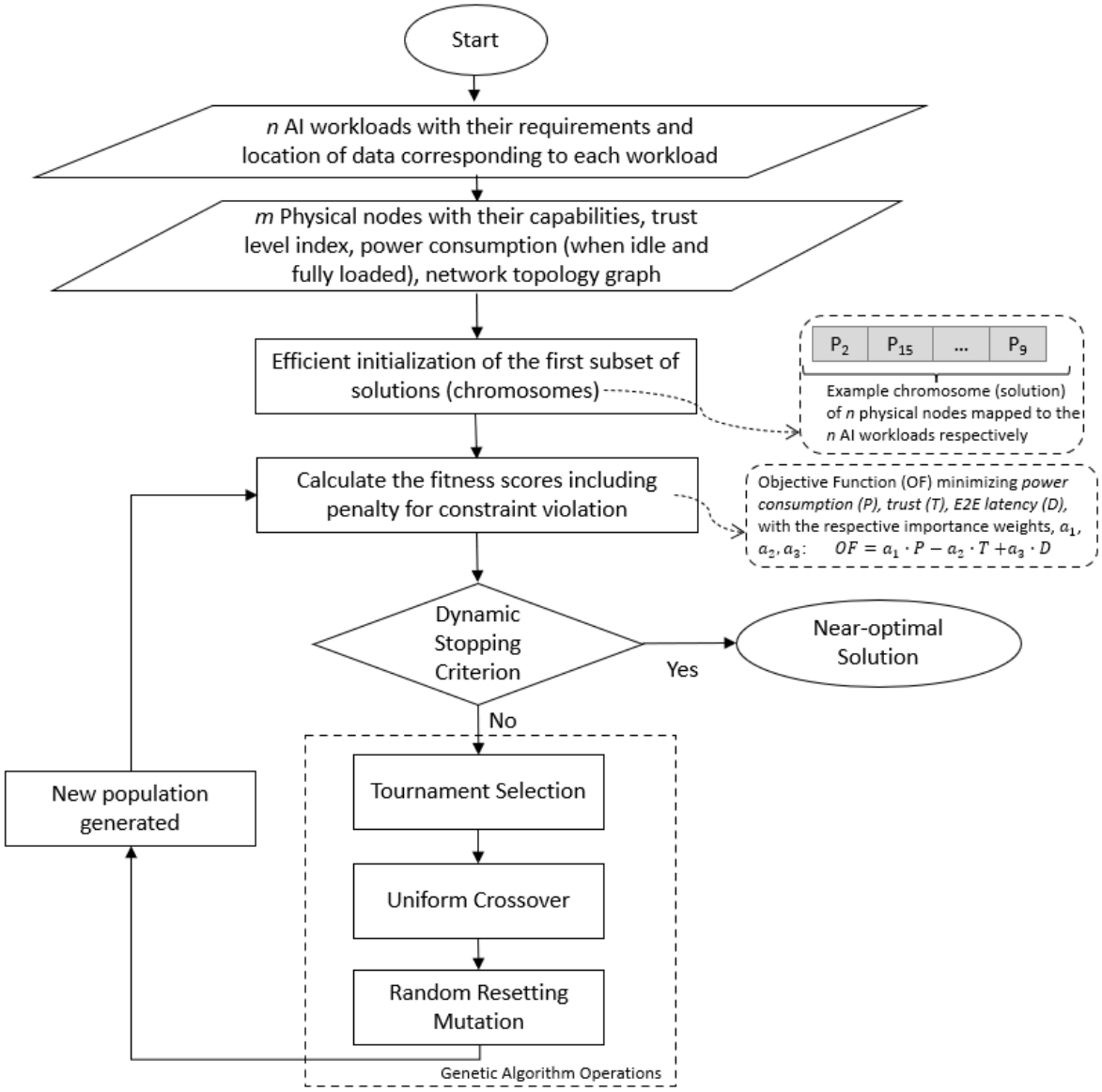


Figure 5-3: Flowchart of AI workload placement algorithm based on genetic algorithm.

Performance testing of the proposed genetic algorithm with the MIP python solver is shown in Figure 5-4. A set of fixed 43 physical nodes is assumed for these measurements with {2000, 2600, 3000} MIPs levels of available CPU, {2048, 4096, 8192} MB levels of available memory, {260, 360, 460} W levels of power consumption when fully loaded and {70, 100, 170} W when idle. The trust level index of these nodes is assumed to be within 0 to 1 range where the most trustful node has trust level of 1. The links between physical nodes have capacity 3.3–- 20 Mbps. The AI workloads have {250, 300} MIPS levels of required CPU, {256, 512} MB levels of required memory, {10, 40} MB levels of data transferred. In these measurements/experiments we used an initial population of 150 chromosomes, 0.8 crossover rate and 0.15 mutation rate.

As it is mentioned above, MIP solvers are computationally intractable in large experimentation, hence an execution time limit of 400 s was applied to the solver. Thus, the solver is terminated when it exceeds this time limit, and the respective best (i.e., lowest value calculated from objective function) score obtained up to that point is selected. The dashed blue line in Figure 5-4 shows the best scores obtained

till that threshold. As it is shown, the proposed genetic algorithm has close to optimum scores within significantly less time than MIP solver as the number of AI workloads increases.

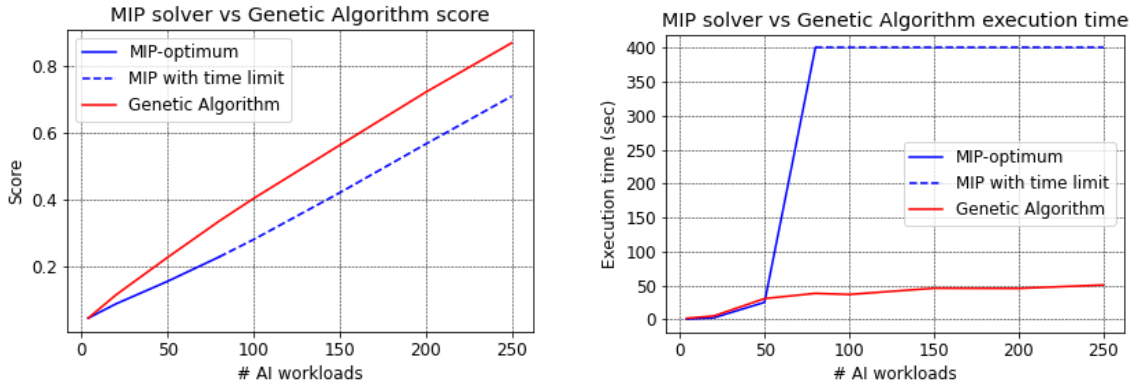


Figure 5-4: Performance testing of proposed genetic algorithm and MIP solver with increasing number of AI workloads (score and time execution measurements).

The described mechanism/algorithm is related to several of the Hexa-X use cases including interacting and cooperative mobile robots, and digital twins for manufacturing. Hence, E2E latency and power consumption was measured for targeting these KPIs. Figure 5-5 shows the reduction of E2E latency and power consumption with increasing number of AI workloads when the number of physical nodes is 43. For these measurements, the baseline model utilised was the random but feasible placement of the AI workloads to the available physical nodes. In each graph there are three curves72odelled72tnding to different levels of a_3 weight for E2E latency and a_1 for power consumption. As the number of AI workloads increases, the E2E latency gains decrease. Similarly, as the number of AI workloads increase, the gains in power consumption increase, until a critical point, at which the physical nodes/number of AI workloads becomes considerably smaller than 1. In other words, the described solution provides higher gains when there is sufficient availability of physical resources, thus solution space and optimisation potential. The scarcer those resources become, the lower gain potential is observed.

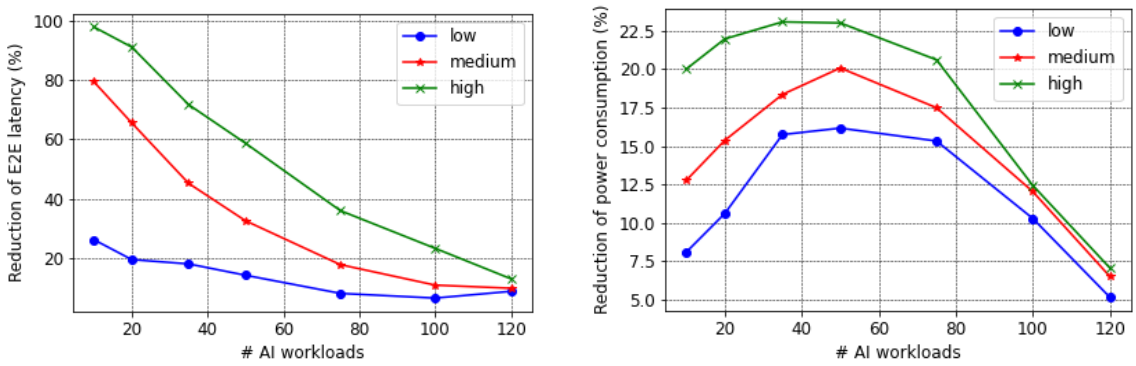


Figure 5-5: Reduction of E2E latency (left) and power consumption (right) with increasing number of AI workloads for different weight levels a_3 and a_1, respectively.

5.2 Efficient inference for distributed AI

In the Communication to Learn paradigm, there is an increasing adoption of intelligent components among higher-layer in-network functions and external applications. For instance, for use cases such as

interacting and cooperative mobile robots or *digital twins in manufacturing* require real-time intelligent decisions based on distributed and resource efficient data and model sharing. These use cases require *hyperconnected resilient network infrastructures*, in which a huge amount of data will be distributed on thousands or millions of devices, all of which is not feasible to be shared due to communication, capacity, privacy, complexity, and other reasons. These requirements call for a joint communication and computation co-design, leading to network services and APIs with seamless exploitation of network knowledge for both in-network and external applications. The challenges of the wireless environment, energy efficiency, device capabilities and data handling constraints require 6G networks to provide efficient platform support for distributed AI learning and inference functions.

Section 5.2.1 focuses on *distributed sensing and communication* scenarios where the applications benefit from tight integration of sensors and communications. Section 5.2.2 provides a neural network splitting solution for cooperative edge inference, building on the fact that portions of (deep, convolutional) neural networks can be computed at different locations of the network in a mix of local resource poor devices and cloud servers. Section 5.2.3 addresses the goal-oriented communication, in which the actual performance of communication is measured not by bit rate but by its effectiveness at higher layers by measuring the inference reliability, accuracy, or confidence. Finally, Section 5.2.4 describes experiments for a proactive edge application that predicts network impairment and allows the edge device to prepare for a temporal networkless operation.

5.2.1 Scalable and resilient deployment of distributed AI

Hexa-X envisions several use case families with large application systems distributed on a massive number of devices and network components. The *interacting and cooperative mobile robot* use case requires real-time intelligent decisions based on distributed and resource efficient data and model sharing. Similarly, in applications of *hyperconnected resilient network infrastructures* a huge amount of data will be distributed on thousands or millions of devices, and all this raw data is not feasible to be shared due to communication, capacity, privacy, complexity, and other reasons. Between the two extreme solutions, i.e., all raw data is communicated among devices and the cloud, and the fully autonomous devices using local AI algorithms only, it is more feasible to realize a heterogeneous AI and data sharing landscape. Some application elements will be loosely coupled and just sharing small, maybe delay-tolerant messages, other ones require real-time sensor sharing, yet another type will distribute the AI processing, not just the learning but also inference. The shared information can also be explicitly defined, and AI components use it as independent inputs, but it can also be implicit model state information or internal (latent) states during inference in model split architectures. In the latter case the shared data is typically a result of jointly trained models.

This study focuses on *distributed sensing and communication* scenarios where the applications benefit from tight integration of sensors and communications. Cooperative perception in is one of the advanced vehicular use cases in [5GAA20], which involves sharing sensor information about the current driving environment among the vehicles and other roadside stations. Using sensor data from nearby objects allows the participating vehicles to increase accuracy of the estimated parameters and form a more complete state of environment, including e.g., visually blocked objects. However, this shared sensor data can be highly redundant and the added value of a given sensor stream for a joint inference task in an AI component depends on the input quality (noise, environmental effects, etc), communication link quality and the level of redundancy compared to other sensor sources, as depicted in Figure 5-6:.

Distributed sensing and communication scenarios can involve hundreds or thousands of collaborating AI components in close proximity to be served by the wireless network. Conventional implementation of these systems imposes extremely high wireless traffic load and massive number of connections with the contradicting requirements of low-latency joint inference. This problem is also formulated in the connecting intelligence target T3 (Resilient communication and compute network services for distributed AI applications in large scales), which is addressed by this study.

In an AI-based cooperative perception application, e.g., in visual analysis (object identification) either raw sensor data or locally processed data about the same objects can be made available from multiple sources. For instance, in this vehicular application where visibility in a road junction is limited, a car may receive relevant but highly overlapping information from other vehicles, from external sensors, pedestrian wearables, city infrastructure sensors, traffic control systems, etc. This system of AI components will constitute a redundant and, therefore, more resilient mesh of raw and AI-processed information. The resulting application-level resilience would also be able to serve latency critical use cases without requiring ultra-reliable and ultra-low latency communication layers. Apart from redundance, AI-type data itself have unique requirements, like high tolerance for lost data or similarly high tolerance for added noise, which allows new technical enablers in communication and AI compute co-design.

This aspect was studied in [BCDH+22] in the context of Distributed Wireless Spiking Neural Network (SNN), which proved to be highly resilient to data (spike) losses, while further benefits were shown if wireless networks can support such AI applications to differentiate the treatment of SNN internal traffic with respect to input spikes. Application-level latency can also be addressed efficiently with such a redundant mesh of AI components by cutting the long tails caused by individual elements with low connection quality or local computing latencies. An interesting enabler for such operation is compressed sensing based multiple access [CSD+17], which provides means for lightweight communication for large number of users if the average activity level is low and the system is tolerant for loss, which fits very well to sparse and redundant AI architectures.

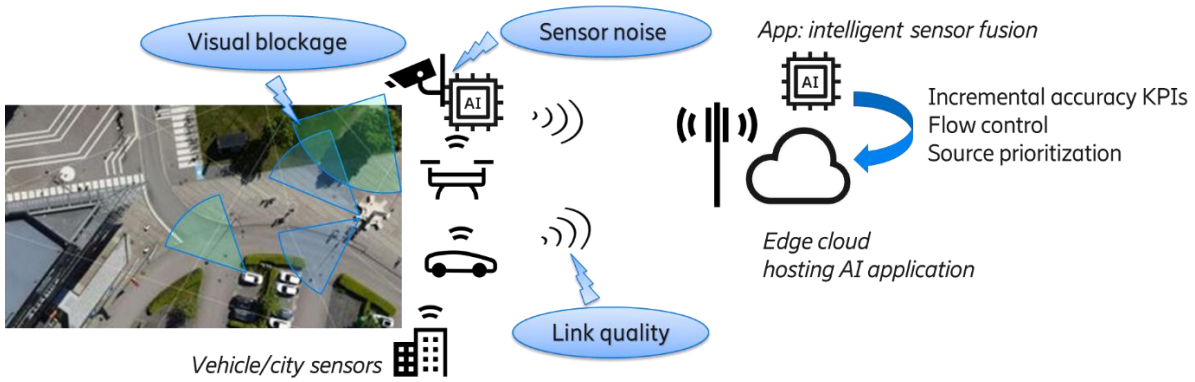


Figure 5-6: Sensor sharing system with distributed AI components.

The system model consists of sensors distributed on multiple wireless devices generating the input data and a model-split distributed AI inference system, with local processing in the devices and a central component aggregating all the device streams with AI-type traffic for a joint inference task (Figure 5-6:). The sensor inputs are assumed to experience a certain level of noise each, therefore resulting in low accuracy, but the combination of multiple streams increases joint accuracy and confidence. The inputs are also assumed to be overlapping in information content. The baseline system is a model-split Neural Network for visual object detection, where each sensor input is partially processed in the wireless device and the latent neural layers are sent to an edge cloud processing function, hosting the combiner and probabilistic layers on the NN, and performing the joint inference for the object identification. With many contributing devices this setup requires high over-the-air traffic, which is also delay-sensitive. Designing this AI system for a required level of accuracy is hard due to unpredictable input sensor and communication quality levels.

To solve the above problems an AI application and communication system architecture is proposed as described below. The aim is to reduce both the overall inference latency and communication load by relying on an incremental evaluation framework. *Incremental inference* can be used in several AI models. One example is the family of ANN-to-SNN converted neural networks [RSP+20]. These frameworks are built and trained on the principles of spiking neural networks, but they can also be implemented on traditional compute architectures. The neurons are stateful and it can run in a

discretized manner (time slots) and produce spike sequences in a spatio-temporal encoding fashion. The accuracy and reliability of the inference increases with each simulated timeslot, even in the presence of high sensor noise or high communication loss. This incremental inference enables the application to evaluate the utility of sources and prioritize the individual streams during the inference process to obtain the required accuracy level. This can be done on a (sub-)millisecond level, which requires *fast feedback from the application* and *quick reaction from the communication layer*.

A specific scenario was simulated to assess the potential traffic load and latency savings with a suitable joint communication-compute designed system. The inference task is object identification based on visual input (camera or ultra-low latency event camera). 50 input sensors are distributed over multiple devices. The image quality from each camera suffers a certain degree of quality degradation, (due to blur, rain, light conditions, partial blockages, etc.), which is simulated with pixel noise and masks. A spatio-temporally trained ANN-SNN converted neural network was used for inference (based on the model from [HWD+20]), which is distributed over the input sources and a central combiner performing the aggregation in both time and device dimensions. It means that the data processing starts separately at the input sources and the final neural layers are done jointly in the central unit (e.g., in edge cloud). Note that, due to the applied neural architecture, the communication between neural layers is spikes (or bits), as opposed to traditional NN, where activation levels are communicated in a split model over the air. The communicated spikes are very small data packages, typically sparse and tolerant to losses. (These aspects and their implications are not discussed here, for more details see [BCDH+22].) It also means that a time step can be well below millisecond timescale, supporting ultra-low latency use cases.

It is also assumed that there is an Inference Control Function in the application to prioritize between different input streams. This partial evaluation is performed after each time step. An input stream utility assessment is made, which may stop a device sending a stream or adjust communication bandwidth among the live streams according to data utility to increase inference accuracy. The increasing accuracy in the simulated system is shown in Figure 5-7/a, for both the baseline system and the one with application and communication control. It can be seen that early inference with lower precision is possible (which can be very useful depending on use case, e.g., prepare for breaking/slowing down), but higher accuracy requirements can also be fulfilled with more delay. Application control can further increase the accuracy by filtering out noisy and misleading input streams.

Further significant gains can be observed in the wireless communication load, both in terms of traffic level and number of active connections (Figure 5-7/b). By gradually eliminating redundant input streams, the average traffic load decreases to less than 20% compared to the baseline case and the average number of required connections is also in this range.

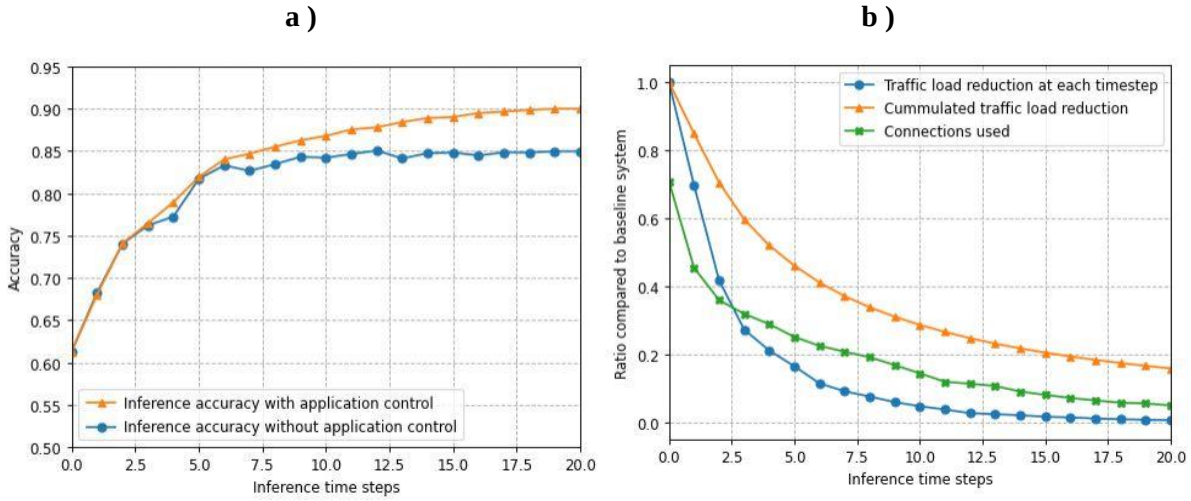


Figure 5-7: Incremental inference for an ANN-SNN converted image recognition neural network. a) Accuracy-latency trade-off can be controlled. b) Load reduction due to the joint application-network control is significant, less than 20% compared to baseline.

Further simulations of the scenario with more devices (with 500-1000 input sensors) results even higher relative gains, which is explained by the observation that the inferencing accuracy curve depends more on the absolute number of high-utility inputs (with low sensor noise and good connection quality), regardless of the number of the remaining lower-utility inputs, which are filtered out early by the incremental inference mechanisms. The device density requirements for the demanding 6G use cases in D1.3 (collaborating mobile robots, smart cities) are up to 5-10 devices/m^2, which is 5-10 times higher than 5G requirements. Obviously, the number of supported devices depends on many technical components in the E2E path, but the above method with joint fine-grained control from the application layer and millisecond level traffic control by the networking layers has a significant contribution to reach the targeted numbers in a *distributed sensing and communication* scenario with ~1000 collaborating AI components (also contributing to the connecting intelligence target T3).

The proposed joint compute and communication system has several advantages over the traditional approaches. With the option of accuracy-latency trade-off, it is possible to do both early phase ultra-low latency inference, as well as higher accuracy at the cost of higher delay. In use cases like above, where multiple inputs provide overlapping information for the inference task, over-the-air communication can be significantly reduced with the help of application domain functions (the assessment of input stream quality and utility) interfacing with network layers on a fast and high data granularity level, which may require per packet control on millisecond timescale. The benefits, however, are a reduction of traffic and connection load down to 20% compared to the baseline system, which can also be translated to higher device density for distributed AI.

5.2.2 Joint communication and computation orchestration for edge inference

Learning and inferencing at the edge require to collect, pre-process (e.g., extract features), transmit, and process data (or features) remotely in a continuous fashion. Then, intermediate computation results or raw data need to be continuously and periodically exchanged among heterogeneous intelligent agents to perform complex tasks in a collaborative way (e.g. federated learning or cooperative inference) with an energy frugality perspective, and with target levels of dependability, entailing accuracy, latency, inference confidence, etc. In this section, going beyond the results presented in [HEX-D42] on edge inference, the concept of cooperative inference via DNN splitting is introduced, building on the fact that portions of (deep, convolutional) neural networks can be computed at different locations of the network (e.g., a part locally at a resource poor end device, and part of it in the edge cloud), in order to reduce the communication overhead, but also to smartly exploit current availability of (possibly volatile) radio and computing resources, both at user and network side. This concept is known in the literature as DNN splitting [MLR22], and is foreseen to be one of the enablers of edge machine learning, thanks to its flexibility in distributing workload across different agents in a system. The proposed contribution is related to the orchestration of resources in the presence of DNN splitting. Namely, going beyond the state of the art, it is proposed to jointly optimise the splitting point choice and the transmit power in a dynamic and adaptive way.

In particular, it is well known that some DNN architectures, such as Convolutional NN (CNN), have natural bottleneck, i.e. layers that extremely reduce the input size. This can help in reducing the communication overhead of such partial offloading services, while still guaranteeing end-to-end latency constraints involving communication and computing. Also, edge computing resources are generally volatile and less reliable, in terms of availability, than central cloud resources. This means that, when accommodating several services with different priorities, computing resource shortages can occur and force end devices to perform part of the computation on-board. To efficiently exploit edge resources, the decision on local and remote workload, as well as on radio and computing resources, should be optimised dynamically, and adapted to current connect-compute (i.e., wireless and computation) resources. An exemplary scenario in which edge inference is performed is depicted in Figure 5-8, where a user equipment, embarked with a pretrained DNN model, uploads extracted features (or raw data depending on the splitting point choice) to a Mobile Edge Host (MEH) through the wireless connection

with an AP. The DNN can be split at different points, Local computing delay, communication, and remote computing delay are considered in the overall estimation of the end-to-end inference latency.

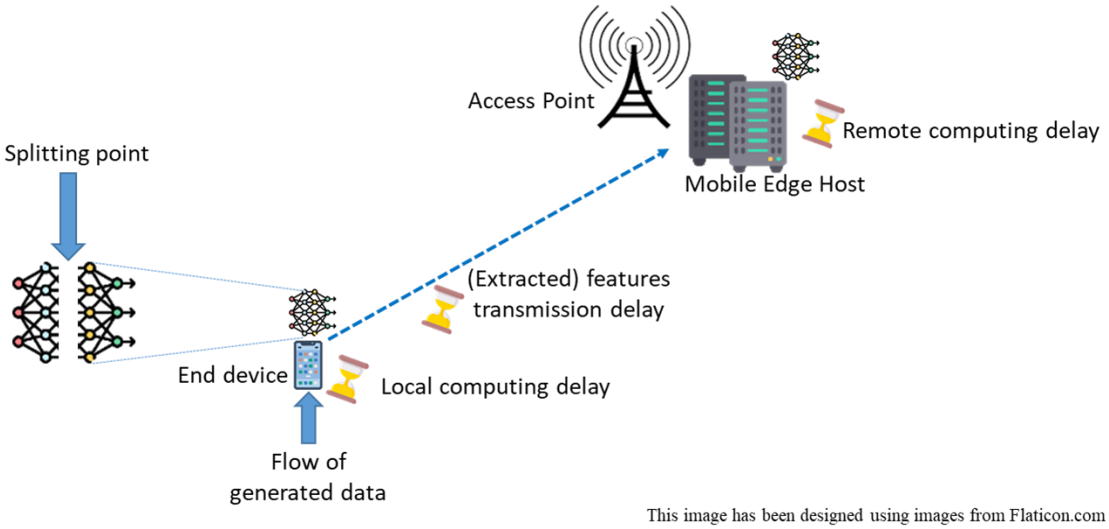


Figure 5-8: Cooperative edge inference via DNN splitting.

In this case, several KPIs/KVIs are considered, including energy efficiency (in terms of energy per task), E2E latency (in terms of communication and computing), and inference accuracy. Potentially, in this context, privacy arguments can be considered, but this goes beyond the framework proposed in this section. The goal is to obtain accurate inference results on time, with the least energy consumption. As quantified target, energy reduction at the device side thanks to workload offloading play a key role. Also, among the others, cooperative mobile robots (cobots) is a typical use case in which radio (to communicate and exchange information) and computing (to perform complex cooperative inference and control) are exploited, and their joint optimisation is beneficial to achieve target effectiveness (based on, e.g., inference accuracy levels) with increase efficiency. Then, building on the concept of splitting point, the proposed solution aims at jointly optimising the splitting point selection and device transmit power, under time-varying wireless channel conditions and Mobile Edge Host (MEH) availability. The goal is to minimize device energy consumption (including transmission and computation) under end-to-end service delay constraints of the edge inference service (both in average and probabilistic sense), which involves: i) local computation delay, ii) wireless uplink delay, and iii) remote computation delay. While the average delay constraint only considers the expectation, the probabilistic delay focuses on outages, defined as the probability that the E2E delay exceeds a threshold. A predefined target is set for the outage probability. The system involves one end user, which generates, pre-processes, and transmits data, and one MEH, which remotely performs the remaining computations, based on the selected splitting point. An adaptive algorithm based on stochastic optimisation is used to dynamically optimize the splitting point selection and the end user transmit power, in a per-slot basis, based only on instantaneous observations of: i) wireless channels, iii) data arrivals, and iii) currently available remote computing resources, supposed to be dynamically assigned by the MEH, based on the current computational traffic (assumed to be an exogenous variable here). Technical details can be found in [LMA+23].

The algorithm is tested in the context of edge image classification. The end user aims to classify images from the Imagenet data set [DDS+09]. The state-of-the-art MobileNetv2 CNN [San18] is exploited to classify data, and assumed to be pre-trained and pre-uploaded at the device and the MEH. Among the 53 convolutional layers of MobileNetv2, 20 possible splitting points are considered, including splitting point “0” (i.e., full offloading), and splitting point “19” (i.e., full local computation). Current input size and cumulative number of computations (in terms of Multiply and Cumulate – MAC operations) are shown in Figure 5-8, and are based on formulas available in [LMA+23], [LKR+22].

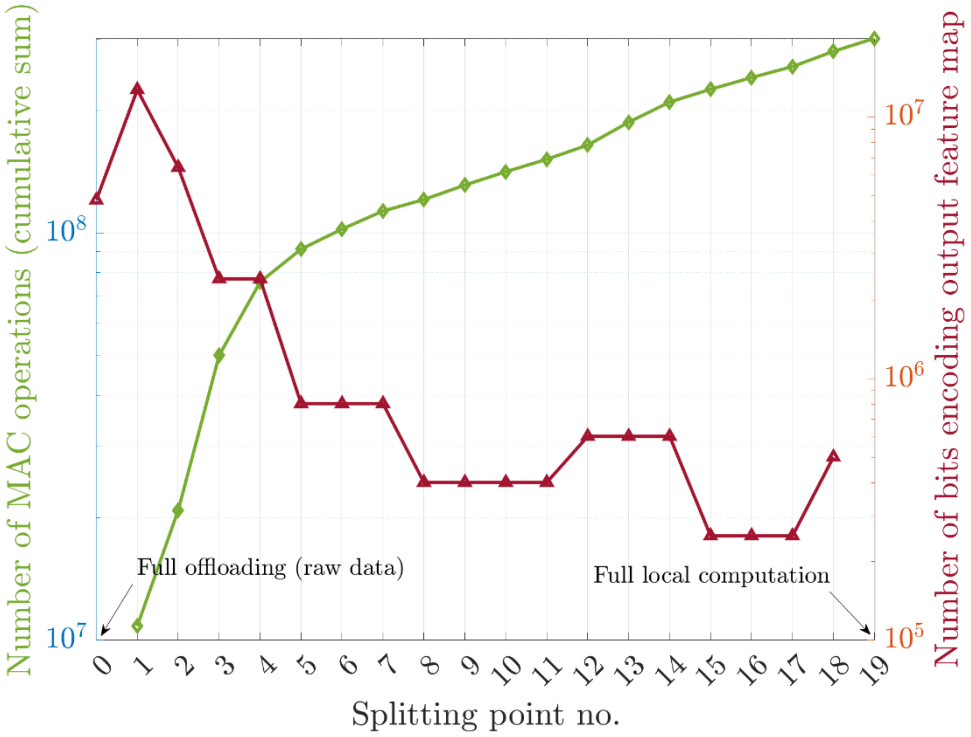


Figure 5-9: Number of local MAC operations and output size as functions of the splitting point.

The end device energy consumption is computed through a recently developed empirical model [LKR+22], which considers NVIDIA edge boards, and provides a model based on the MAC operations, with different parameters that depend on the exploited hardware. At the end device, the edge board NVIDIA board *Jetson TX2* is assumed, while at the MEH, the edge board *Jetson Xavier NX* is assumed. The total computing power of these edge boards is 1.3 TFLOPS and 21 TFLOPS, respectively. Also, different computing resource availabilities are considered at the MEH. Namely, by considering a parameter $\alpha_{r,max}$, i.e., the total computing power 21 TFLOPS is multiplied by a random variable uniformly distributed in $[0, \alpha_{r,max}]$ and changing over time without a priori knowledge of its statistics, with $\alpha_{r,max} \in [0,1]$. The AP operates at 3.5 GHz, with 200 MHz bandwidth, and noise power spectral density -174 dBm/Hz. The device is placed 30 m away from the AP, path loss exponent 3 is used to generate the path loss, and Rayleigh fading with unit variance is assumed. The maximum and minimum transmit power of the device are 100 mW and 10 mW, respectively, while the number of newly generated patterns at each time instant follows a Poisson distribution with parameter 12.

Numerical results are shown in Figure 5-10:. In particular, Figure 5-10:a shows the trade-off between the average E2E delay and the average energy saving with respect to the full local computation, for different MEH's CPU availability conditions. Thus, these results inherently show the comparison between the proposed splitting point selection method, and the full local computation. As it can be noticed, the average E2E delay increases as the energy gain increases, until reaching a predefined threshold of 50 ms, set a priori as average delay requirement. The curves clearly show that higher energy saving requires higher delay (going through the curves from left to right). Nevertheless, the method is able to always guarantee the delay constraint (black horizontal dashed line).

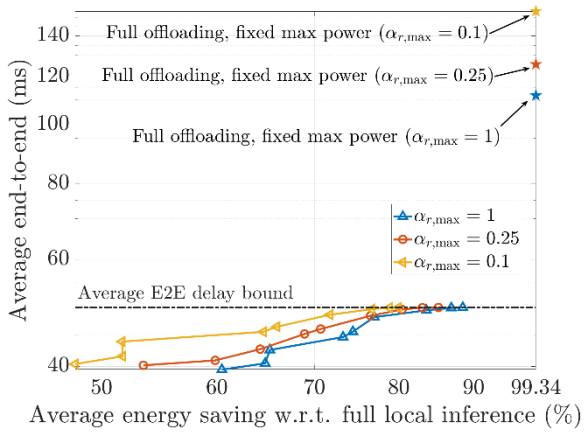
Moreover, while the higher CPU availability leads to up to 90% energy savings (rightest point of the blue curve), the reduction of MEH's CPU availability incurs in lower savings, as more and more computations need to be pushed locally at the device. As a further comparison, Figure 5-10:a shows the average E2E delay obtained by fixing the splitting point to "19", i.e., always full offloading. . Although the full offloading solution is the best in terms of energy saving, it does not guarantee the end-to-end delay constraints. This is due to the higher computation time at the MEH, and it shows the intrinsic relation between remote computing and wireless resources. For $\alpha_{r,max} = 1$, solutions are comparable due to the fact that the MEH's CPU is highly available, i.e., full offloading is convenient. To complement the results, Figure 5-10:b shows the complementary cumulative distribution function

of the E2E delay. Again, our method is able to keep the outage (delay higher than a threshold) under the predefined target, while this is not the case for the full offloading strategy, which experiences long distribution tails due to unexpected MEH's CPU unavailability.

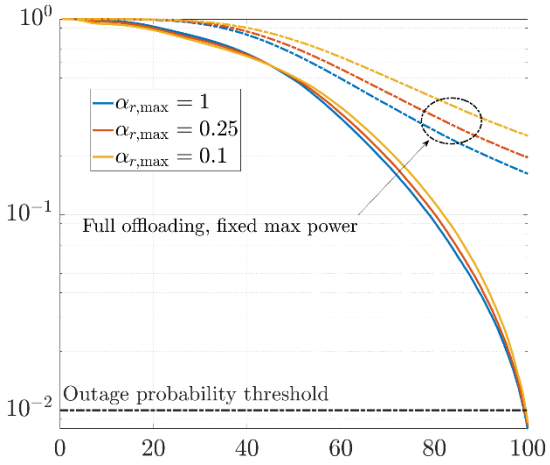
The maximum energy saving of each curve is represented by the rightest points of the curves (i.e., the ones that exactly attain the E2E delay requirement). Looking at these points for the three curves (blue, orange, and yellow), it is possible to notice that a lower energy saving is obtained in case of reduce MEC resources availability. This is mainly due to the fact that, as the availability reduces, more and more computations are pushed locally at the device.

To further validate this statement and the proposed method, Figure 5-10:c shows the average splitting point selection, as a function of the MEH's CPU availability, for different channel conditions, i.e., different path loss exponent, which we denote a β.This can be interpreted as the average depth of local computations. As it can be easily notice, highly available remote computing resources result in less local computational burden, i.e., the method choses to often offload data without pre-processing. However, when reducing the MEH's CPU availability, computations are more and more pushed locally, as the device needs to make it up with remote CPU power volatility, to attain the desired end-to-end delay constraints. This shows that the method is able to autonomously select the right splitting point to minimize the energy consumption under delay constraints. Also, the standard deviation experiences an interesting behaviour, which depends on the path loss exponent as well. Namely, for favourable propagation conditions, it is possible to observe the following: for low availability, full local inference is preferred most of the time (high average and low standard deviation). Then, increasing MEH's computing availability, the mean decreases (i.e., more computations are pushed remotely on average), while the standard deviation increases, as the method has more degrees of freedom in selecting the best SP, based on current connect-compute resources. By further increasing the MEH's availability, the mean and the standard deviation decrease again, due to the fact that full offloading decisions become more frequent and mostly preferred, although not always possible due to channel fluctuations.

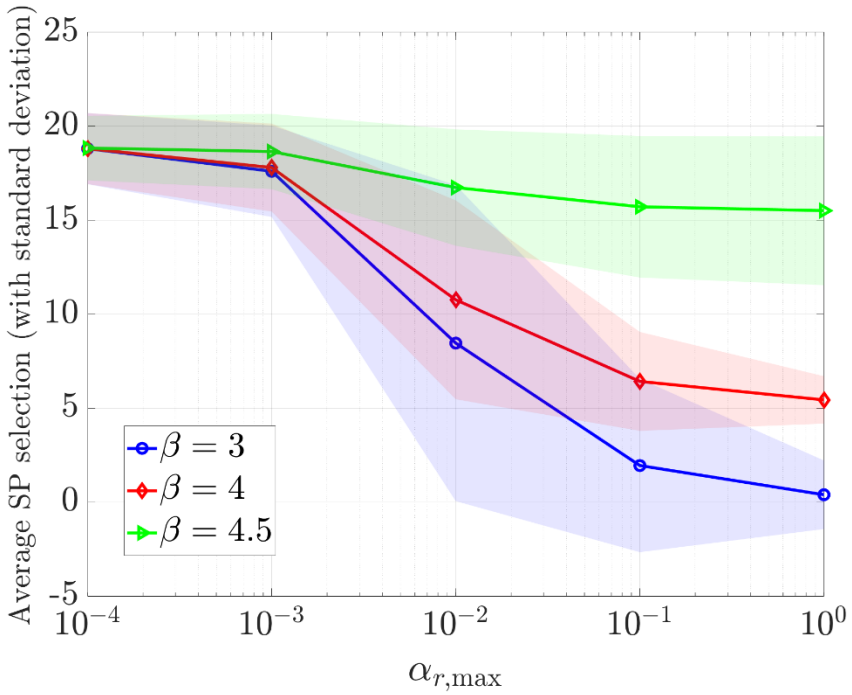
Finally, the fact that the method is able to make it up with remote CPU unavailability, suggests that future work could take into account the optimisation of remote computing resources to save energy at the network side, as additional quantifiable target.



a) trade-off between average E2E delay and energy saving for different MEH's computing resource availabilities



b) Empirical complementary cumulative distribution function of E2E delay



c) average splitting point selection (i.e., depth of local computations) for different MEH's computing resource availabilities and path loss exponents

Figure 5-10: Trade-off between energy and delay, as a result of splitting point selection through proposed method, and average splitting point selection as a function of MEH's CPU availability and path loss exponent β.

Conclusions

This section proposed an online adaptive method to split computational workload for an edge inference service. In particular, an end devices collects data and dynamically decides whether to fully offload inference computations, or perform some computations locally, depending on wireless channels, data arrivals, and remote CPU availability. The proposed method, whose technical details can be found in [LMA+23], is able to autonomously adapt the splitting point selection, to always attain the minimum energy consumption under a predefined average E2E delay constraint. The comparison with the full offloading and the full local computation settings show the superiority of the method and its degrees of

freedom in selecting the optimal CNN splitting point. Future works involve the optimisation of remote computing resources to accommodate different type of traffic, and the use of realistic model of user transmit energy, taking into account the end-to-end system energy consumption. Finally, the concept of goal-oriented communications is a further direction to be explored, with a possible merge of the proposed framework, with the one presented in the next section.

5.2.3 Goal-oriented communication approach for edge inference

Building on the concept of the previous section and on [HEX-D42], this section aims to introduce a novel communication paradigm, which is receiving considerable attention due to its potential to achieve desired application performance, without necessarily pushing communication KPIs to their limits. The results of this section and the technical details can also be found in [MFB+22]. Indeed, while the classical (data-oriented) communication paradigm is to reliably transmit bits (i.e., it focuses on bit-quality oriented metrics), goal-oriented communications measure the actual performance of communication by its effectiveness at higher layers, e.g., by measuring the inference reliability (accuracy, confidence, etc.). The idea is to potentially accept a certain level of "bit-level" errors (or packet losses), without necessarily focusing on communication reliability, as far as the application performs as needed, a concept that one can refer to as *goal-effectiveness*. Also, from an operational perspective, the challenge is to not define, a priori, a set of communication KPIs for a given service, but rather adapt such parameters to current connect-compute network states, based on application measurements and responses. In particular, for the goal-oriented approach to be successfully applied, the following definitions are required [MFB+22]: i) *goal value*, i.e., a measure of how close is a user to achieving the goal or, also, a measure able to determine if the goal has been achieved; ii) *goal-effectiveness*, i.e., the probability of the goal value being above a threshold; iii) *goal cost*, i.e., the price to pay to achieve the goal in terms of, e.g., wireless resources or energy consumption.

Although in this section the goal-oriented communication paradigm is applied to an edge inference scenario, the concept is more general and can be applied to several use cases involving communication and computing, but also control. As such, the Hexa-X interactive and cooperative mobile robots use case is the most relevant one to the proposed technical enabler, with KPIs involving energy efficiency, E2E latency and, at the application layer, inference reliability.

However, in this section, focus is on an edge inference service, comprising an end device (e.g., a user equipment or a sensor) continuously generating data and uploading them to an MEH, through the wireless connection with an AP. In this case, with focus on a classification task, the goal value can be related to the classification accuracy. The goal-effectiveness is defined as the probability that a pattern is correctly classified within a target E2E deadline, comprising communication and computing [MFB+22REF]. Also, at the network side (i.e., at the edge cloud) it is assumed the possibility to perform computations in multiple MEHs (ensemble inference), to increase the reliability of computation in case of server failures due unpredictable compute resource shortages, typical of resource limited edge computing scenarios. The system scenario under investigation is depicted in Figure 5-11, with an end device (e.g., a sensor) uploading, through the wireless connection to an AP, input data to nearby MEHs hosting intelligent agents, which are on-boarded with pre-trained ML models (e.g., neural networks). The intelligent agents, in their turn, produce inferencing outputs (e.g., predictions or recommendations). Differently from the previous section, no computations are performed locally. The general assumed MEC infrastructure deployment consists of multiple MEHs deployed across a network coverage area. The first one is a *primary MEH*, which is collocated to the AP. Also, device requests can be further possibly treated by another MEH, interfaced with the primary MEH via wired (e.g., fibre) backhaul connection; this is termed after as a *helper MEH*. The aim of the helper is to improve performance, e.g., in terms of goal accomplishment, i.e., inference effectiveness, thanks to offering single point of failure avoidance, as well as through inference diversity gains (ensemble inference). It is assumed that bit error rate on transmitted data is the cause of inference accuracy degradation, and the aim is to explore the trade-off between energy consumption (at both user and edge cloud server side) and goal-effectiveness. Technical details can be found in [MFB+22]. Next, numerical results involving goal-effectiveness, energy consumption (goal cost) and communication KPIs (BER) are presented, to link goal-

effectiveness to classical communication performance. Details on simulation parameters can be also found in [MFB+22].

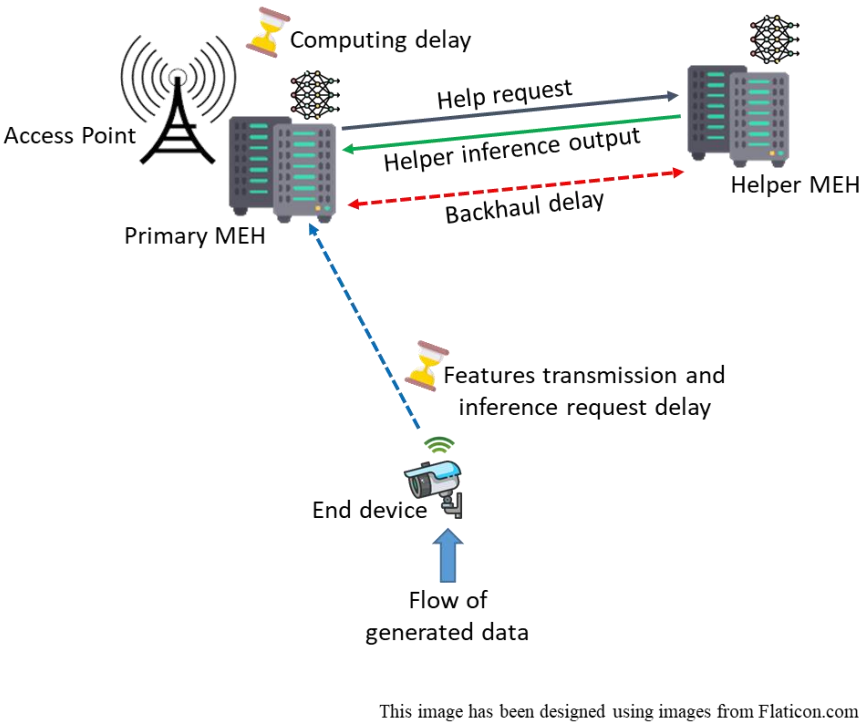
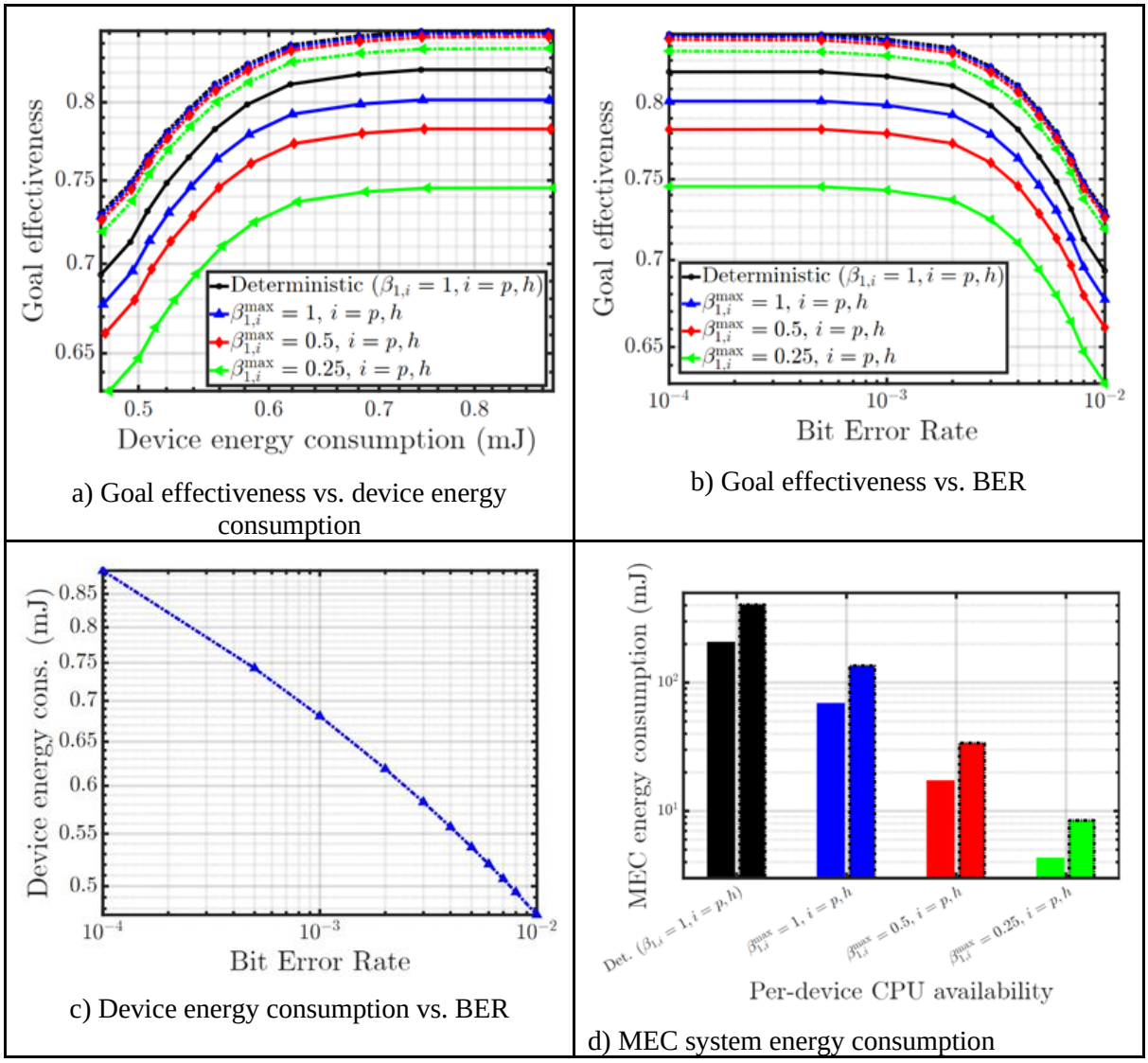


Figure 5-11: Network scenario [MFB+22].

As a first performance evaluation Figure 5-12:a shows the goal-effectiveness as a function of the device transmit energy, obtained by varying the BER requirements during transmission. Also, to have a complete view of system performance, Figure 5-12:b and Figure 5-12:c show, respectively, the goal-effectiveness and the device energy consumption as functions of the same BER requirements used to obtain Figure 5-12:a. In these figures, the different colours encode different values of MEH's CPU average availability. At the same time, solid lines represent standalone inference (i.e., only the primary MEH participates, while the helper is not implicated in the inference process.



a) Goal effectiveness vs. device energy consumption

b) Goal effectiveness vs. BER

c) Device energy consumption vs. BER

d) MEC system energy consumption

Figure 5-12: Goal-effectiveness, wireless reliability, user and servers energy consumption

The effect of BER on goal effectiveness: We recall that the goal-effectiveness is defined as the probability that a pattern is correctly classified within a target E2E deadline First of all, the effectiveness increases, as the device transmit energy consumption increases, as expected, with different performance depending on MEH's availability levels. However, above a given threshold, BER requirements do not strongly affect goal effectiveness, meaning that a slightly increased BER leads to similar performance in terms of the effectiveness, in all network conditions (obviously with different values according to MEH availability). It is noteworthy that effectiveness is invariant to communication reliability changes in the range BER = [10^{-4}, 10^{-3}] (see Figure 5-12:b). This first observation suggests that classical communication related reliability performance can be relaxed almost without any impact on goal-effectiveness, thus non-negligible gains in terms of device energy consumption are expected, as visible from Figure 5-12:c, where the device energy consumption is plotted as a function of the BER.

The effect of ensemble inference: Now, an analysis of the beneficial effect of the helper MEH presence into the system is provided. In particular, from the dashed dotted lines in Figure 5-12:a and b (i.e., the curves on the top of the figure), it can be noticed how goal-effectiveness experiences a great improvement, if compared to the standalone inference case, on the basis of the same device energy consumption cost (i.e., the same BER requirements). Equivalently, the same effectiveness can be achieved at a lower goal cost, by further relaxing the BER requirements (e.g., see the 80 % effectiveness case). Of course, the lower energy consumption at the device side translates into a higher energy

consumption of the MEHs, as visible from Figure 5-12:d, where the energy consumption of the MEC system for a single inference request is shown, as a function of MEHs' CPU availability, in case of standalone (first bar) and ensemble inference (second bar), both of the same colour to represent the same MEHs' CPU availability. By comparing standalone and ensemble performance for the same MEH availability, an approximately doubled energy consumption at the MEC infrastructure level for the ensemble case, as visible from Figure 5-12:d, due to the obvious fact that, on average, the two MEHs address requests at the same CPU speed. Nevertheless, an interesting outcome can be noted: by comparing the ensemble inference at half maximum available MEH CPU (red dashed dotted line), with the standalone inference at full maximum available CPU availability (blue solid line), better performance in terms of effectiveness against device energy consumption is guaranteed for the prior case, while also achieving a lower energy consumption at the MEC network infrastructure level. This result is due to a quadratic law of the dynamic CPU energy consumption with respect to the CPU cycle frequency, and it shows that better performance in terms of goal achievement can be obtained without paying any price (but actually experiencing a gain), in terms of network energy consumption. In particular, this behavior suggests that, in the case of ensemble inference, it is convenient to have two available CPUs at half availability, rather than one CPU at full availability (on average), for a twofold reason: i) the total MEC energy consumption decreases; ii) the reliability improves, since single point of failure issues are counteracted. Also, for the deterministic, the full CPU, and the half CPU availability cases, ensemble inference outperforms the deterministic standalone method, due to the fact that the two classifiers cooperate to improve their standalone capabilities. Also, the ensemble inference at 25% CPU availability outperforms the deterministic standalone classifier, with a greatly decreased energy consumption (around 20 times) at the MEC infrastructure level.

Conclusions

A performance evaluation of a goal-oriented communication system has been proposed, to show the difference and the links between communication reliability and KPIs (e.g., BER) and certain application performance, a concept that can be referred to as goal-effectiveness. Through numerical simulations, it has been shown how the effectiveness can achieve good performance also in the presence of relaxed BER requirements, suggesting that high communication reliability is not necessarily needed, as far as inference performance reaches target levels. Furthermore, the power of ensemble inference (implemented thanks to MEHs' cooperation) in improving goal performance has been shown. This depends on specific deployment characteristics, such as the availability of MEHs. More numerical results can be found in [MFB+22]. Future directions involve goal-oriented optimization strategies aiming at dynamically adapt communication KPIs to the actual goal-effectiveness [MFB+23].

5.2.4 Network impairment resilience of autonomous agents

Our goal is 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. 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 collected labelled training data from developer mobile applications and build predictive ML models by using the data rate and network measurement time series. The problem was solved by a combination of neural networks and gradient boosting based on data collected at the edge devices and uploaded to a server. As a key new idea, gradient boosting was deployed for explainable AI to guide the feature engineering procedure. New features involved time series similarity with respect

to preselected normal and abnormal release time series. The final model was extracted from the Python modelling tool as Java code for the edge device.

As a special case then the agent contributes to a FL model, a separate low resource ML component located at the edge device is responsible for the prediction of network impairment. Since this component only performs prediction and not learning, it can issue a warning to the agent with sub-millisecond latency. (Note that model training is performed centrally in a server, based on data collected at the edge and uploaded to a sever.) In case of a warning, the FL component (1) first sends out a (very small size) message to its peers to inform them the potential disruption, thus avoiding unnecessary delays due to waiting for timeout, and (2) as long as sufficient bandwidth is available, transmits its state so that the FL system can redistribute the tasks of the agent.

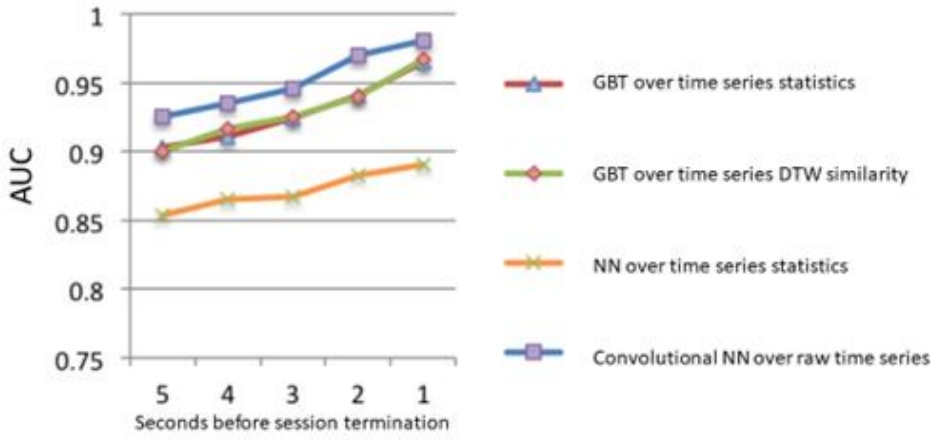


Figure 5-13: Session abnormal release prediction performance of different methods.

As a continuation of our preliminary experiments in Deliverable D4.2, we use generated data to predict interference caused by a moving obstacle. In our experiment, we generated UE sessions and built prediction models on the normal vs. abnormal release. We generated time series of signal strength, signal to noise ratio and uplink, downlink network performance. As the function of the seconds before session termination, we compared the performance of the following four methods, as seen in Figure 5-13::

- Gradient boosted trees over statistics (mean, median, minimum, maximum for the time series values and their change);
- Gradient boosted trees over the Dynamic Time Warping similarity of the time series;
- Neural networks over the time series statistics;
- Convolutional neural networks over the raw time series, the best performing method.

We target the Massive twinning and Robots to Cobots Hexa-X Use cases, both in which real-time intelligent decisions have to be made based on distributed data available partly locally and partly from the network. We also target AI partners, agents that can be much more general-purpose and act as a partner which autonomously and adaptively interacts with other agents (humans/machines), by interpreting intents and surroundings, performing challenging and risky tasks. Finally, AI-assisted V2X rely on data gathered through the automotive services offered by communications networks with low latency essential for the AI algorithms that control the traffic. We address latency in conjunction with network impairment, which means very fast implementation of the available offline backup control mechanisms.

Primarily, we address the AI and computation family KPIs, including dependability attributes, including safety (constraints for human-machine co-working guarantees, in case of network connectivity impairment) and maintainability (recovery of network failures by preparing for the offline optimisation

of the processes at the agent). We also address the security KPI of AI and computation by signalling of incidents in network operation ahead in time.

5.3 Efficient training for distributed AI

During AI workload placement, the specific requirements of distributed AI deployments should be taken into account considering among others data and resource availability as well as trust levels. In Section 5.3.1, optimization problems in a multi-cell multi-user MIMO system are considered as a case of decentralized execution of ML models. In Section 5.3.2, an algorithm is provided for load balancing in hyperconnected infrastructures. In cases when data is distributed on thousands or millions of devices, we achieve low latency and high quality distributed ML service by providing load balancing to remedy potential hot spots and ensure data type diversity. Finally, in Section 5.3.3, a technique is presented to achieve communication and storage-efficient FL training by customising FL tasks to the available compute, communication and energy resources of the contributing network entities.

5.3.1 Centralized training and decentralized execution (CTDE) approach to multi-cell multi-user MIMO

In wireless access networks, data are inherently distributed due to geographic distribution of cells. Ideally, the contextual information on state of network and channel at one site can be collected across sites for a globally optimal solution to many optimization problems in such cellular networks. However, the centralized approach poses difficult challenges due to the dynamic nature of cellular networks. Centralized training and decentralized execution (CTDE) is a promising in-network ML framework for future cellular networks, allowing to evolve towards more intelligent and scalable architectures. An example of CTDE is illustrated in Figure 5-14:. This subsection considers a CTDE approach to precoding/beamforming problem for multi-cell multi-user MIMO systems.

Downlink multi-cell multi-user MIMO is a promising technique to achieve higher throughput in a multi-cell environment. However, in general, the optimization problems in a multi-cell multi-user MIMO system are difficult to solve in practice. For instance, coordinated multi-point (CoMP) with joint transmission (JT) is a cellular data transmission technique involving simultaneous transmission from multiple base stations (BSs) to the same user. However, potential solutions to the JT scheme require significant amounts of global channel state information (CSI) and data sharing between the base stations, which is not only expensive but also difficult in real-world cellular systems. The optimal JT scheme can be reduced to coordinated beamforming (CB) schemes based on the transmission of the signal by a single base station that require local CSI and no inter-cell data sharing. In this case, each BS operates independently by treating the interference as background noise and the multi-cell multi-user setup can be modelled as MIMO interference channel (IFC). Compared to a single-cell system, the performance of MIMO IFC can be severely impacted by inter-cell interference, which becomes a crucial limiting factor.

Multi-cell, multi-user precoding problems can be seen as a multi-agent system that learns to coordinate transmission schemes (or action policies) in interaction with other base stations (or other agents). The multi-agent problem requires complex inter-cell interference coordination in the sense that each BS should exhibit cooperative behaviour to maximize the signal power to a desired user, while minimizing the interference power to other users in the multi-cell environment. This problem poses two main challenges: i) multiple actors (or agents) with partial observability and ii) multi-dimensional continuous action space. To address the two main challenges, we adopt a CTDE framework based on a multi-agent deep deterministic policy gradient (MA-DDPG) . The basic idea is illustrated in Figure 5-14:, where decentralized actors with partial observability can learn a multi-dimensional continuous policy in a centralized manner with the aid of shared critic with global information.

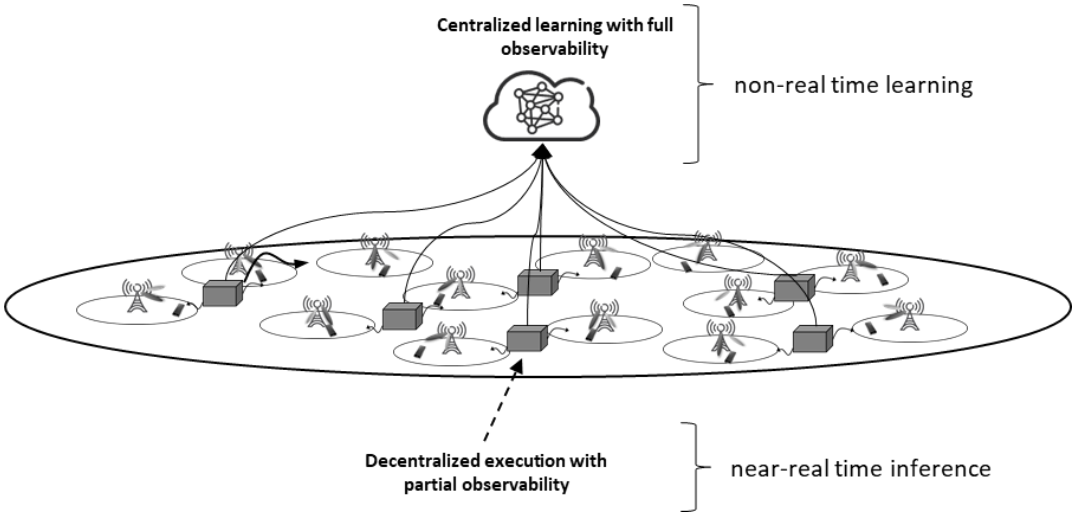


Figure 5-14: CTDE approach to precoding/beamforming problem for multi-cell multi-user MIMO systems

To address the multi-cell multi-user MIMO precoding problem, we use a MA-DDPG algorithm to train decentralized actors and share critic in multi-cell multi-user MIMO systems under the assumptions of local CSI and no inter-cell data sharing. This allows for non-real-time training based on global information and real-time execution based on local information. More precisely, MA-DDPG algorithm comprises decentralized agents (i.e., actors) with partial observability (e.g., local CSI) to learn a multi-dimensional continuous policy and one shared critic which has access to global information (e.g., global CSI and precoding vectors of every cell) that trains actors to collaborate to optimize the global objective. To fulfil the real-time requirement of beamforming schemes, each actor $i \in \{1,2\}$ chooses a unit Frobenius norm precoding vector w_i based on the local CSI information $h_i = [h_{i,1}, \dots, h_{i,n_t}]$ and $g_i = [g_{i,1}, \dots, g_{i,n_t}]$ at its associated BS, where the j-th elements $h_{i,j}$ and $g_{i,j}$ denote the path gain from the j-th antenna of BS i to the desired UE i and the other UE, respectively. The shared critic learns an action-value function in a centralized manner based on the global information about the channel states and the actors' policies and uses it to learn the action policies that maximize the expected collective reward based on achievable rate pairs $r_1 = \log_2\left(1 + \frac{|h_1 w_1|^2}{\sigma_n^2 + |g_2 w_2|}\right)$ and $r_2 = \log_2\left(1 + \frac{|h_2 w_2|^2}{\sigma_n^2 + |g_1 w_1|}\right)$.

Figure 5-15 presents a numerical example with two BSs, where each BS equipped with four antennas is serving a single-antenna user. For benchmark purposes, we plot the maximum single-user and sum rate points along with the time-sharing inner bound on the pareto-boundary of achievable rate region that is defined as the set of achievable rate pairs. Then, we plot a performance trajectory of MA-DDPG during training with two actors and one critic. The critic aims to achieve a Pareto-optimal beamforming strategy by maximizing the collective reward, defined as a weighted sum of the achieved rates of the two users. We denote by α (a value between 0 and 1) the weighting scalar for the rates of the two users. The dashed lines in Figure 5-15 depict the corresponding learning behaviours under different weighting scalars. We can see a performance gap between the rate pairs achieved by the naïve learning implementation and the Pareto-optimal rate pairs. To close the gap, we propose a feature engineering method, called phase ambiguity elimination (PAE), as a pre-processing step on the input of channel states to improve the learning performance. As shown by the solid lines in Figure 5-15, the improved implementation with PAE can learn a Pareto-optimal beamforming strategy. The takeaway from addressing the phase ambiguity issue in this case study is that applying AI with deep domain knowledge of the underlying communication technology is crucial for achieving optimal performance in AI-enabled RAN.

In summary, the CTDE framework enables centralized learning with decentralized execution, providing a practical and realistic approach for real-world cellular environments that require varying levels of

observability and time constraints. As one of the use cases that can benefit from this framework, the multi-cell multi-user MIMO problem was addressed. The targeted KPIs in this use case are maximizing the sum-rate for multi-cell multi-user MIMO and scalability to accommodate the increasing number of users and cells, while ensuring real-time fulfilment of beamforming schemes.

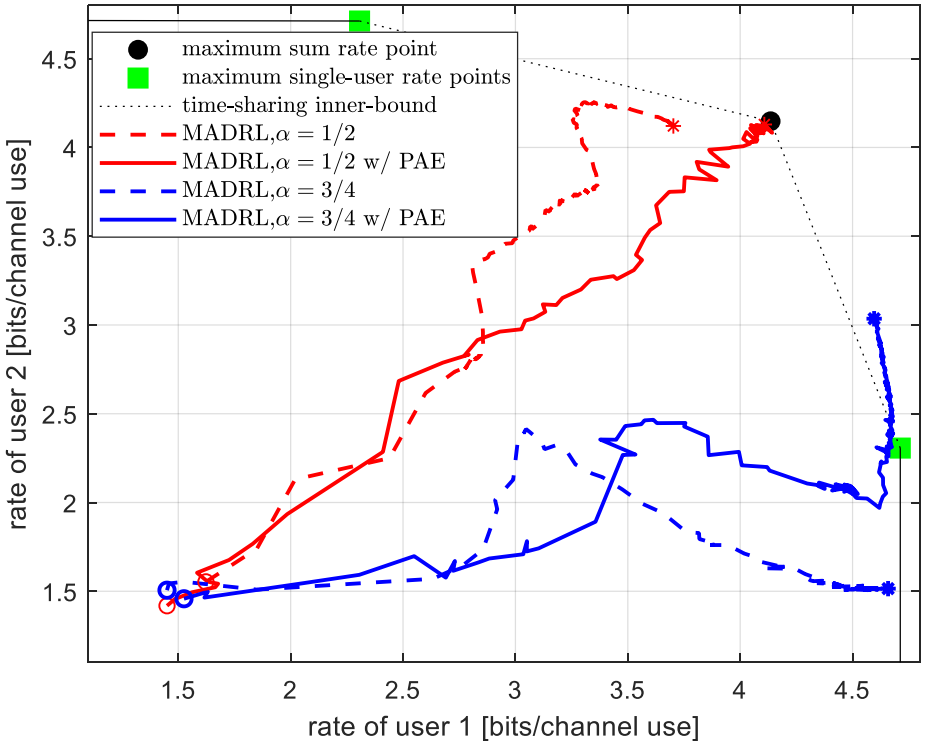


Figure 5-15: Convergence of two user performance in two-cell two-user multi-antenna system,.

5.3.2 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. Massive twinning relies on a fully synchronized and accurate digital representation of the physical and human worlds based on a wide variety of sensor information., To enable more efficient interaction of production for digital twins in manufacturing, we have to encompass a larger extent of the respective processes, and also to achieve the transfer of massive volumes of data from a wider range of sensors and actuators within the factory, including the cooperation among multiple digital twins in a flexible production process. In an immersive smart city, effective management of all factors of persons, vehicles, infrastructure, weather, pollution etc, on various time scales require a large number of heterogeneous sensors to be connected.

In the use case scenario of Figure 5-16 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.

Our implementation relies on a central server that collects data type histograms from the FL nodes and makes reconnection decisions by using our algorithm based on the [ZSBB21]. Each FL node runs a module responsible for collecting local histograms and executing the reconnection requests from the server.

Our architecture shown in Figure 5-16 is based on the Dynamic Reconnection Master (DRM) central authority. Dynamic Reconnection Workers (DRW) are assigned to each FL node. The roles of DRM are

- Collecting local histograms from the DRWs;
- Executing the partitioner algorithm described next to reach balance;
- Sending reconnection commands to DRWs, including both the original and the new DRW corresponding to each ML edge device that needs to be reconnected.

The roles of DRW are

- Computing the local histogram of data volume by data type using sampling;
- Periodically submitting the histogram to DRM;
- Executing the migration step, ordering the ML edge device to reconnect to another FL node, including necessary state migration to the new FL node.

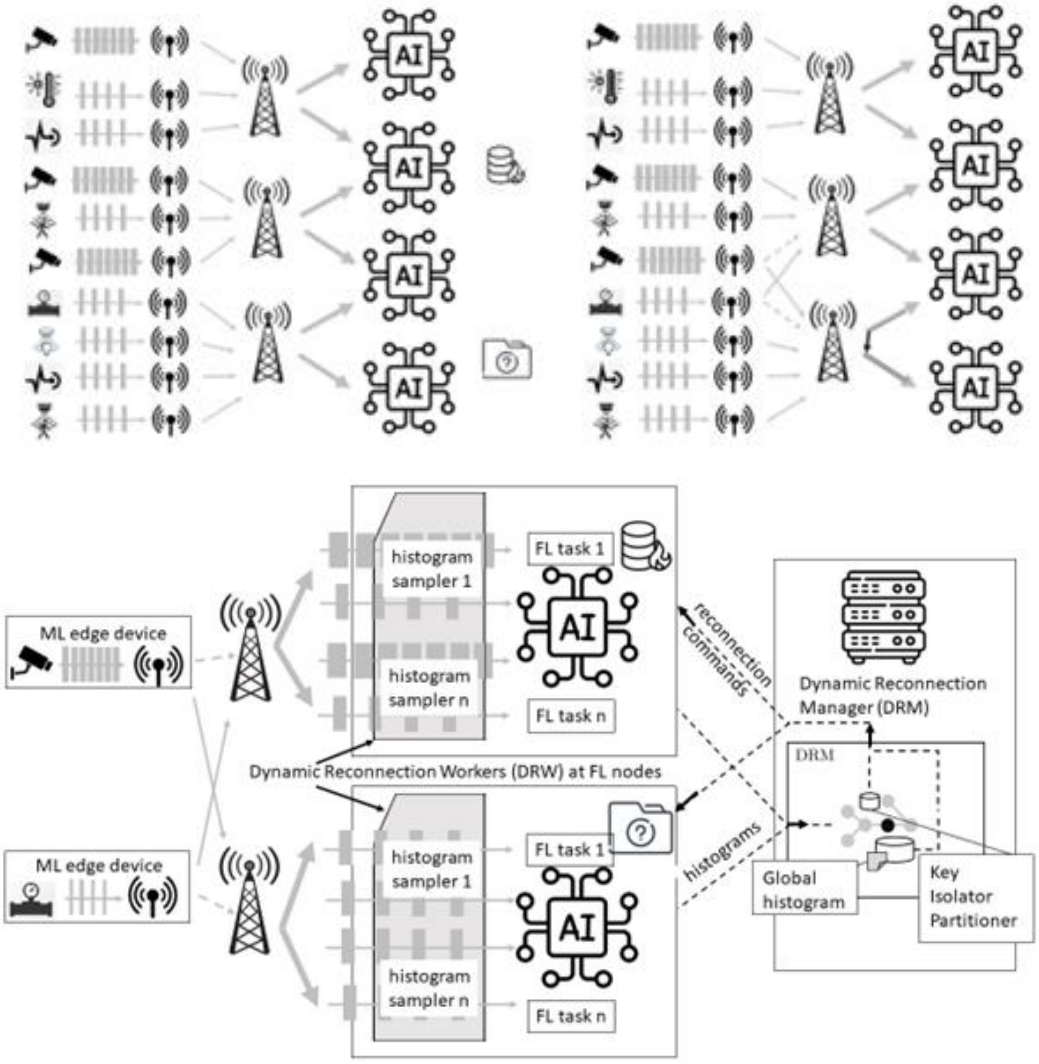


Figure 5-16 L**eft: a FL hot spot with too much data (top) and an insufficient data with one type (camera) missing (bottom). Right: a reconnection decision causes state migration between the bottom AI agents.**

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.

The dynamic reconnection method is based on our Key Isolator Partitioner (KIP) [ZSBB21], which is a heuristic combination of explicit placement and weighted hash partitioning to improve balance in cases of heavy data skew. KIP involves a distributed top-k histogram computation, where locations with heaviest load are ordered by decreasing frequency in a histogram object. The ideal maximal load of the partitions is calculated using a soft threshold to guarantee a good balance. In KIP, first the highest load is arranged greedily by considering radio accessibility. KIP attempts to keep UEs in their current connection to minimize migration costs, and non-heavy keys are handled by the weighted hash partitioner. The average load of a node is computed, and the uEs are rerouted as necessary by greedy

bin packing. KIP prepares for potential reconnection by making minimal modifications to the existing network state.

In our experiments following up work in Deliverable D4.2, we rely on the Simulation of Urban Mobility (SUMO) generator [http://sumo.dlr.de/index.html]. In our experiment, we generated 10,000 collaborating AI components (sensors and actuators) connected to 1-50 AI nodes. We compared the performance of KIP against random partitioning, Readj [G14] and Mixed [F+16]. We concluded that KIP gains 10-20% improvement over balance compared to the existing best method.

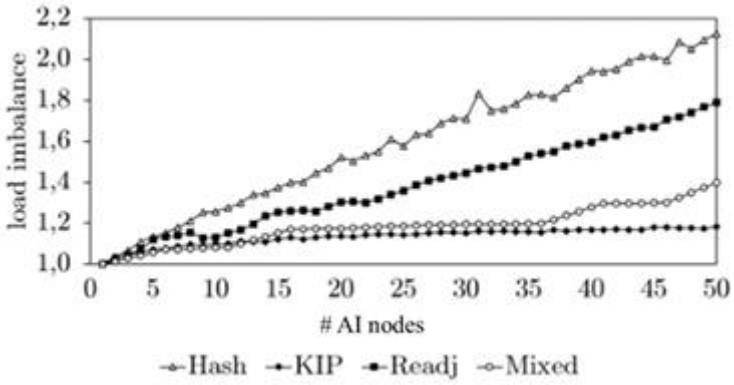


Figure 5-17: Performance of different rebalancing methods over simulation data.

We provide a mechanism for orchestration of edge AI by a dynamic and automated adjustment of distributed AI/ML system. We target the KPI family regarding the optimal assignment of resources (network and production) for training/inference time in a distributed AI system. By load balancing, we decrease latency while increasing AI agent density. Balanced connection structure improves both network energy efficiency and AI agent availability, thus also contributing to enabling Sustainability by energy-optimised services.

5.3.3 Frugal Federated Learning

Federated Learning (FL) is a popular technique for collaborative, multi-device over-the-air ML. Its main feature is that training data locally available at a contributing device are not centrally collected at a network node (e.g., an edge cloud server) for ML training purposes. There are several technical challenges when it comes to state-of-the-art FL implementations. Firstly, the FL training scheme takes place iteratively over-the-air, thereby resulting into increased communication signaling in each training round, both in uplink (for the upload of updated local ML models) and in downlink (for the broadcasting of the updated aggregated ML model). Such radio resource occupation may become extremely intensive in the case where the involved ML models consist of large model parameter sets (e.g., weights and bias values for a NN). Additionally, the exchange of large ML model parameter sets during FL training precludes that there is considerable processing, memory, storage and energy resources available both at the aggregator side and at the contributing devices themselves during the whole FL training procedure and also after it for inferencing purposes. This is, however, a non-realistic assumption, especially when the contributing devices are Internet-of-Things (IoT)-like ones with very limited compute platform capabilities. Even for more compute and energy capable devices, availability of computing and communication resources may significantly fluctuate, due to incoming workloads different from ones relating to FL training.

There are several state-of-the-art techniques proposed to achieve communication and storage-efficient FL training. One technique is ML model dropout, which consists in randomly dropping out a (fixed) fraction of model neurons before model transfer (e.g., for FL training or for transfer learning purposes) [SRI13]. A second technique is ML model sparsity-based compression. According to this technique,

(updated) neuron weights which are below a threshold (e.g., close to zero) are not communicated at all to the recipient (i.e., to the aggregator in case of FL model training). Such compression can be applied, e.g., in combination with model dropout or as standalone [ZS17]. A third technique is ML model parameter quantisation. For example, 8-bit representations are used instead of higher-resolution ones. This technique can be possibly applied also in combination with model dropout and model sparsity-based compression (for the“"survivin”" model parameters) [YBM20].

The state-of-the-art techniques, although, when applied, generally result to lower communication overheads and small inferencing accuracy loss, as compared to the case where no model pruning/ compression etc. is applied during FL model training, are generally applied *uniformly* (i.e., applying the same dropout rate and/or sparsity criterion and model parameter quantisation scheme) across all contributing devices. This way, several per-device characteristics are overlooked, therefore, limiting th92odelled92al of FL model training in terms of inferencing accuracy, model training time and End-to-End (E2E) energy efficiency. Examples of such per-contributor characteristics are: (i) availability of: computational/memory/ storage/ networking resources and connectivity options, energy resources (e.g., connected to the power grid or not, battery lifetime); (ii) *prior experience* in the specific task, e.g., prediction, classification, clustering, etc.–- availability of a relevant local ML model at the start of FL model training procedure and (iii) level of *significance of training data*, measured by the model“"concept drif”" during local ML model updating and/or by the change of -empirical- statistical distribution of learning data over time. In addition, another caveat of present solutions is that, when ML model pruning, sparsity-based compression and/or model parameter quantisation is applied, still quite a significant number of bits needs to be transmitted over-the-air during the whole FL model training phase. How to exchange local and aggregated ML model updates using only a few bits for (most of) training iterations is still an open issue, to the best of our knowledge.

In this section, the aim is to propose a solution framework for customising FL training to each contributing network entit’s (e.g., device, edge cloud server) available compute, communication and energy resources. Another aim is to minimise bi-directional communication signaling during FL training, as well as the risk of model theft over-the-air (e.g., by eavesdropping), while, at the same time achieving high accuracy of the resulting FL model during (local) inferencing. 6G use cases that may benefit from the proposed approach are those involving several devices of limited available radio/ storage resources (e.g., interacting and cooperative mobile robots). 6G KPIs of relevance are the ones of inferencing accuracy, latency, and E2E energy efficiency. The proposed solution aims at enhancing FL system scalability and also reducing energy consumption via more lightweight uplink/ downlink signaling during FL training stage.

5.3.3.1 Device capability-adaptive and low overhead over-the-air FL

Our proposed method for device capability-adaptive and low overhead over-the-air FL consists of the following solution components:

- A *data structure* (at application level) containing as attributes the characteristics (e.g., hardware, energy autonomy, software–- including available ML model- both static and dynamic) of a device contributing to FL model training that is periodically communicated to the FL aggregator (e.g., during each training iteration or on an event basis).
- A *base“"codeboo”" of ML models* initialised at the FL model aggregator, the cardinality of which can increase/decrease online and the members of which can be adaptively replaced, averaged, compressed, or decompressed per device capability changes and model training feedback per contributing device during FL model training. Basis model codebook adaptation results to per-device ML model codebooks maintained at the FL model aggregator side.
- A *ML model codebook lifecycle manager entity* instantiated both at each contributing device and at the FL model aggregator performing ML model codebook lifecycle operations.
- A *model similarity score* evaluating concept drift/ divergence between a ML model codebook member and a newly updated ML model, the latter issued either: i) on-device, based on local training data or ii) at the FL aggregator, based on ML model aggregation (e.g., averaging).

- *FL training methods* implementing the proposed device capability-adaptive ML model codebook-based FL training framework.

In the next subsections, more details of each of the proposed solution components are provided.

5.3.3.2 FL contributing device reporting to FL aggregator of its capabilities and learning status during FL training

During FL training, each contributing device provides a *device capability and learning status report* to the FL aggregator, e.g., by means of uplink transmission to a radio Access Point (AP) collocated with an Edge Server hosting the FL aggregator. The payload of this report message includes two parts. A first information element is a *FL device capability status report* containing information about the devic''s current computing availability and energy autonomy. For example, attributes of this message may include: current device availability of computational resources (Central Processing Unit–- CPU/ Graphics Processing Unit–- GPU/ Neural Processing Unit–- NPU/ FPGA etc.) and hardware acceleration capabilities, device memory resources, device storage resources, device energy resources (e.g., indicator showing whether the device is connected to the power grid or not, current expected battery lifetime), and device connectivity capabilities (e.g., multi- RAN connection capability).

A second information element, assuming an ML codebook-based approach for FL (to be later introduced in this section), is a *FL device learning status report* containing information relating to the FL training process, from the perspective of the contributing device. For example, attributes of this message may include: (i) For a given FL training round, indices of the FL device model codebook member of highest and possibly lowest similarity to a ML model updated by consuming local learning data, the latter ML model being updated asynchronously with respect to the FL learning process. Alternatively, in the lack of a local device ML model trained in standalone fashion (different from FL training procedure), these indices correspond to the FL device model codebook members that, after using -recently generated- local device data as a ML model testing set, showcase the smallest and possibly also largest testing error; (ii) For a given FL training round, *degree of similarity* between the indicated FL device model codebook member of highest and possibly lowest similarity to a ML model updated by consuming local learning data. Alternatively, in the lack of a local device model trained in standalone fashion (different from FL training procedure), value of this model distance can be replaced by the testing error based on using -recently generated- local device data.

The proposed device capability and learning status report may be provided by a FL device during FL model training, either: (i) in each FL training round, until FL model convergence, or, till a stopping criterion is met (e.g., maximum number of training rounds) or (ii) only on an event-based fashion until FL model convergence, e.g., only when the device capability changes significantly and/or when the index of the“"bes”"/“"wors”" member of the ML model codebook maintained at the device changes, as compared to the previous FL training round. Figure 5-18 below illustrates a possible implementation of the (uplink)“mess”ge“conta”ning the FL device capability and learning status reports to the FL aggregator for a given “L tr”in“ng ro”nd.

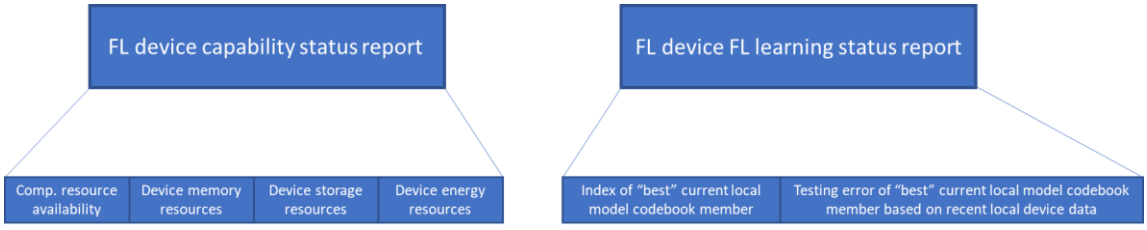


Figure 5-18: Proposed status report by each FL device–- for upload to the FL aggregator in each FL training round.

5.3.3.3 ML model codebook–- construction & maintenance operations

With the aim of reducing the bi-directional communication overhead in each FL training round, which involves exchanges of whole ML models (i.e., uploading locally updated ML models from contributing devices to the FL aggregator and the latter broadcasting the -current- aggregated ML model to all contributing devices) we propose a *ML model codebook-based approach for FL model training*. According to this approach, a ML model codebook is a set of ML models -the members of the ML model codebook-, all focused on the same task the FL model would address itself to (e.g., Quality-of-Service–- QoS prediction, image classification, etc.) and distinguished by their different parameter sets and, therefore, generalisation capabilities during inferencing. Members of an ML model codebook can be sourced in different possible ways. For instance, the FL aggregator and/ or (some or all) devices joining the FL setup may already have ML model codebooks available from previous FL training procedures on the same (or a very similar) task or host a single relevant ML model that has been locally instantiated and trained. Synchronisation in local and FL training procedures is not required.

Regardless of how a ML model codebook is sourced, either entity (FL aggregator or contributing device) may join (e.g., by means of subscription using a RESTful API) a FL setup offering its available -and, relevant to the task- ML model or its whole ML model codebook. This can be thought of as a“"bring your own ML mode”" (BYOM) criterion for admission to a FL setup by a FL manager or FL controller entity. From an ML capability monetisation standpoint, a network entity (e.g., a device user or an ML service provider) requesting subscription to a FL setup, may be incentivised to join when bringing a local ML model (or, a whole set of ML models), relevant to the task the FL setup aims to address. Means of incentive may e.g., be priority in compute resource usage, shared by other available network entities and/or a discounted fee to join the FL setup.

Since FL training is an iterative procedure during which: (i) the compute, communication, energy etc. capabilities of the contributing device, (ii) the amount and/or statistical distribution of learning data, (iii) the availability of a local ML model, relevant to the task, albeit trained asynchronously with the FL training procedure, may change significantly across FL training rounds, ML model codebooks should be also flexibly manipulated (i.e., compressed/ decompressed and/or increased/ reduced in the number of their members) in order to be customised to the capabilities of each contributing device during the FL training procedure. For this reason, two types of ML model codebooks are proposed: (a) a“*"bas”"* *ML model codebook* representing a specific training round, constructed by and maintained at the FL aggregator, based on feedback provided by a ML model codebook Lifecycle Manager (LCM), the latter instantiated e.g., at the same Edge Server as the FL aggregator or in a different entity. This ML model codebook represents a specific FL training round; (b) a *device-specific ML model codebook* for each FL training round (or, for each significant device capability change over time during FL training) which is constructed by a ML model codebook LCM instantiated in proximity to the FL aggregator and also transferred to the corresponding contributing FL device using periodic unicast/ multicast transmissions. Such a ML model codebook is a processed (i.e., compressed, encoded, reduced in size) version of the base ML model codebook, adapted to the capabilities of a specific contributing device. It is noted that, to produce a device-specific ML model codebook, state-of-the-art techniques can be applied for ML model compression/ encoding (e.g., ML model dropout, ML model sparsity-based compression, ML model parameter quantisation etc.).

Figure 5-19 shows an example focusing on the initialisation of a FL training procedure. For ease of exposition, two devices are involved in FL training, Device 1 with a local ML model and Device 2 without any local ML model available. In this example, the assumption is that the FL model aggregator already has a base ML model codebook available before the FL training procedure starts (e.g., carried over from another FL setup on the same or a similar task). Then, the steps to follow for the FL setup initialisation phase are the following:

Step 1: each of the joining devices sends its device capability status report and Device 1 also uploads its local ML model.

Step 2: The ML model codebook LCM at the FL aggregator side produces an updated version of the base ML model codebook, e.g., by replacing the ML model codebook member of highest age by the ML model shared by Device 1.
Step 3: The ML model codebook LCM at the FL aggregator side issues two compressed versions of the (updated) base ML model codebook, with compression factors tailored to the communicated processing, storage, memory and networking capabilities of Device 1 and Device 2.
Step 4: The compressed (device capability-adapted) ML model codebooks are transferred to Devices 1 and 2 by e.g., unicast downlink transmissions.

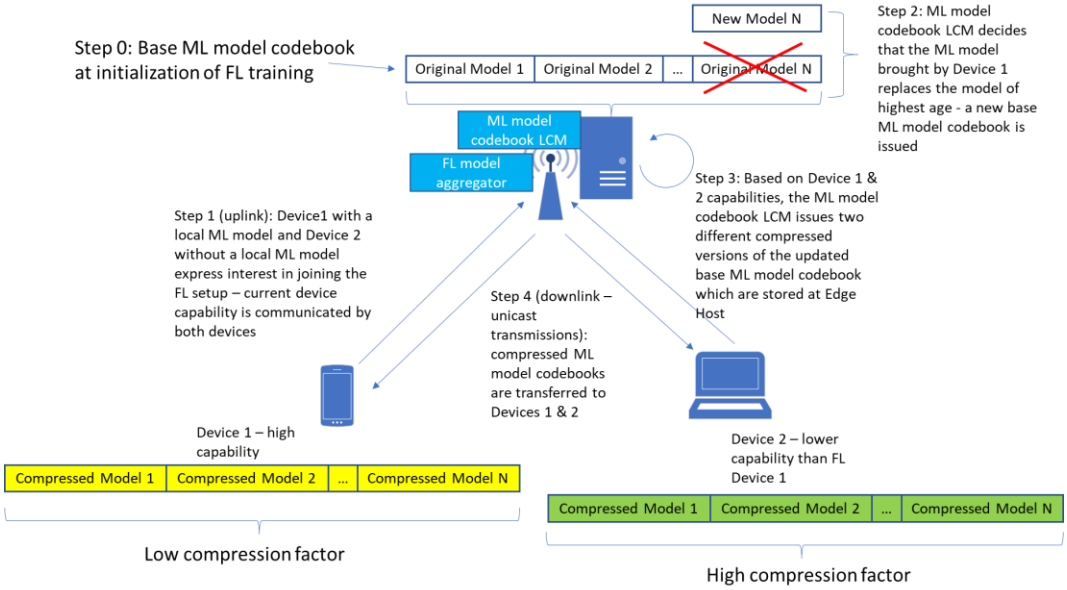


Figure 5-19: Example of producing different device-specific ML model codebooks, derived by a base ML model codebook and tailored to device capabilities–- FL initialisation/ establishment stage.

5.3.3.4 ML model codebook LCM and model similarity score

ML model codebook LCM: The ML model codebook LCM is an entity responsible for lifecycle operations of ML model codebooks during the FL training procedure. Such operations include, for example: ML model codebook member replacement, addition, removal, compression and decompression. This entity can be instantiated both at FL aggregator and FL device side. Figure 5-20 shows how this entity operates when deployed at FL aggregator side.

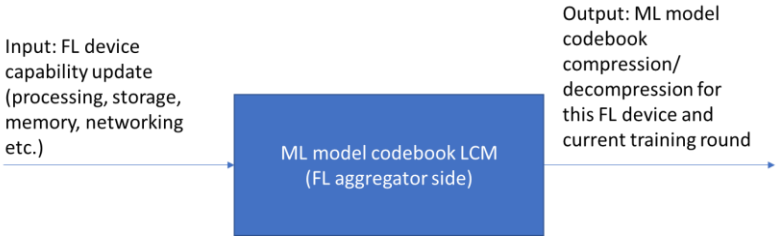


Figure 5-20: Exemplary operation of ML model codebook LCM at FL aggregator side.

Model similarity score: A model similarity score is a numerical value (e.g., a number between 0 and 1, or a percentage value) indicating the degree of concept drift/ divergence between a ML model codebook member and a newly updated ML model. An ML model codebook LCM can decide to perform ML codebook member LCM operations based on this score. Depending on where the ML model codebook LCM is instantiated, the updated ML model compared to the ML model codebook

members can be either obtained: i) on-device, based on local training data or ii) at the FL aggregator, based on ML model aggregation (e.g., averaging).

5.3.3.5 FL training methods applying the proposed device capability-adaptive ML model codebook-based FL training

Two methods for FL training can be implemented, based on the proposed device capability-adaptive ML model codebook-based FL training.

Method 1–-“"average best-scoring ML models and replac”": According to this method, after base ML model codebook initialisation at the FL aggregator side and transfer of device capability-adapted copies of it in the beginning of FL training, in each subsequent FL training round, only the index of each device-side ML model codebook member of highest similarity score (or of lowest testing error) is provided back to the FL aggregator side using uplink transmissions. Then, to produce an updated base ML model codebook for the next training round, the FL aggregator averages these“"best scorin”" ML models (using the FedAvg procedure of FL literature)–- which are already stored at the ML model codebook LCM from the previous round. Then, the ML model codebook LCM, in its turn, replaces a member of the current ML model codebook (e.g., the one of highest age) by the obtained average of best scoring ML models. The updated base ML model codebook is then distributed to the contributing FL devices leading to the next training round. FL model convergence is declared when e.g., two consecutive models obtained by the FedAvg operation have a similarity degree lower than a pre-defined threshold, or when a maximum number of iterations is reached. The advantage of this method is that FL device communication in the uplink will be of minimal payload (only a few bits representing the index of the best-scoring ML model codebook member).

Method 2–-“"score-based successive codebook size reductio”": This method differs from Method 1 in that the base ML model codebook size is not kept fixed during the whole FL training procedure, but it is successively reduced, based on a similarity score (or, testing error) performance criterion. According to this method, in each FL training round (after initialisation), each FL contributing device communicates to the FL aggregator the indices of the worst-scoring ML model codebook members. Then, at FL aggregator side, the ML model codebook LCM counts how many times, each ML model codebook member has been classified as worst (score-based) performing. The one of highest such count is removed from the updated base ML model codebook and the FL aggregator communicates the index of the removed ML model codebook member to the FL contributing devices; then, the ML model codebook LCM entities at the FL device side produce updated (and reduced in size) device ML model codebooks. This procedure keeps up until either the base ML model codebook is composed of a single member, or by a small number of members which can then be averaged to produce the FL model. The benefit of this method is that in both communication directions only ML model codebook member indices are exchanged. ML model parameters need only to be communicated over-the-air at the FL training initialisation stage.

6 Security, privacy, and trust in AI-enabled 6G

AI and ML will have distinct impact on future networks. They will be used as technical enablers to enhance the performance of the 6G networks and provide efficient services. Massive volumes of data will be processed by the network and its services in the 6G era. Thus, not only the AI/ML-based services themselves have to be safe and secure, but also the sensitive data of users that is used in these services have to be kept private. In this chapter, we focus on three aspects of trustworthy AI: security, privacy, and explainability. The security of AI/ML focus on identification and mitigation of attacks including poisoning, model evasion, and model extraction attacks targeted at the weaknesses of AI/ML-based

procedures in the networks to degrade the performance of the model causing it to behave unexpectedly. To this end, the AI system should be secure, robust, and safe throughout its entire life cycle. For privacy aspect, to have fully AI-aided digitalized world without data being compromised, anonymization and privacy protection technologies that achieve a balance between limited data exposure and legitimate analytics will be necessary. These technologies include homomorphic encryption, secure multi-party computation, federated learning, differential privacy, etc. For explainability aspect, it is important to understand and interpret predictions which are provided by machine learning models. To this end, AI explainability refers to the specifics and justifications a model provides to make its functioning trustable, clear and simple to comprehend [BLSS+20].

The concepts and technical enablers described in this chapter address the Trustworthiness KVI by providing security, privacy and explainability solutions. From security aspects, a use case scenario with adversarial evasion attack is shown to mislead the machine learning model into producing an incorrect output. The provided defence mechanism increases the robustness of the machine learning model against such attack. From privacy perspective, the privacy enhancing methods such as federated learning can already provide some level of privacy because the data remains on the individual devices or servers. However, these model update parameters can still disclose sensitive information. Privacy can be strengthened by using privacy enhancing methods for each individual device.

6.1 Security for AI-enabled 6G Networks

In order to provide a fully automated network, 6G rely on AI/ML technologies. Although AI/ML systems have a track record of success in wireless applications, they raise security challenges. According to recent studies [BNHGT+21], numerous adversarial attacks can be successfully conducted against DNN-based wireless systems. When compared to traditional wireless attacks like jamming [SRBE+11], adversarial ML-based attacks are more covert and difficult to spot because of their small footprints The two main categories of adversarial attacks against AI systems are evasion attacks and poisoning attacks, with the former occurring during the inference phase and the latter occurring during the learning phase. In the following, we will investigate the potential effects of adversarial evasion attacks targeting AI-driven power control systems in D-MIMO (distributed multiple-input multiple-output).

6.1.1 Adversarial evasion attacks in AI-driven power allocation

In D-MIMO, power control is crucial to optimize the spectral efficiencies of users and Max-Min Fairness (MMF) power control is a commonly used strategy as it satisfies uniform quality-of-service to all users. The optimal solution of MMF power control requires high complexity operations and hence deep neural network (DNN) based AI solutions are proposed to decrease the complexity. Although quite accurate models can be achieved by using AI, these models have some intrinsic vulnerabilities against adversarial attacks, where carefully crafted perturbations are applied to the input of the AI model. In this study, we planned to assess the degree of threats against targeted AI model, which might be originated from malicious users or radio units and observe the network performance by applying a successful adversarial sample even in the most constrained circumstances. For instance, we evaluate the success of the adversary under both white-box and black-box setting on spectral efficiency in the network.

For the system model, M single antenna RUs (Radio Units) are considered for a D-MIMO network which are connected to a central processor (CP) via wired fronthaul links. In a specific time/frequency resource block, we assume that K single antenna UEs are jointly served by all RUs. In Figure 6-1, we present an example D-MIMO network with potential attack resources where the network includes 16 RUs, 4 UEs, and a CP.

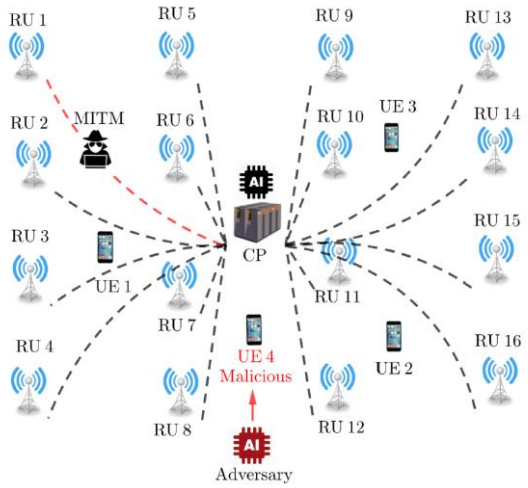


Figure 6-1: D-MIMO network with potential attack.

In D-MIMO network, we consider two source of adversarial attack scenarios, the first related to malicious rUs or fronthaul links, the second related to malicious UE. In the first scenario, a malicious RU can apply a man-in-the-middle (MITM) type attack to modify the channel related data transmitted from RUs to CP. In the second scenario, there may be some malicious UEs in the network that have modified RF and baseband components, causing them to apply some perturbation to pilot signals in order to change their related channel information. In TDD (time-division duplex) systems, uplink pilots that are received from uEs are used to gather channel information at the RU side. Consequently, a change in pilot signaling may result in incorrect channel knowledge at RU side. The effects of both these attack scenarios are analysed in this section using several simulations.

From the adversary's perspective, there are three important constraints which limits the success of the adversarial attack in a D-MIMO network. Firstly, the adversary mostly does not have access to the details (architecture and weights) of the original AI model, therefore cannot use it in a white-box setting for crafting adversarial samples. Secondly, the adversary may not have complete knowledge of the input features of the AI model. Because it is almost impossible for the adversary to know the channel information of each UE. Lastly, in a practical scenario, the adversary does not have the capability to introduce perturbations to all parts of the input vector, even if the channel information is known beforehand. However, despite all these limitations, there are proven ways in literature which increase the success of the attacker. Regarding the first limitation, it has been shown that a surrogate AI model might be sufficient to launch an effective attack due to the transferability nature of the adversarial samples [PMGJCS+17]. Regarding the other limitation, the universal adversarial perturbation (UAP) method is proposed for cases where the complete input knowledge is not available.

To craft a perturbation when there is partial input knowledge, we proposed a modified UAP (m-UAP) method [FKK+2023] which is based on the version suggested by Santos et al. [SMSL+22] where principal component analysis (PCA) is applied to a set of output vectors obtained by the attack method. In this study, we used Basic Iterative Method (BIM) [KGB+17] as the base attack algorithm when crafting the final perturbations within m-UAP method (Figure 6-2).

Algorithm 1: PCA-based modified UAP (m-UAP) method under L_∞ norm

Input: $\mathbf{x}, \epsilon$
Data: $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \widetilde{f}(\cdot, \boldsymbol{\theta})$
Output: $\boldsymbol{\delta}$

1 Define a matrix $\widehat{\mathbf{X}}^{N \times MK}$ using the known entries of the input $\mathbf{x}$ and the vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ taken from the test set. Firstly, initialize the matrix $\widehat{\mathbf{X}}$ as $\widehat{\mathbf{X}} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots \mathbf{x}_N]^T$. Then update $\widehat{\mathbf{X}}$ using the known entries of $\mathbf{x}$, i.e., $[\widehat{\mathbf{X}}]_{i,j} = [\mathbf{x}]_j$ for all known term indices j and for all $i = 1, 2, \ldots, N$.

2 For each row $\widehat{\mathbf{x}}_i^T$ of $\widehat{\mathbf{X}}$, apply the BIM to generate the matrix $\boldsymbol{P}^{N \times MK} = [\boldsymbol{\rho}_{\widehat{\mathbf{x}}_1}, \boldsymbol{\rho}_{\widehat{\mathbf{x}}_2}, \ldots, \boldsymbol{\rho}_{\widehat{\mathbf{x}}_N}]^T = [\nabla_{\widehat{\mathbf{x}}_1} J(\widetilde{f}(\widehat{\mathbf{x}}_1, \boldsymbol{\theta})), \nabla_{\widehat{\mathbf{x}}_2} J(\widetilde{f}(\widehat{\mathbf{x}}_2, \boldsymbol{\theta})), \ldots, \nabla_{\widehat{\mathbf{x}}_N} J(\widetilde{f}(\widehat{\mathbf{x}}_N, \boldsymbol{\theta}))]^T$

3 Compute the principal right singular vector $\mathbf{v}_1$ of $\mathbf{X}$ as $\boldsymbol{P} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H$, and $\mathbf{v}_1$ is the first column of $\mathbf{V}$.

4 Compute two perturbations $\boldsymbol{\delta}_1 = \epsilon \cdot \text{sign}(\mathbf{v}_1)$, $\boldsymbol{\delta}_2 = -\boldsymbol{\delta}_1$ and calculate the sum objectives corresponding to the perturbed inputs, i.e., $J_u = \sum_{i=1}^{N} J(\widehat{\mathbf{x}}_i + \boldsymbol{\delta}_u, \boldsymbol{\theta})$ for $u = 1, 2$. Find the index $u_0 \in \{1, 2\}$ such that $u_0 = \underset{u \in \{1,2\}}{\text{argmax}}\ J_u$.

5 $\boldsymbol{\delta} = \boldsymbol{\delta}_{u_0}$.

6 **return** $\boldsymbol{\delta}$

Figure 6-2: PCA-based modified UAP (m-UAP) method.

We perform several simulations to see the effects of adversarial attacks under several constraints. We begin our simulations by firstly showing the extreme scenario with the most devastating consequence where the adversary has access (read/modify) to all the parts of input vector fed to the AI model in CP. We set maximum allowed perturbation amount in channel information input vector as ϵ = 8 dB which is equal to the standard deviation of the shadowing in the system. In Figure 6-3, we show the CDFs (cumulative distribution functions) of per-user SEs (Spectral efficiency) under different attack types. It is clear that our proposed m-UAP attack results in much more devastating consequences than standard Gaussian perturbation. Under the attack, the performance of the analytical solution also degrades showing that adversarial training is not suitable for this regression task. Because applying adversarial training will degrade the natural (clean) performance of the system. This shows the necessity of developing new defence solutions for such kind of smart attack threats. Finally, we observe that surrogate model (indicated as blackbox) has slightly less disruptive performance than the original model (indicated as whitebox). This result proves that the adversary does not need to have access to the original AI model for crafting effective adversarial samples.

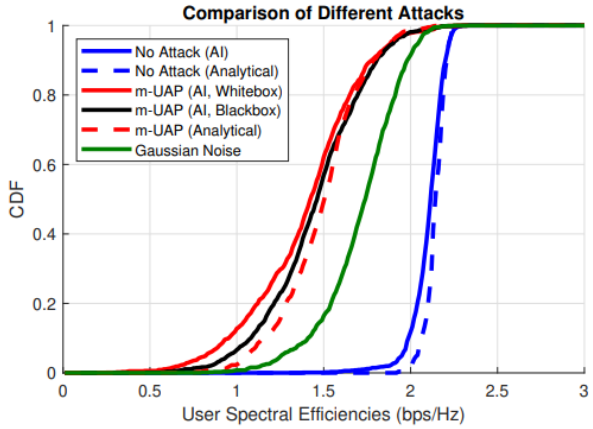


Figure 6-3: Comparison of effect of different attack types on SE (ϵ = 8 dB).

To cover more practical cases which might be seen in a real-world scenario, we assume that the adversary might have a partial channel knowledge and partial perturbation capability on the input. In below figures, we present the median (the SE value corresponding to CDF = 0.5) and the ^{5t}h percentile (the SE value corresponding to CDF = 0.05) per-user sEs for various levels of input information and perturbation capability. Figure 6-4 shows the results of the first type of attack scenario which might be originated by malicious RUs. Figure 6-5 shows the results of the second type of attack which might be launched by malicious UEs. In each of these cases, we assume that some portion of the uEs or rUs are malicious and only their channel information can be known and perturbed by these adversaries. In both scenarios, the adversary uses the surrogate model with ϵ = 8 dB.

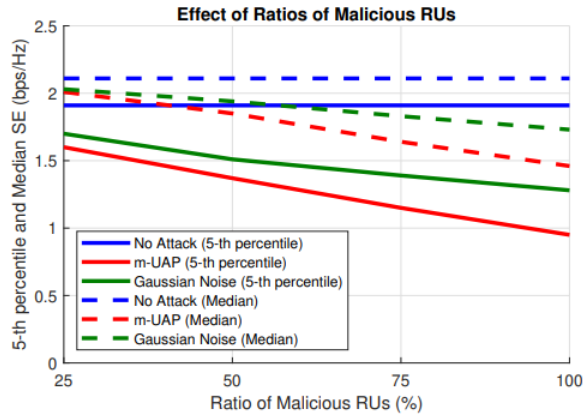


Figure 6-4: The effect of attack on RUs (ϵ = 8 dB).

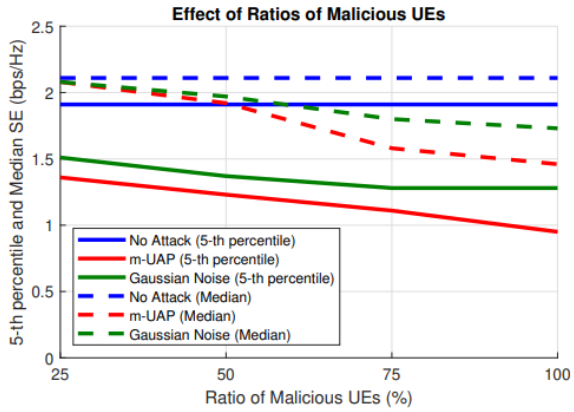


Figure 6-5: The effect of attack on UEs (ϵ = 8 dB).

According to the results obtained in above figures, we conclude that m-UAP attack outperforms conventional approach of applying standard Gaussian noise in all cases. The gap gets larger when the ratio of involved malicious actors in the network increases. When the level of information and perturbation capability about the input become larger, the adversary can degrade the performance more, as expected. In all these adversarial attack cases, we observe a significant decrease in the user SE performance of the D-MIMO network.

The results obtained so far assume max. Perturbation amount ϵ = 8 dB. Considering the shadowing standard deviation (which is equal to 8 dB), it is theoretically very hard for the system to detect an attack with ϵ = 8 dB as the probability of observing that much variation in large-scale fading coefficients is roughly 32%, which is not negligible. On the other hand, for ϵ > 16 dB, the same probability decreases down to 5% making the detection possible. In Figure 6-6, we observe the effect of ϵ for values up to 16 dB. We consider the scenario where half of the rUs are malicious and employ m-UAP method using a surrogate model.

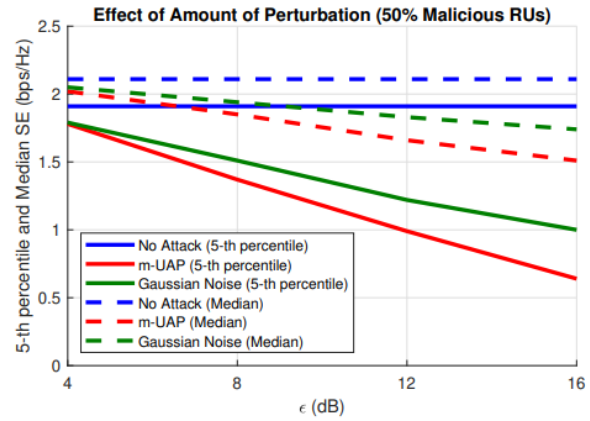


Figure 6-6: The effect of different ϵ values.

The results show that adversarial attacks are more effective than standard Gaussian noise, and the attack becomes more disruptive as ϵ increases.

Lastly, we consider the impacts of the adversarial attacks on the energy efficiency (EE) in the network as sustainability is one of the important concerns in 6G. The power consumption may differ under attack, so we need to analyze the EE separately. The average EE for various ϵ are analyzed for the case where half of the rUs are malicious and we employ m-UAP method using a surrogate model. In Figure 6-7, we see the comparison of no attack, adversarial attack and standard Gaussian noise attack cases. The adversary can decrease EE more than 8 percent in ϵ = 8 dB case and the effect becomes more severe

with increasing ϵ values. We conclude that adversarial attacks can also degrade the energy efficiency of the system.

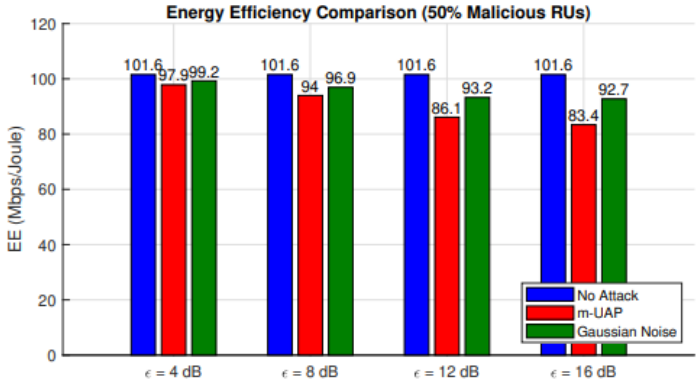


Figure 6-7: The effect on energy efficiency for various ϵ values.

In 6G era, we believe that use cases related to usage of AI/ML in wireless tasks such as beam forming and power allocation in D-MIMO may be more adversary-sensitive. The use cases under use case families of "Enabling sustainability"and "immersive telepresence for enhanced interactions" in the Hexa-x can be considered more adversary-sensitive use cases.

Defining a KPI to measure how much a system is secure or how much the security of the system is improved is quite hard. However, we can define some metrics to measure the adversarial attack success rate. The more the success rate, the more vulnerable the model against adversarial attacks. Thus, any proposed defence mechanism must reduce the success rate of such adversarial attacks. In the following, we suggest two KPIs for AI-driven D-MIMO task from both attacker's and defender's perspective.

$$Advtacksuccessrate = \frac{SE_{orig} - SE_{attack}}{SE_{orig}} * 100$$

where,

SE_{orig} is the spectral efficiency without any attack or defence (for ML), SE_{attack} is the spectral efficiency without any defence (for ML), and SE is the spectral efficiency obtained after defence.

In Figure 6-3, we present the 95%-likely per-user spectral efficiencies of users for various cases. Considering the values measured by the points corresponding to the 0.05 value of the CDF curves, when perturbation amount ϵ = 8 dB, the approximate success rate of adversarial attack is estimated as:

$$Adv_attacksuccessrate = \frac{1.95 - 0.77}{1.95} * 100 = 60.51\%$$

6.1.2 Defence mechanism to increase robustness of AI-driven power allocation against adversarial attacks

Adversarial attacks have the potential to seriously comprise the security of AI-powered systems and pose significant risks, particularly in fields like communications where security is of utmost importance. A number of defence strategies have been put forth in the literature to mitigate adversarial attacks, with adversarial training being one of the most popular ones. Yet, defensive strategies involving adversarial training are ineffective for the majority of regression problems and models. AI-driven power allocation for D-MIMO is one of these tasks where it is not possible to employ adversarial training to increase robustness of AI-driven systems. The reason behind is that applying perturbation to the channel information vector will lead to different optimum power allocation values in comparison to no attacks case, thus leading to poor performing models in normal conditions. In a numerical regression task which

is modelled by the analytical function f(.), mapping input vector β to output vector η, the perturbation applied to input β to get β' will not lead with the same result (i.e. $f(\beta) \neq f(\beta')$) as in the case of image classification tasks for example.

To mitigate the effects of adversarial attacks, we suggest a proactive defence approach. We propose a method in which we try to find an imaginary perturbation δ which yields a better power allocation output by maximizing the user SEs. By injecting a virtual perturbation, the method improves the output user SEs of the AI solution by optimizing the sum of the user SEs. Instead of using channel information vector (β′) to predict the output, we propose to iteratively compute the derivative of our objective function with respect to β′ and add the resulting accumulated gradient vector δ to β′ to predict the output. We compute the virtual perturbation vector δ iteratively. At each iteration, we compute the gradient $\partial SE_{AI,\,sum}/\partial\beta'$ via the AI model and update the virtual perturbation vector considering the direction of the gradient so that the $SE_{AI,\,sum}$ increases. After a finite number of steps, the algorithm converges, and we find the final δ vector. The details of our proposed defence algorithm is given in Figure 6-8.

Algorithm : The proposed defense algorithm.

Input: $\beta', \widetilde{f}(\cdot), i_{\max,2}, \epsilon, \alpha_2$
Output: $\boldsymbol{\delta}$

$\boldsymbol{\delta}_0 = \mathbf{0}$, $i = 0$.
while $i < i_{max,2}$ **do**
  Compute the AI output: $\boldsymbol{\eta}''_{\mathrm{AI}} = \widetilde{f}(\boldsymbol{\beta}' + \boldsymbol{\delta}_i)$.
  Compute the SE vector: $\mathrm{SE}''_{\mathrm{AI}} = q(\boldsymbol{\beta}', \boldsymbol{\eta}''_{\mathrm{AI}})$.
  Compute the sum of SEs: $\mathrm{SE}''_{\mathrm{AI,\,sum}}$ is equal to the sum of the elements of $\mathrm{SE}''_{\mathrm{AI}}$.
  Update the virtual perturbation vector: $\boldsymbol{\delta}_{i+1} = \mathrm{clip}_{\epsilon}\left(\boldsymbol{\delta}_i + \alpha_2 \cdot \mathrm{sign}(\partial \mathrm{SE}''_{\mathrm{AI,\,sum}}/\partial\boldsymbol{\beta}')\right)$.
  Check if the algorithm is converged: If the difference between $\boldsymbol{\delta}_{i+1}$ and $\boldsymbol{\delta}_i$ is low enough, terminate. Otherwise increase i by 1.
return $\boldsymbol{\delta} = \boldsymbol{\delta}_i$.

Figure 6-8: The proposed defence algorithm.

The effectiveness of the proposed defence technique is verified through experimental results. Figure 6-9, involves CDF under different attack scenarios for ϵ = 4 dB with and without our proposed defence solution. We observe that the proposed defence technique significantly enhances the performance for UEs with low SE values. With the proposed technique, 0.28 bps/Hz enhancement on 5th percentile per-user SEs is obtained. Considering that the remaining gap to AI solution without any attack is equal to 0.19 bps/Hz, it can be concluded that more than half of the performance loss due to adversarial attack can be regained by the proposed defence method. When there is no attack or there exists a WGN attack, a slight improvement is obtained over AI solution without any defence. This shows that the proposed method can be used without any performance loss regardless of the attack situation. The proposed defence method is effective against adversarial attacks which are the most disruptive ones for AI-based power allocation methods

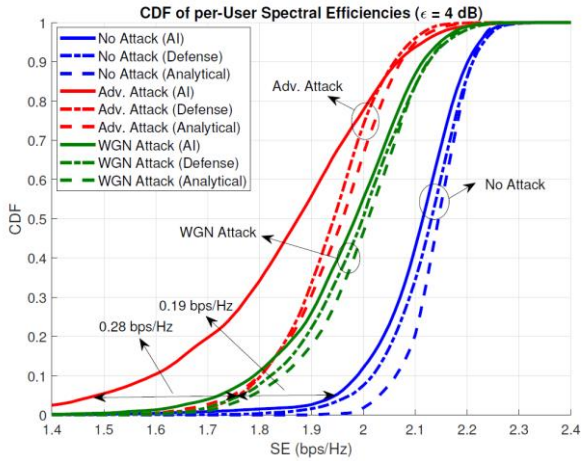


Figure 6-9: CDF of oer-user Ses for various cases (ϵ = 4 dB).

To measure the robustness of the proposed defence method, we define a metric $r_{robustness}$ as:

$$r_{robustness} = \frac{SE_{defense} - SE_{attack}}{SE_{no_attack} - SE_{attack}}$$

where SE_{no_attack}, SE_{attack} , $SE_{defense}$ indicates 5th percentile per user sEs of AI solution without attack and defence, under adversarial attack without defence, and under adversarial attack with the proposed defence method, respectively. $r_{robustness}$ shows the ratio of performance that can be regained by the defence method for UEs with low SE values.

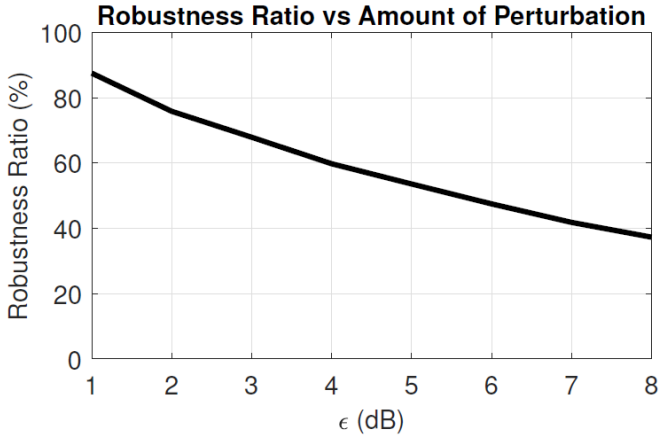


Figure 6-10: $r_{robustness}$ for various ϵ values.

In Figure 6-10, we illustrate the robustness ratios for various ϵ values. We can observe that for all $\epsilon \leq$ 8 dB, the robustness ratio values are greater than 40%. The robustness ratio is a decreasing function of ϵ, because as the perturbation size rises, it gets harder to offset the effects of adversarial attacks. It is important to note that at ϵ values greater than 8 dB, which is the shadowing standard deviation, the system may be able to identify an attack by spotting irregular changes in the channel coefficients and take appropriate action.

Applying defence method, we expect increase in robustness of the system and decrease in adversarial success rate. According to Figure 6-10, the robustness of the proposed defence method, when perturbation amount ϵ = 8 dB, is estimated as:

$$r_{robustness} = \frac{SE_{defense} - SE_{attack}}{SE_{no_attack} - SE_{attack}} = \frac{1.21 - 0.77}{1.95 - 0.77} = 37.29$$

And the adversarial success rate is estimated as:

$$Adversarial\ attack\ success\ rate = \frac{1.95 - 1.21}{1.95} * 100 = 37.94\%$$

We can observe that attack success rate is decreased from 60.51% to 37.94%, which mean (60.51–-37.94) / 60.51= 37.29% drop in success rate of attacker.

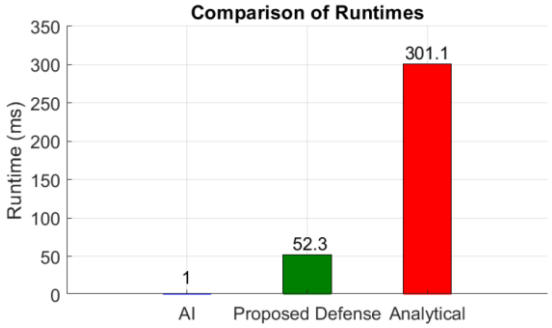


Figure 6-11: Comparison of runtimes of AI, proposed defence, and analytical solutions.

In Figure 6-11, we compare the average computational cost of the proposed defence method with analytical, and AI solutions. As expected, the proposed defence technique is slower that the AI solution, but it is roughly 6 times faster than the analytical solution.

6.2 Privacy for AI-enabled 6G Networks

As network complexity keeps expanding, the reliance on artificial intelligence (AI) and automation solutions is growing quickly in the telecom sector. The machine learning (ML) models that many mobile network operators (MNOs) use to forecast and address problems before they have an impact on user QoE are one example [VIDBA+19]. To create an ML model, data from different sources is transferred to a central server which can be problematic from a data security perspective. Thus, solutions be able to create an ML model without needing to transfer voluminous amounts of data, perform centralized computation and/or risk exposing sensitive information are needed. Federated learning (FL), with its ability to do ML in a decentralized manner, is a promising approach. In FL setting, only the local model updates are shared with the server to create a global model which help users to keep their local data private. Although this technique increases the privacy of user data, the local model updates may still leak information about each client. Bellow, we explain in more detail about the problem and the alternative solution.

6.2.1 Security mechanism friendly privacy solutions for federated learning

Although FL enhances the privacy of clients by enabling each client to train its own local data, there is still a possibility of information leakage when clients send local model updates to server to aggregate to a global model. Generally, secure aggregation techniques are used to prevent this kind of data leaking so that the server can only get the aggregated result rather than the individual local model updates. However, using such mechanism may prevent server to analyse clients local model updates to detect security attacks to model training such as poisoning and backdoor attacks. Thus, there is trade-off between providing privacy and security in FL setting, and solutions which provide privacy and allow server to detect anomalies resulted from security attacks are demanding. The 6G use cases in Hexa-X which expected to be benefited from this solution include any use case where multi agents using ML/AI try to learn a model or cooperatively operate in a private manner. Some of such use cases can be “interacting and cooperative mobile robots”, “flexible manufacturing”, “fully merged cyber physical systems”, and “immersive smart cities”. In [HEX-D42], we provided a security-friendly privacy solution for federated learning, which was based on multi-hop communication to hide identities of clients but ensured that the forwardee clients in the path between the source client and the server could

not execute malicious activities such as altering model updates and sending more than one model update. In this deliverable we provide two new enhancements on top of the previous solution to make it robust against possible malicious packet drop behaviours by the forwardee clients.

In order to detect malicious forwardee clients who drop packets, we introduce and utilize a detector entity to the network which monitors packet exchanging procedure between a sender and a forwardee to find whether the process is fully accomplished or not (Figure 6-12). In this solution, when a sender client sends a packet including a local model update to a forwardee client, it also sends a message to the detector. The message contains sender client ID, forwardee client ID and hash of the packet sent to the forwardee client. By this way, the detector tracks the information attached to the message to ensure there's no packet drop. Accordingly, whenever a forwardee client receives a packet, it sends a message to the detector with the same information, forwardee and sender client IDs and hash of the packet, to inform a packet has been received. Then the detector checks the messages and if they match, it sends an ACK packet to the sender client which means the packet transmission has been completed successfully. Otherwise, if sender client doesn't get an ACK message from the detector, sender client executes same steps for a randomly chosen forwardee client. The Figure show example interactions between server, one client, and detector.

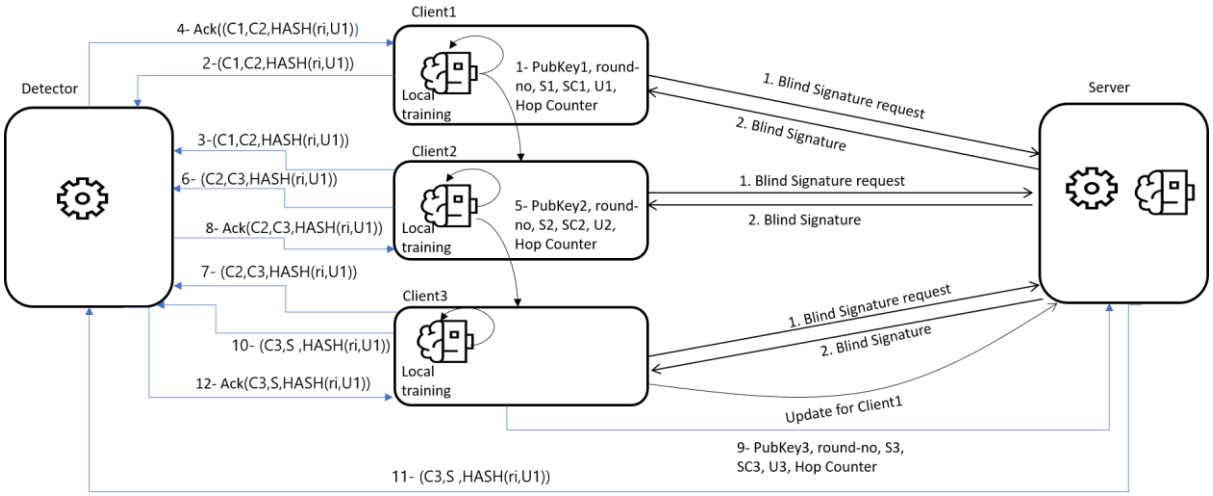


Figure 6-12: Example interactions between server, clients and detector. Note that only the packet flow for client1 data is shown as an example.

Note that in the solution it is assumed that the detector and the server do not collude. The detector does not get the local model update packets however knows all the traffic by observing the messages from forwardee and receiver clients. Other clients and the server are not aware of this traffic since detector would not share this information.

Defining a KPI to measure how the privacy of data in the provided solution is improved is difficult. However, we can define KPIs to measure the utility of the provided solution in term of accuracy. This measure demonstrates how much the defence mechanism disrupts the normal behaviour of the model in term of accuracy. We can also measure the total overhead of the defence mechanism from communication and computation point of view with respect to the mechanism without defence.

$$Utility = \frac{Accuracy}{Accuracy_{noDefense}} * 100 > 90\%$$

$$Overhead_{Total_{comm_{comp}}} = \frac{Delay - Delay_{noDefense}}{Delay_{noDefense}} * 100 < 30\%$$

To assess the computation overhead our approach adds to the FL, we used a common user laptop equipped with an Intel Core i5 CPU and 32 GB RAM to implement our protocol using Python. We used

the Python implementation[2] of Abe and Okamot''s partially blind signature scheme, which was first described in [[AO+00]]. We preferred (2048-bit, 224-bit) as the DSA parameter lengths for 112-bit security and modelled the clients and the server as sequentially called functions. In typical FL with 20 clients, the local model update size is roughly 4 MB, and local model training for each client takes an average of 7 seconds. We assessed the computation time of our approach, excluding the local training part, while taking the local model size into consideration. One partially blind signature procedure takes 90 ms to complete on the client and server sides when our protocol is used. The signature operation performed by each client is the other significant activity in terms of calculation time. This signature operation took 10 milliseconds. The final significant operation, performed by the server, took 25 milliseconds to validate the signatures on the local model updates and the client'' public keys. As a result, the overall computation overhead of our protocol for client and server operations is 125 ms. Given that each clien''s local model training requires 7 seconds (7000 ms), our protoco''s overhead is minimal (1.8% longer execution time).

We also consider the communication cost overhead of sending extra messages for the execution of proposed protocol. It takes 61 ms in a network with a throughput of 524.29 Mb/s when we just consider delivering local model updates from the client to the server that are 4 MB in size. Local model update transmission takes 6100 ms when we have network throughput of 5.24 Mb/s. Given that local model training takes 7 seconds, it is clear that communication time with a not good throughput becomes a significant time factor. Fortunately, when the client and server have greater connection capabilities, the communication time is much shorter than the computation time. Although multi-hop communication requirements result in significant communication delays in case the network have low bandwidth and high RTT, these can be ignored in the better network setting. Figure 6-13 illustrates this conclusion where the increase in communication speeds, decrease the overhead of our method

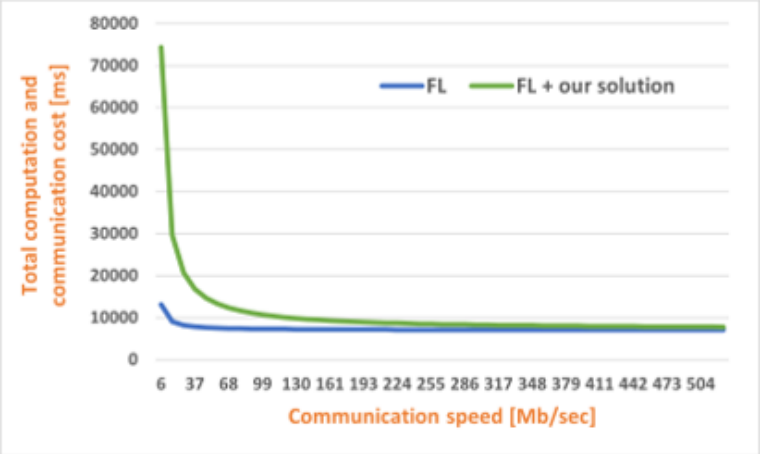


Figure 6-13: Overhead of our solution depending on the communication speed.

For the KPI/KVI, the utility of the proposed method in term of accuracy will not change with respect to the typical FL. In case of overhead, the proposed method has 10.76 % increase in comparison with typical FL when communication speed is 504 Mb/sec.

$$Overhead_{Total_{C}omm_{C}omp} = \frac{7824.067 - 7063.552}{7063.552} * 100 = 10.76$$

[2] https://github.com/cowlicks/partially/blob/master/partially.py

6.3 Explainable AI

Transparency of ML models is one of the fundamental requirements towards trustworthiness and is at the core of the explainable AI (XAI). In this section, the trade-off between accuracy and interpretability of inherently interpretable models is discussed, and an approach for FL of such models is introduced.

6.3.1 XAI models: Fuzzy regression trees and TSK Fuzzy Rule Based Systems

Rule Based Systems (RBSs) and Decision Trees (DTs) are generally considered more inherently interpretable than other popular "opaque" models (e.g., Neural Networks and Random Forest) and still able to achieve competitive performance, especially when data comes in tabular form.

Among the ML-based approaches commonly employed to solve regression problems, Takagi-Sugeno-Kang Fuzzy RBSs (TSK-FRBSs) and Fuzzy Regression Trees (FRTs) are particularly relevant in the XAI domain, as they feature both high modelling capability and interpretability. As per the latter, the inference process of TSK-FRBSs and FRTs, is in fact equivalent to the application of simple *if-then* rules, that can be regarded as very much akin to human reasoning. Furthermore, the adoption of linguistic representation of numerical variables allows a direct human interaction, further enhancing the model interpretability.

However, accuracy and interpretability are two conflicting objectives and investigating the trade-off between them becomes crucial even with respect to model configuration choices. In the following, an overview of the two models is provided along with some empirical results on benchmark datasets.

FRTs are the regression counterpart of Fuzzy DTs for classification, described in a previous deliverable [HEX-D42, Section 5.2.1]. The sequence of tests (modeled by inner nodes) partitions the input space into subspaces that contain subsets of training set. In regression problems, the rationale is to obtain subsets in which instances have output values as close as possible to each other. Since the goal is the prediction of a real value, leaf nodes are characterized by a regression model defined on the input variables.

Notably, two configuration parameters deeply affect interpretability and accuracy of FRTs:

- The inference strategy, namely *maximum matching* (**MM**) (among the multiple paths activated by an input pattern, consider only the one with the highest strength of activation) or *weighted average* (**WA**) (average the output produced by multiple paths, weighted based on their strength of activation). Notably, MM ensures a higher level of local interpretability [HEX-D42, Section 5.2.1].
- The order of the polynomial model used in the leaves, namely zero-order (**0**) (constant value) or first-order (**1**) (linear model).

An experimental analysis has been carried out to evaluate the performance obtained with the possible combination of these parameters, which lead to four variants of FRT, namely FRT-MM-0, FRT-MM-1, FRT-WA-0, FRT-WA-1. [BCD+22]. They have been tested on several regression dataset using 5-fold cross-validation. Table 6-1: summarizes the results.

FRT-	MSE Train	STD Train	MSE Test	STD Test	MSE Train	STD Train	MSE Test	STD Test
	Weather Izimir [WI] (Features: 9 – Samples: 1461)				**Mortgage ($\times 10^{-3}$) [MO]** (Features: 15 – Samples: 1049)			
-MM-0	27.56	6.83	27.98	6.31	635.03	71.88	620.32	125.12
-WA-0	14.55	0.29	14.89	1.14	303.06	1.05	310.16	24.13
-MM-1	1.31	0.07	1.40	0.30	5.90	0.32	8.31	1.84
-WA-1	1.22	0.07	1.32	0.28	5.12	0.20	6.90	1.36
	Treasury ($\times 10^{-3}$) [TR] (Features: 15 – Samples: 1049)				**California ($\times 10^{9}$) [CA]** (Features: 8 – Samples: 20460)			
-MM-0	789.32	66.74	844.54	91.02	8.56	0.08	8.57	0.16
-WA-0	398.72	6.31	407.18	73.62	9.54	0.03	9.54	0.14
-MM-1	31.49	4.29	43.03	19.13	4.25	0.06	4.28	0.19
-WA-1	30.45	4.14	39.05	19.67	4.11	0.03	4.15	0.14

Table 6-1: Average MSE and standard deviation over cross-validation for each dataset and for each variant.

It can be observed that the adoption of a first-order polynomial model in the leaves leads to better results than the constant models. Furthermore, the adoption of a maximum matching approach does not particularly degrade the accuracy of FRTs compared to the weighted average strategy. The proposed FRT variant with first-order polynomial model in the leaves and maximum matching as inference strategy (indicated as FRT-MM-1 in Table 6-1) represents an effective solution for applications in which high accuracy and high explainability are required.

An example of rule, extracted from an FRT induced on *California* benchmark dataset, is reported in the following: IF MedianIncome is Low AND Latitude is Low AND Longitude is Medium THEN : MedianHouseV alue = 0.89–− 1.10 · Longitude–− 1.03 · Latitude + 0.10 · HousingMedianAge–− 1.56 · TotalRooms + 2.08 · TotalBedrooms–− 2.33 · Population + 0.41 · Households + 1.27 ·MedianIncome. Notably, we can characterize the impact of each attribute on the output value produced by the local linear model.

As for FRTs with first-order polynomial models, in a TSK model the antecedent of a rule identifies a specific region of the attribute space, whereas the corresponding consequent evaluates the output within such a region as a linear combination of the input variables. The rule base, however, is generated in a different manner: first the number of rules and the conditional part of the rules are determined; this is typically done either with grid-partitioning of the input space or exploiting fuzzy clustering methods. Then, local linear models are fitted on data by pseudo-inversion or by a least square method.

Our approach to enforce interpretability in TSK-FRBSs [CDE+22] differs from state of art approach for building TSK-FRBSs [FSN+20] in two aspects:

- inference process is based on maximum matching, instead of weighted average,
- rule antecedent parameters based on a priori grid partitioning, instead of exploiting a data-driven methodology based on clustering over the input-output product space.

The impact of these design choices is evaluated on several benchmark datasets, by comparing the state of art approach [FSN+20] with the proposed model in case of MM (TSK-MM) and WA (TSK-WA). Table 6-2 summarizes the results.

	TSK-MM		TSK-WA		TSK [FSN+20]	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
WI	1.28	1.38	1.28	1.37	1.48	1.52
TR	24.06	40.30	24.42	39.18	32.07	62.93
MO	3.99	6.36	4.29	6.14	4.49	8.22
CA	4.78	4.81	4.82	4.85	4.62	4.64

Table 6-2 Comparison of our approach (TSK-MM = maximum matching, TSK-WA = Weighted Average) and a state of art approach [FSN+20] for building TSK-FRBSs, in terms of MSE.

Results suggest that the performance of the proposed approach is comparable to, or even better than, the one obtained in [FSN+20]. Furthermore, as observed in the context of FRTs, the inference process based on WA only slightly outperforms the one based on MM, but at the cost of a lower semantic interpretability.

The analysis of the trade-off between accuracy and interpretability of the XAI models (FRT and TSK) supports the Fed-XAI technical enabler (See Section 5.3.2), which, in turn, can be regarded as an enabler for several families of use cases envisioned for 6G. Notably, inferencing accuracy represents the most relevant KPI, which must be pursued together with the KVI of explainability, deemed as a crucial requirement for trustworthiness. Model complexity can be considered as a proxy for the interpretability level.

6.3.2 Fed-XAI: Federated Learning of Explainable AI models

An approach for Federated Learning of TSK-FRBSs is proposed [CDE+22]. It involves multiple parties that collaboratively learn a model under the orchestration of a central server and without exposing their private data to others. It encompasses the following steps (an overview is reported in Figure 6-14):

- Communication step A: configuration of the learning process;
- Step 1: local learning of TSK-FRBSs. Each client trains a model based on its local data;
- Communication step B: transmission of local models to the central server;
- Step 2: federated learning of the global TSK-FRBS: aggregation of the models;
- Communication step C: transmission of the aggregated model to the clients;

The aggregation step is detailed in the following: first, the server generates a rough global model (i.e., a global rule base), as the juxtaposition of the rules collected from the clients. Since this global rule base aggregates knowledge from different sources, the rule base must be refined by resolving conflicting rules, namely rules with the same antecedent but different consequents.

Let CR be the set of conflicting rules for a specific antecedent. Moreover, each rule as an associated rule weight, computed as the harmonic mean of the rule support (i.e., how much a rule is activated), and rule confidence (i.e., average quality of the prediction of the rule) as measured on the training set. A single rule is obtained from CR as follows:

- the new antecedent is the same of the rules in CR;
- the coefficients of the new consequent are estimated as the weighted average of the coefficients of the consequents in CR, each weighted by the respective rule weight;
- the rule weight associated with the rule is computed as the average of the rule weights in CR.

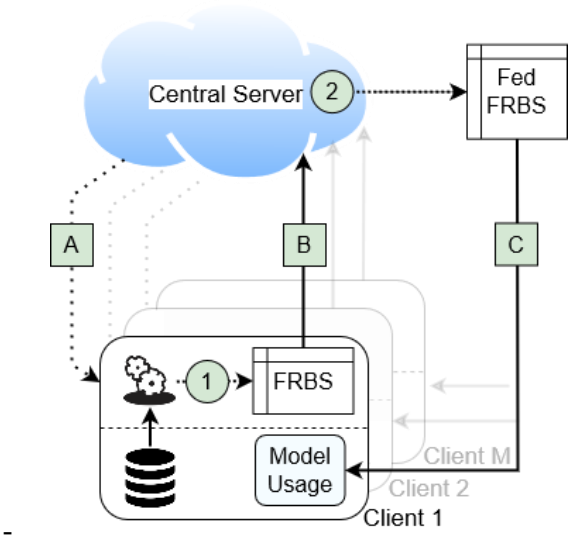


Figure 6-14. Overview of the proposed approach. Squared markers (A, B, C) denote communication steps. Circle markers denote local learning (1) and model aggregation (2) steps. Figure from [CDE+22]

The proposed approach for federated learning of TSK-FRBSs is experimentally evaluated on benchmark datasets by considering data scattered over 5 clients and is based on the comparison between three scenarios (schematized in Figure 6-15): The *federated* one is compared with the *centralized* and the *local* setting. In the centralized setting each participant shares its training data with the central server for building an FRBS based on the overall training set. In the local setting each client learns and tests the local model locally, thus ruling out any form of collaboration.

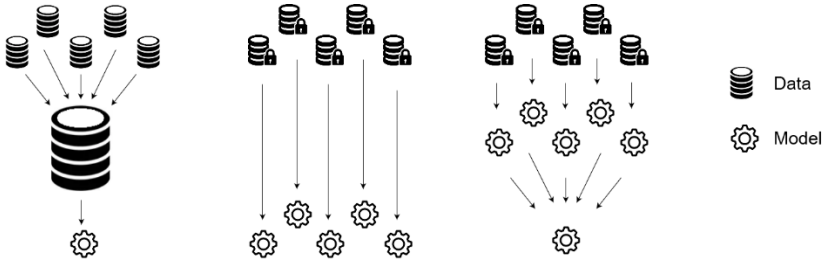


Figure 6-15. Three learning scenarios: (left) Centralized. (center) Local. (right) Federated.

Table 6-3 reports the experimental results.

Results suggest that federated scenario always outperforms, on average, the local one. This is particularly relevant as it demonstrates the benefit of the FL process for participating clients. On the other hand, the centralized scenario achieves comparable or better performance than the federated one, but it is typically not viable due to privacy issues and/or communication constraints.

	Local		Federated		Centralized		Local		Federated		Centralized	
Client ID	Train	*Test*	Train	*Test*	Train	*Test*	Train	*Test*	Train	*Test*	Train	*Test*
	Weather Izimir						**Mortgage ($\times 10^{-3}$)**					
1	1.33	*2.02*	1.44	*1.57*	1.40	*1.54*	2.29	*78.08*	9.70	*15.96*	5.20	*7.55*
2	1.09	*1.62*	1.25	*1.41*	1.22	*1.34*	1.44	*15.08*	9.14	*7.35*	3.47	*5.22*
3	0.96	*1.40*	1.25	*1.32*	1.22	*1.29*	1.22	*38.18*	14.61	*9.52*	3.31	*5.22*
4	1.07	*7.10*	1.23	*1.30*	1.20	*1.28*	1.54	*53.84*	9.38	*35.90*	4.24	*8.83*
5	1.19	*1.64*	1.41	*1.51*	1.38	*1.46*	1.09	*43.36*	14.78	*5.14*	3.74	*4.98*
Avg.	1.13	*2.76*	1.32	*1.42*	1.28	*1.38*	1.52	*45.71*	11.52	*14.77*	3.99	*6.36*
	Treasury ($\times 10^{-3}$)						**California ($\times 10^{9}$)**					
1	7.11	*377.40*	82.20	*112.72*	21.97	*46.13*	4.73	*4.87*	4.75	*4.86*	4.77	*4.78*
2	19.28	*192.70*	53.64	*79.41*	37.69	*51.355*	4.62	*4.73*	4.57	*4.58*	4.60	*4.62*
3	7.72	*337.25*	429.38	*174.18*	26.86	*41.97*	4.71	*4.89*	4.71	*4.74*	4.72	*4.75*
4	9.31	*110.47*	72.86	*378.61*	20.51	*41.69*	4.77	*5.10*	5.23	*5.34*	5.18	*5.24*
5	10.37	*133.83*	57.04	*40.85*	13.24	*20.37*	4.70	*4.82*	4.63	*4.64*	4.65	*4.68*
Avg.	10.76	*230.33*	139.02	*157.15*	24.06	*40.30*	4.71	*4.88*	4.78	*4.83*	4.78	*4.81*

Table 6-3: Experimental results: for each dataset and scenario, the average MSE over cross-validation is reported for each client, along with the overall average values.

As for 6G KPIs and KVIs, the proposed solution targets inferencing accuracy and explainability, respectively, and it can be considered as an enabler for several use cases families. As an example, it has recently been proposed as an enabling technology in 6G systems for an automated vehicle networking use case [RDM+22].

7 Demonstration activities–- Federated eXplainable AI (FED-XAI) demo

The Fed-XAI Demo#2 activity is carried out jointly in Work Packages 4 and 5. The main modules of a UE application with Fed-XAI capability and its functional requirements have been described in [HEX-D42]. The considered automotive scenario and the simulation testbed have been described in [HEX-D51] and [HEX-D52]: several instances of vehicular User Equipment (UE), connected to a B5G/6G network, receive (or send) a video stream, whose perceived quality is crucial for the availability of advanced driving assistance systems, such as see-through (or tele-operated driving). The Fed-XAI model aims at forecasting the perceived video quality based on the available contextual, QoS and QoE metrics; furthermore, the Fed-XAI model is learned in a collaborative fashion and is highly interpretable by design. In this section we first describe the implementation details of the proposed framework and then discuss the results obtained on the above-mentioned simulated scenario.

The experimentation described in this deliverable concerns a scenario compatible with a see-through service. The investigation of a scenario compatible with tele-operated driving, in which the mobile network simulation is dimensioned according to data gathered from a live RAN, is currently under development.

7.1 Fed-XAI framework: implementation details

The Fed-XAI framework allows multiple users to collaboratively train an XAI model, under the orchestration of a server in a centralized communication topology, and then to perform inference over local data. The system also provides a dashboard to enable users, upon authentication, to visualize the model prediction and uncover the reason behind the model decision. The framework is compliant with an edge computing environment and exploits a fully virtualized architecture: each module is deployed inside a docker container, to allow portability of each module regardless the underlying hardware and software infrastructure. Communication among the different modules is implemented through messages exchanged via RESTful APIs over Secure HTTP protocol (HTTPS) as a security mechanism.

The actual FL process exploits the Intel OpenFL library [FSE+2022], which has been extended to support FL of inherently interpretable models, such as FRT and TSK FRBS (as discussed in Section 5.3.2). A Repository module, which manages the storing of users' information and trained global models, has been implemented as an instance of MongoDB in a docker container.

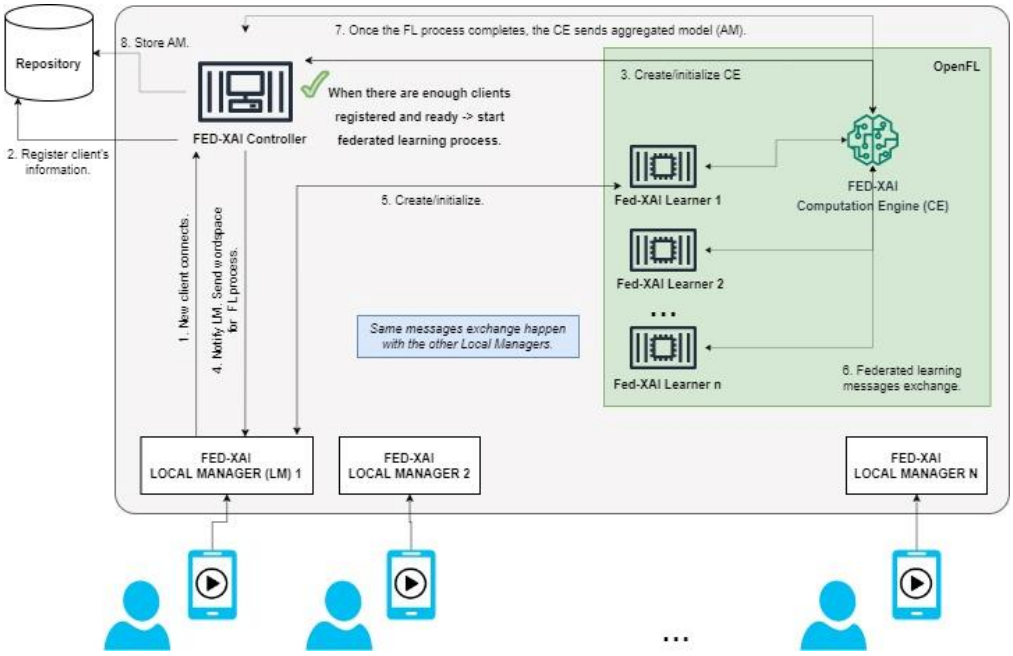


Figure 7-1: Overall workflow of the framework.

Figure 7-1: schematizes the overall workflow of the framework. In a nutshell, once a user applies for a framework service, such as QoS/QoE prediction, an instance of Fed-XAI LocalManager (LM) is created and possibly becomes available to participate in the federation. When enough users are available to participate to the FL process, the Fed-XAI Controller initializes the federation, instantiates the Fed-XAI Computation Engine (CE), generates the OpenFL workspace and shares it with the Fed-XAI LMs participating to FL. Each Fed-XAI LM instantiates a Fed-XAI Learner component, which takes part to the OpenFL learning process as client-side collaborator. The final federated model, aggregated by the FED-XAI CE using the strategy discussed in [CDE+22], is sent to the Fed-XAI Controller to be stored persistently. Once the FL process terminates, all the OpenFL containers can terminate their execution.

A client aiming at making predictions requests a trained model to the FED-XAI controller through its FED-XAI local manager. Then, a container for making inference with the requested model is instantiated. The inference container receives the stream of data from the UE and from the network and leverages the trained model for making predictions.

Figure 7-2 shows the dashboard window during the inference phase, w.r.t. a QoE forecasting case study (further details provided in Section 7.2). The high interpretability of FRBS can be appreciated: the "Prediction" tab shows to the user the value of the prediction (in the example we show the predicted video quality as a percentage) for a pre-fixed horizon window, given a tuple of input statistics calculated on QoS metrics. Due to the maximum matching strategy, the prediction depends on a single rule (the most activated by the input statistics): the "Antecedents" tab shows the linguistic value ("low", "medium" or "high") of each feature in the antecedent part of the rule. For instance, end2endDelay_mean_W represents the average value of the end-to-end-delay metric over the past time window of size W. The "Consequent" tab shows the weights, namely the importance level, of each feature for current prediction, whereas the "History" tab tracks the most recent prediction values.

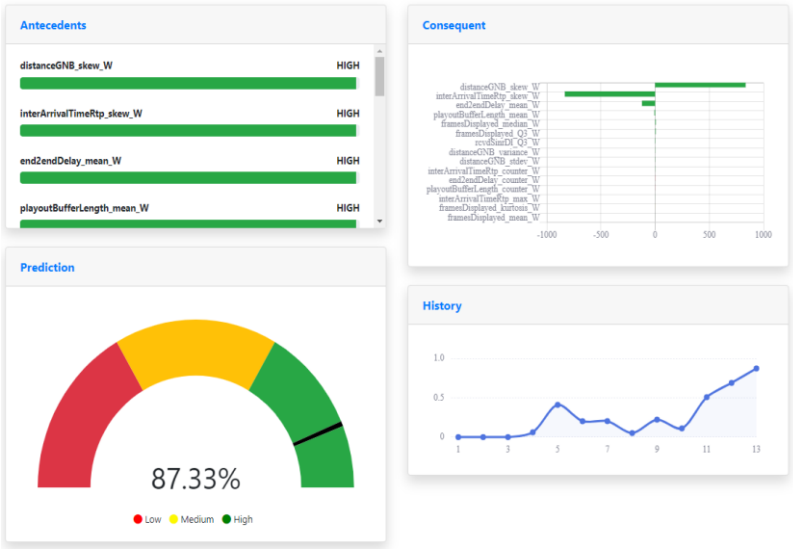


Figure 7-2: Snapshot of the Fed-XAI dashboard during inference.

7.2 Quality of Experience forecasting case study

A realistic dataset for Quality of Experience (QoE) forecasting has been generated through Simu5G, an open-source model library for the OMNeT++ framework. The dataset entails seven gNBs and 15 UEs, each moving around in the floorplan and experiencing a video stream. 24 independent replicas of a 120-second simulation are produced, leading to the generation of time-tagged for each UE. The collected metrics are reported in Table 7-1.

Name	Level	Description
UE position	Application	(x, y, z) coordinates of the UE in the floorplan
UE speed	Application	Speed of the UE in $\frac{m}{s}$
avgServedBlocksDl	Network	Number of Resource Blocks occupied in downlink
averageCqiDl	Network	CQI values reported in DL
rcvdSinrDl	Network	SINR value measured at packet reception
servingCell	Network	ID of the new serving cell after the handover
frameSize	Application	Size of the displayed frame (Byte)
rtpPacketSize	Application	Size of the RTP packet (Byte)
end2EndDelay	Application	Time between transmission and reception of an RTP packet
interArrivalTimeRtp	Application	Interarrival time between two RTP packets
rtpLoss	Application	RTP packets of frame lost
framesDisplayed	Application	Frame percentage arrived at the time of its display
playoutBufferLength	Application	Frame buffer size
firstFrameElapsedTime	Application	3 values: 1) timestamp of the UE request, 2) timestamp of the sender ACK, 3) time between the request and the first frame displayed

Table 7-1: Description of the metrics included in the dataset.

The QoE forecasting problem is formulated as a regression problem, specifically aimed at predicting the value of the *framesDisplayed* metric. Details about dataset generation, feature extraction and preprocessing steps are available in [CDM+22].

The approach for FL of TSK-FRBS (described in Section 5.3.2) is exploited to build the Fed-XAI model. Each UE involved in the simulation acts as a registered client in our FL framework, available for participating to the FL process. To assess the performance of the proposed approach, the FL setting is compared to two alternative learning settings:

- Local Learning (LL): each UE locally learns its own TSK-FRBS. This setting is privacy preserving, but it rules out any form of collaboration among Ues.
- Centralized Learning (CL): the local training sets are merged in a global training set, which is used to train a global TSK-FRBS. This setting violates the privacy of data owners (as local data are gathered to a central location), but represents the utmost form of collaboration, albeit often unviable.

We train our models on a training set composed of 20 out of the 24 replicas of the simulated scenario. The remaining replicas are used as test set (each UE uses a model for making prediction on its own test set portion). The quality of prediction is evaluated in terms of MSE and coefficient of determination (R^2).

Figure 7-3 (a) shows the empirical cumulative distribution function (ECDF) of the difference between the MSE score of FL setting and the MSE score of LL setting, and between the MSE score of FL setting and the MSE score of CL setting for each of the 60 video sessions, i.e., 4 test replicas for each of the 15 Ues). The plot can be interpreted as follows: a curve lies in the negative half-plane if and only if the MSE values of the FL model are lower than those of the model with which it is compared. Figure 7-3 (b) reports analogous evaluation on the R^2 score: in this case a curve lies in the positive half-plane if and only if the R^2 values of the FL model are higher than those of the model with which it is compared. It can thus be noticed that in the 80% of the cases the FL setting obtains better performance compared to the relevant local model, thus demonstrating the benefit of the federation.

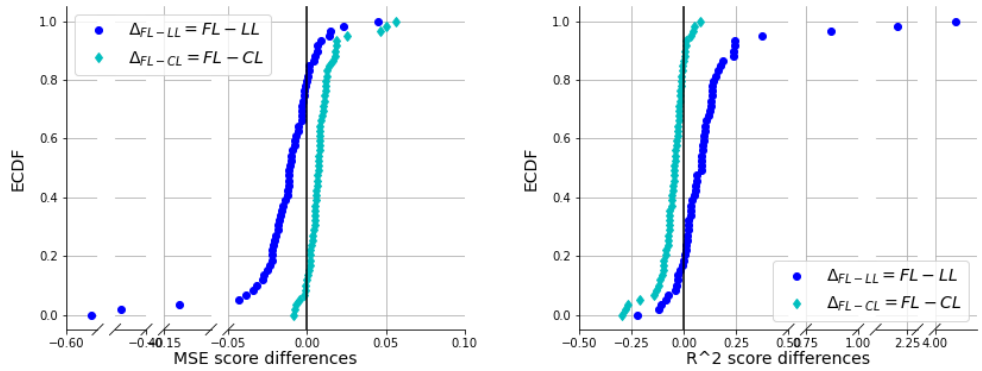


Figure 7-3: Empirical cumulative distribution function (ECDF) of the differences of MSE scores (a) and R^2 scores (b) between FL and LL (dark blue circles) and between FL and CL (light blue diamonds).

Table 7-2 shows the average values of MSE and R^2 obtained with the three learning settings on the test set. Results confirm that the federated TSK model outperforms the locally learned models, on average. Furthermore, under the optimistic assumption that privacy preservation is not a concern, the ability to train the model on the whole training set leads to the best average results.

The interpretable by design TSK model is compared with an opaque model (MLP with 2 hidden layers with 64 neurons each). For the FL setting, we set the number of rounds to 10, the number of local epochs to 5 and the local batch size to 64.

		FL	LL	CL
TSK	**MSE**	0.066	0.094	0.057
	R^2	0.559	0.376	0.617
NN	**MSE**	0.059	0.061	0.056
	R^2	0.606	0.590	0.611

Table 7-2: Average values of MSE scores and R^2 scores on the test set obtained with the three learning settings by the TSK and NN models.

In the centralized setting, the performance obtained by the TSK and by the NN models are comparable: this suggests that, upon large data availability, both XAI and black-box models are suitable for tackling the task. In our specific scenario, the NN achieves higher modelling capability in the local learning (LL) setting and slightly higher also in the FL setting, as evidenced by the MSE values (the lower the better) and R^2 values (the higher the better) reported in Table 7-2. However, the gain in inference accuracy is limited (R^2 from 0.559 to 0.606), i.e., within 10%, and is counterbalanced by a gain in terms of interpretability.

8 Hexa-X quantified targets in Connecting intelligence towards 6G

8.1 AI for BER/BLER improvements (T1)

Target T1 is defined as the increased AI algorithm robustness to system parameter volatility, lower complexity and significant BER/BLER gain, as compared to classical approaches. The goal is to design and implement AI/ML-based solutions which provide BER/BLER gains for the identified use cases,

and KPIs are thus BER/BLER gain, NMSE in channel estimation error, complexity gain, and flexibility. The proposed technical enablers under the AI driven air interface design mainly cater for the "Interacting and cooperative mobile robots", "Merged reality game/work" use cases, and the "AI-assisted Vehicle-to-Everything" service. Interacting and cooperative mobile robots use case would utilise novel system architectures such as D-MIMO, cell-free massive MIMO and RIS-assisted communications which could be used to manage a cluster of drones over a 6G network for improved end-to-end performance. The technical enablers ML-based end-to-end learning of RIS-assisted communication systems and ML-based channel estimation for RIS-assisted systems with mobility in Section 3.1.1 and Section 4.2 specifically target improving the BER and the NMSE in channel estimation in RIS-assisted systems, respectively.

Furthermore, the NN/ML aided channel (de)coding for constrained devices proposed in Section 3.1.2 provides improved end-to-end BER/BLER and complexity gain. The low complexity channel estimation using neural networks presented in Section 4.3 provides improved BER by reducing the NMSE in channel estimation error. Finally, the AI-empowered receiver for PA non-linearity compensation solution and the end-to-end learning solution presented in AI-based enhancements for sub-THz are applicable for all the relevant use cases in general which have shown to provide BER/BLER improvements.

In Table 8-1 the technical enablers under the AI driven air interface design relevant to Target T1 are listed, along with the targeted 6G KPIs/ KVIs for each technical enabler.

Technical Enablers	Quantifiable Targets	Targeted 6G KPIs/KVIs
ML-based end-to-end learning of RIS-assisted communication systems (Section 3.1.1)	Improved end-to-end BER/BLER	BPSK: BER < 0.001, QPSK: BER<0.01 for 16-element RIS and SNR > 15dB (for the full-CSI case)
NN/ML aided channel (de)coding for constrained devices (Section 3.1.2)	Improved end-to-end BER/BLER	BLER ~ CCSDS /w n=128 (3iterations/)
AI-empowered receiver for PA non-linearity compensation (Section 3.1.4)	Improved BER/BLER	BLER gain: ~1 dB gain @ 10% BLER
AI-based enhancements for sub-THz (Section 3.1.5)	Improved BLER	BLER gain: 1-2 dB
Channel charting based beamforming (Section 3.1.6)	Normalized correlation between the precoder and the target channel	n.a.
ML-based channel estimation for RIS-assisted systems with mobility (Section 4.2)	Channel estimation error (providing improved channel estimation error over baseline algorithms, while reducing the resource overhead)	NMSE, Spectral efficiency 5-10 dB reduction in NMSE
Generalizable low complexity channel estimation using neural networks (Section 4.3)	Channel estimation error reduction, improves the throughput by reducing the BER	NMSE Within 3 dB range of MMSE ChanEst performance (the optimal solution) in terms of NMSE

Table 8-1: Summary of technical enablers relevant to Target T1 along with targeted 6G KPIs/KVIs.

8.2 AI for efficient resource utilisation (T2)

Target T2 is defined as the increased AI algorithm robustness to system parameter volatility, lower complexity and efficient resource utilisation and rate gain as compared to classical approaches. The goal is to design and implement AI/ML-based solutions which provide efficient resource utilisation and throughput improvement for the identified use cases and services. The proposed technical enablers under the AI driven air interface design mainly cater for the “Interacting and cooperative mobile robots”, “Merged reality game/work” use cases and the “AI-assisted Vehicle-to-Everything” service. Relevant KPIs applicable for these use case under T2 are throughput, complexity gain (reducing the processing time/number of operations in the relevant problem scenario where other problem-specific metrics are achieved), and efficient resource utilisation as applicable for the relevant problem scenario.

Considering the interacting and cooperative mobile robots where novel system architectures such as D-MIMO and cell-free massive MIMO could be used, the AI based compressed sensing for beam selection in D-MIMO solution in Section 3.1.3 enables rate gain via mobility support and efficient resource utilisation via faster beam scanning time and higher beam selection frequency. Furthermore, the low complexity radio resource allocation in cell-free massive MIMO in Section 4.1 achieves complexity gain via utilising a ML algorithm for RRM. Finally, the AI-empowered receiver for PA non-linearity compensation solution and the end-to-end learning solution presented in AI-based enhancements for sub-THz are applicable for all the relevant use cases in general which can provide throughput gains and resource utilisation as shown in Section 3.1.4 and Section 3.1.5 respectively.

In Table 8-2 the technical enablers under the AI driven air interface design relevant to Target T2 are listed, along with the targeted 6G KPIs/ KVIs for each technical enabler.

Technical Enablers	Quantifiable Targets	Targeted 6G KPIs/KVIs
NN/ML aided channel (de)coding for constrained devices (Section 3.1.2)	Complexity gain	n.a.
AI based compressed sensing for beam selection in D-MIMO (Section 3.1.3)	Efficient resource utilisation, rate gain via mobility support (faster beam scanning time, higher beam scanning frequency)	Beam scanning time reduced to 5-20% of the baseline exhaustive scan (in a deployment with 100-1000 beams)
AI-empowered receiver for PA non-linearity compensation (Section 3.1.4)	Improved throughput and resource utilisation	Throughput gain: ~20%
AI-based enhancements for sub-THz (Section 3.1.5)	Improved throughput and resource utilisation	Throughput gain: 10-20%
Low complexity radio resource allocation in cell-free massive MIMO (Section 4.1)	Complexity gain	Complexity gain: more than x50 times reduced computational complexity and above 95% SE performance compared to the baseline optimization-based algorithm, Flexibility and generalisability: at least 90% SE performance compared to baseline when using the same trained-DNN model for different system configurations.

Table 8-2: Summary of technical enablers relevant to Target T2 along with targeted 6G KPIs/KVIs.

8.3 Resilient communication and compute for large scale distributed AI (T3)

The target is defined as “Resilient communication and compute network services for distributed AI applications in large scales (e.g., applications with >1000 collaborating AI components)”. There are several targeted use case families where the application systems are highly distributed on different devices and network components. For example, the “Interacting and cooperative mobile robots” use case requires real-time intelligent decisions based on distributed and resource efficient data and model sharing. Similarly, in applications of “Hyperconnected resilient network infrastructures” a huge amount of data will be distributed on thousands or millions of devices, all of which is not feasible to be shared due to communication, capacity, privacy, complexity, and other reasons. These AI-enabled components will jointly realize a heterogenous AI compute and data sharing landscape where stored data, real-time sensor input, processing and control capabilities are distributed.

Use cases realizing real-time and critical functionalities in such distributed environment would pose stringent requirements on the communication network in terms of packet latency, loss rates, and bandwidth stability, along with a high signalling overhead to manage it over wireless in a transparent manner. However, the key performance indicators relevant for the AI-enabled applications can also be realized by joint design of application-level control, network services (e.g., AIaaS), and supporting communication functions, which leads to more relaxed network requirements but still ensuring resilient application behaviour. The KPIs considered relevant for highly distributed resilient AI applications are:

- AI agent density
- AI inference accuracy
- AI inference latency
- AI agent availability and reliability

This quantified target T3 and its related KPIs are addressed by several technical enablers. Scalable and resilient deployment of distributed AI proposed joint compute and communication system with advantages over the traditional approaches. By relying on incremental inferencing frameworks, it is possible to do both early phase ultra-low latency inference, as well as higher accuracy at the cost of higher delay. In a distributed sensor sharing application, where multiple inputs provide overlapping information for the inference task, over-the-air communication can be significantly reduced with the help of application control interacting with network layers on a fast and high data granularity level. In some cases, it can require per packet control on millisecond timescale. The benefits, however, are reduced traffic and connection load by 80% in the investigated scenario. The device density requirements for the targeted demanding 6G use cases are up to 5-10 devices/m^2, which is 5-10 times higher than 5G requirements. The method for resilient deployment of distributed AI with joint fine-grained control from the application layer and millisecond level traffic control by the networking layers has a significant contribution to reach these numbers in a distributed sensing and communication scenario with ~1000 collaborating AI components, as targeted in T3.

The concept of AIaaS is primarily targeting AI agent availability and reliability in high-mobility environments involving safety-critical communications. The proposed solution is expected to enhance system robustness to mobility events, including (i) increased availability by improved mobility solutions, (ii) an increased reliability by accounting for low-quality connections, (iii) mitigating latency due to radio handovers, association to different AI agents in mobility events. AI models can rely on a substantial information pool, possibly the entire knowledge of the network, which can be exploited through providing AI models to the UE without exposing the actual underlying information.

The high volume of data sharing requirement in large scale distributed AI applications is addressed by the CTDE framework, which enables centralized learning with decentralized execution and provides a practical and realistic approach for real-world cellular environments that require varying levels of observability and time constraints. In the investigated use case of multi-cell multi-user MIMO problem, the framework provides means for scalability to accommodate the increasing number of users and cells,

enables high AI agent density by reducing the volume of shared data by using partial observability during inference, while ensuring real-time fulfilment of beamforming schemes.

Large scale deployment of AI workloads on multiple physical nodes can have a significant impact on E2E latency, which eventually results in increased AI training or inferencing latency. The provided algorithm for AI workload placement reduced E2E latency by up to 98% compared to the baseline in the investigated scenario.

Technical enablers	KPIs addressed	Results
Scalable and resilient deployment of distributed AI (Section 5.2.1)	Inferencing latency Inferencing accuracy Device density	Improvements in sensor sharing inference applications with 100-1000 collaborating AI components simulated: ▪ inference latency reduced by 2-5x for early lower accuracy results ▪ incremental inference enabled with gradual tradeoff in accuracy ▪ device density increased by 5x due to reduction of traffic load and active connections
AI workload placement for energy, knowledge sharing and trust optimization (Section 5.1.3)	E2E latency	▪ Up to 23% reduction of power consumption compared to baseline (random feasible placement) ▪ Up to 98% reduction of E2E latency depending on number of AI workloads, compared to baseline (random feasible placement)
Distributed AI for automated UPF scaling in low-latency network slices (Section 3.2.1)	Inferencing latency Training latency Inferencing accuracy	▪ Inferencing latency – less than the half the timestep of the input data (< 30 seconds); ▪ Training latency – time from AI M&O request of training to the instantiation of the ML training pipeline < 1 minute; ▪ Inferencing accuracy – LSTM 89% accuracy on training data, 83% accuracy on runtime data
Centralized training and decentralized execution (CTDE) approach to multi-cell multi-user MIMO (Section 5.3.1)	Inferencing latency Inferencing accuracy AI agent density	▪ Inference latency reduced as model only considers local information. ▪ Signalling reduced as model only considers local information, contributing to increased AI agent density. ▪ Inference accuracy improved compared to a model that is trained only on local information.
Goal-oriented approach for edge inference (ensemble edge inference to increase computing reliability) (Section 5.2.3)	Goal-effectiveness (Inference latency + inference accuracy) Energy consumption	Energy consumption reduction at UE side thanks to goal-oriented approach in edge inference offloading (simulation-based assessment) ▪ In simulated network conditions, 12% energy reduction with no loss is goal-effectiveness (defined as correct inference on time) ▪ Higher gains obtained by trading off goal-effectiveness, due to inference accuracy reduction

Table 8-3: Technical enablers and KPIs addressing target T3 Resilient communication and compute network services for distributed AI applications in large scales.

8.4 XAI model accuracy (T4)

The design of AI systems, which will play pivotal role in future B5G/6G networks, cannot be simply targeted to optimize accuracy and technical robustness but must also comply with additional requirements towards trustworthy AI, such as transparency of AI models, regardless of the specific use case. In this regard, the Connecting Intelligence objective entails a specific quantifiable target, discussed in D4.2, Sec. 7, seeking "*the accuracy of an XAI model within (<10%) of "black box" solutions*".

The KPI to be addressed is indeed inference accuracy, to be pursued together with a high level of interpretability of ML models. The following table summarizes the technical enablers contributing to this target.

Technical Enablers	**KPIs**	**Results**
XAI models: Fuzzy regression trees and TSK Fuzzy Rule Based Systems (Section 6.3.1)	Inference accuracy and explainability	Trade-off between model interpretability and inference accuracy on benchmark datasets: - inference accuracy (measured as MSE on regression task) comparable (or within 10%) to less interpretable models and state-of-art solution; In the case of TSK, the gain of XAI model vs less interpretable state-of-art solution is in the range [-4%, +37%] based on results shown in Table 6-2 Comparison of our approach (TSK-MM = maximum matching, TSK-WA = Weighted Average) and a state of art approach [FSN+20] for building TSK-FRBSs, in terms of MSE. - Higher inference accuracy obtained with first-order polynomial models compared to lower complexity zero-order ones.
Fed-XAI: Federated Learning of Explainable AI models (Section 6.3.2)	Inference accuracy	Inference accuracy of XAI models learned in federated fashion higher than those locally learned - evaluation on benchmark dataset in terms of MSE with gain in the interval [1x,2x]

Table 8-4: Technical enablers and KPIs addressing target T4 XAI model accuracy.

In Section 6.3.1 the accuracy and interpretability of XAI models for regression tasks (Fuzzy Regression Trees and Takagi-Sugeno-Kang Fuzzy Rule-based Systems) have been investigated, with respect to design choices related to the inference strategy and the order of the polynomial regression model. The proposed TSK FRBS with enhanced interpretability is shown to be comparable to (or even better than) a less interpretable state of art TSK-FRBS, based on an empirical comparison on several benchmark datasets. The results confirm the outcome observed in the context of classification analysis, (D4.2, Section 5.2.1), for which inherently interpretable models (decision trees) are slightly outperformed (within 10%, in terms of f-score) by a Random Forest classifier which, as an ensemble model, is generally considered opaque.

The target is also discussed in the context of the Fed-XAI demonstration activity. The Fed-XAI technical enabler (Section 6.3.2) is first evaluated on several benchmark datasets: the proposed approach for collaboratively training inherently interpretable models (TSK-FRBS) is shown to be

suitable, showing a significant improvement over local training. Furthermore, in the considered QoE forecasting problem (Chapter 7), a black-box model (Neural Network) achieves slightly higher modelling capability in the FL setting compared to the highly interpretable TSK-FRBS.

8.5 Energy reduction at the infrastructure and user devices (T5)

The energy consumption target is defined as follow: 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:

Various technical enablers presented in D4.2 and D4.3 address the problem of energy consumption at devices and infrastructure side (separately or jointly), including communication and computing. For these services, energy consumption is traded off with delay and/or accuracy of learning and inference workloads running at the edge. Contributions in D4.3, especially in Chapter 5 include several aspects of workload management and resource orchestration. The scalable and resilient deployment of distributed AI functions is proposed Section 5.2.1, addressing KPIs including inference latency and device density, but also decreased device communication energy by down-prioritizing sources with high communication cost and low utility for application KPI (goal-oriented concept) at a packet-level timescale. Also, going further in workload management, deep neural network splitting criteria and resource allocation to enable low energy edge inference with latency guarantees (Section 5.2.2). Furthermore, the goal-oriented communication paradigm is introduced (Section 5.2.3), to go beyond the legacy concept of reliably transmitting bits, to rather focus on the application performance, in this case the correct (or confident) inference on time, trading off energy consumption and other legacy communication related KPIs. Finally, workload placement for energy, knowledge sharing, and trust optimisation is proposed (Section 5.1.3) to reduce the power consumption, trading off delay, i.e., comparing the proposed method to a random feasible placement approach. End-to-end latency and power consumption are key performance indicators in this direction.

Overall, simulation-based evaluations showed that latency can be reduced by 2-5 times for early lower accuracy results, energy reduction at devices side under high availability of computing resources, in the considered connect-compute network scenario and parameters (available edge resources, path loss, etc.). Different gains (e.g., up to 90% at device side) are achieved, depending on such conditions, thanks to DNN optimal splitting point selection. Also at device level, further gains can be obtained through the goal-oriented approach, by relaxing communication performance while not sacrificing the application. At the network side, optimised workload placement has shown to reduce the power consumption by 23% compared to baseline strategies. At MEC network side, thanks to the cooperative inference setting in Section 5.2.3, the statistical availability of each MEH can be reduced without degrading performance, provided that they cooperate to avoid single point of failure issues. This, in case of theoretical cubic models for CPU power consumption allows the MEC network to reduce the energy consumption. Depending on network conditions and performance requirements, in the simulated setting, 2x to 10x gains are obtained at the MEC network. All these solutions are foreseen to contribute to the targets on energy consumption at device and infrastructure side.

Technical enablers	KPIs addressed	Results
Scalable and resilient deployment of distributed AI (Section 5.2.1)	Inferencing latency Inferencing accuracy Device density	Improvements in sensor sharing inference applications: ▪ inference latency reduced by 2-5x for early lower accuracy results ▪ incremental inference enabled with gradual tradeoff in accuracy ▪ device density increased by 5x due to reduction of traffic load and active connections
DNN splitting and resource allocation for edge inference (Section 5.2.2)	Inference latency Energy consumption	Energy consumption reduction at UE side thanks to partial offloading of inference workloads through DNN splitting (simulation based assessment of splitting point selection algorithm) ▪ Inference latency as a requirement, i.e., not to exceed a predefined threshold ▪ Depending on channel models (e.g., path loss exponent) up to 90% reduction in case of highly available edge computing resources. Gains reduced in case of volatile edge computing resources. Scenario in which full offloading is not feasible under simulated network conditions ▪ Empirical model of computing power consumption but theoretical model for UE transmit power consumption
Goal-oriented approach for edge inference (Section 5.2.3)	Goal-effectiveness (Inference latency + inference accuracy) Energy consumption	Energy consumption reduction at UE side thanks to goal-oriented approach in edge inference offloading (simulation-based performance assessment) ▪ In simulated network conditions, 12% energy reduction with no loss is goal-effectiveness (defined as correct inference on time) ▪ Higher gains obtained by trading off goal-effectiveness, due to inference accuracy reduction ▪ At the MEC network side, 2x to 10x energy consumption reduction obtained in simulated network setting thanks to cooperative inference
AI workload placement for energy, knowledge sharing and trust optimisation (Section 5.1.3)	E2E latency power consumption	▪ Up to 23% reduction of power consumption compared to baseline (random feasible placement) ▪ Up to 98% reduction of E2E latency depending on number of AI workloads, compared to baseline (random feasible placement)

Table 8-5: Technical enablers and KPIs addressing target T5 Energy reduction at the infrastructure and user devices.

8.6 Increased trustworthiness of AI (T6)

The target T6 is defined as increased trustworthiness of AI through privacy and security enhancing technologies. The goal is to ensure that AI systems are developed in a way that minimizes potential privacy and security risks, while also maintaining the performance and functionality of the AI system. We proposed two technical enablers to enhance the privacy and security of the AI systems. To evaluate the effectiveness of the AI system, KVIs and KPIs are defined. In Federated Learning (FL), although each participant trains on its own data and only shares the model updates with other participants, the local model updates can still disclose information about training data of clients. It is hard to measure how much the privacy of the FL system is improved when using privacy enhancing methods, however we can use the model accuracy and total overhead as KPIs and compare the values for these KPIs before and after applying privacy enhancing methods to the FL. In case of security, to evaluate the effectiveness of the adversarial attacks on AI system and applicability of mitigation technique, attack success rate, robustness of the AI model to withstand adversarial attacks (i.e., evaluating the effectiveness of defence technique), and efficiency (i.e., the computational cost of the defence technique) can be considered as KPIs.

Technical enablers	KPIs addressed	Results
Security friendly Privacy solution for federated learning (Section 6.2.1)	Model Accuracy Total overhead	▪ Accuracy of model remains same (100%) ▪ Up to 10.76% increase in total overhead in comparison with typical FL when communication speed is 504 Mb/sec.
Defence mechanism to increase robustness of AI-driven power allocation against adversarial attacks (Section 6.1.2)	Attack success rate Robustness of AI model	▪ Up to 37.29% reduction of Attack success rate ▪ Up to 37.29% increase in robustness of the Al model against adversarial attacks

Table 8-6: Technical enablers and KPIs addressing target T6 Increased trustworthiness of AI

Conclusions

AI-driven communication & computation co-design will constitute a major leap forward by 6G systems over previous generations. Instead of addressing AI and computation tasks on higher layers only, corresponding enablers will be deeply anchored in future 6G system and will rely on AI- and compute-native design approaches.

In the present deliverable, corresponding key enablers have been identified and studied – further adding details, substantive insight and evaluation results over the previous Hexa-X Deliverables [HEX-D41] and [HEX-D42]. Finally, all new approaches have been evaluated again Hexa-X Key Performance Indicators (KPIs) and Key Value Indicators (KVIs) as they have been defined on a project wide level.

Sustainable 6G is one of the key objectives of the Hexa-X project and AI- as well as computation based solutions are clearly expected to provide support. A novel *Low complexity radio resource allocation in cell-free massive MIMO* is introduced; specifically, existing unsupervised learning concepts are extended for the pilot and data power control problem in uplink of a cell-free massive MIMO network. It is shown that the proposed power control improves the system sum rate. Furthermore, an *ML-based*

channel estimation for RIS-assisted systems with mobility is introduced relying on reconfigurable intelligent surfaces (RISs) to control the wireless propagation environment with software-controlled reflections. A novel ML based channel estimation is proposed and evaluated specifically tailored to RIS aided systems, while considering mobility. This study is complemented by a proposal on a *Generalizable Low Complexity Channel Estimation using Neural Networks*, which is inspired by the MMSE estimator for the linear problem of noisy pilot observations and introduces a Neural Network which resembles such an MMSE estimator. Under a proposed approximation, it is possible to treat the problem as three smaller problems and further even decompose the spatial domain as vertical and horizontal domain. This allows for smaller NNs that can be even trained separately, which roughly translates to smaller training sample complexity. A further approach relies on *Deep unfolding for efficient channel estimation*, extending a deep unfolding approach for channel estimation as explored in [HEX-D42] in the context of MIMO channels with a single subcarrier; it is demonstrated how to use the same technique for SISO-OFDM channels with multiple subcarriers. The result is a novel frugal channel estimation algorithm with reduced computational complexity when compared to generic deep learning- based approaches. Finally, a novel *hybrid model for channel charting* is considered. Channel charting is an unsupervised learning method which falls within the realm of machine learning in general, and dimensionality reduction in particular. It aims at projecting high-dimensional channel observations into a low-dimensional space, typically of 2 or 3 dimensions, in order to learn a channel chart. In fact, physical channel models indicate that channel observations are subject to the manifold hypothesis, meaning that although their original space is of high dimension, they are, in reality, governed by a small set of parameters.

Security, privacy, and trust in AI-enabled 6G related solutions are essential building blocks of communications systems. In this context, *Adversarial attacks and mitigation techniques in AI-driven power allocation* are studied in detail. In a D-MIMO network context, we consider two sources of adversarial attack scenarios, the first related to malicious RUs or fronthaul links, the second related to malicious UE, considering the success of the adversary under both white-box and black-box setting. The results show that adversarial attacks are more effective than standard Gaussian noise. We introduce a proactive defence mechanism in which we look for an imaginary perturbation, that by maximizing the user spectral efficiency produces a better power allocation outcome. A further focus is on *security mechanism friendly privacy solutions for federated learning*. Although FL enhances the privacy of clients by enabling each client to train its own local data, there is still a possibility of information leakage when clients send local model updates to server to aggregate to a global model. In order to detect malicious forwardee clients who drop packets, we introduce and utilize a detector entity to the network which monitors packet exchanging procedure between a sender and a forwardee to find whether the process is fully accomplished or not. To further enhance inferencing accuracy and explainability, we consider *Fed-XAI: Federated Learning of Explainable AI models* as well as *XAI models: Fuzzy regression trees and TSK fuzzy rule based systems*. We exploit the fact that Rule Based Systems (RBSs) and Decision Trees (DTs) are generally considered more inherently interpretable than other popular "opaque" models (e.g., Neural Networks and Random Forest) and still able to achieve competitive performance, especially when data comes in tabular form. The trade-off between accuracy and interpretability of the XAI models are analysed.

Building on sustainability and security objectives, we further study solutions to enable the future **6G network as an efficient AI platform**. First, we consider a disruptive new communications paradigm in the context of *Goal-oriented communication approach for edge inference*. While the classical (data-oriented) communication paradigm is to reliably transmit bits (i.e., it focuses on bit-quality oriented metrics), goal-oriented communications measure the actual performance of communication by its effectiveness at higher layers, e.g., by measuring the inference reliability (accuracy, confidence, etc.). Through numerical simulations, it is shown how the effectiveness can achieve good performance also in the presence of relaxed BER requirements, suggesting that high communication reliability is not necessarily needed, as far as inference performance reaches target levels. Furthermore, we acknowledge that future 6G networks will be designed such that AI capabilities are natively integrated. This leads to the principles of *AI-as-a-Service (AIaaS) - seamless exploitation of network knowledge*. Building on an initial solution proposal [HEX-D42] and a proposed protocol framework for AI-as-a-Service (AIaaS)

[HEX-D51], we provide further details and propose a set of data structures to support a proposed method, applicable to scenarios calling for frequent inferencing-based decisions. This approach is further complemented by a study on *Network impairment resilience of autonomous agents*. The objective is to increase resilience towards connection problems on the AI agent side, such as abnormal bearer session release (i.e., bearer session drop) in cellular telecommunication networks which seriously impact the QoS of mobile users. Specifically, we target the Massive twinning and Robots to Cobots Hexa-X Use cases, both in which real-time intelligent decisions have to be made based on distributed data available partly locally and partly from the network. The management of distributed learning data is furthermore addressed in the proposal of a *Federated ML model load balancing at the edge*, where large number of heterogeneous sensors are connected to FL nodes at the edge and the goal is to provide a low latency and high quality distributed ML service. We propose new approaches to 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. Similarly, our investigation on *scalable and resilient deployment of distributed AI* focuses on distributed sensing and communication scenarios where the applications benefit from tight integration of sensors and communications. Our solutions reduce both the overall inference latency and communication load by relying on an incremental evaluation framework. Incremental inference can be used in several AI models. Furthermore, we address *Joint communication and computation orchestration for edge inference*, acknowledging that learning and inferencing at the edge require to collect, pre-process (e.g., extract features), transmit, and process data (or features) remotely in a continuous fashion. The proposed solution aims at jointly optimising the splitting point selection and device transmit power, under time-varying wireless channel conditions and Mobile Edge Host (MEH) availability. The goal is to minimize device energy consumption (including transmission and computation) under end-to-end service delay constraints of the edge inference service (both in average and probabilistic sense), which involves: i) local computation delay, ii) wireless uplink delay, and iii) remote computation delay. Further efficiency improvements are expected through *Frugal Federated Learning*. We propose a solution framework for customizing FL training to each contributing network entity's (e.g., device, edge cloud server) available compute, communication and energy resources. Also, we minimise bi-directional communication signalling during FL training. From those holistic and generic studies, we move to the specific case of *centralized training and decentralized execution (CTDE) approach to multi-cell multi-user MIMO*. To address the multi-cell multi-user MIMO precoding problem, we use a MA-DDPG algorithm to train decentralized actors and share critic in multi-cell multi-user MIMO systems under the assumptions of local CSI and no inter-cell data sharing. All of these approaches obviously require appropriate compute resources, likely made available through a distributed Compute-as-a-Service (CaaS) approach [HEX-D52] for which we study the *Flexible compute workload assignment, CaaS*. Among the available code types Source Code, ConfigCode and Executable Code, we now consider the optimum code type selection for the Hexa-X Use Case Families and individual use cases thereof. The related choice is integrated into the Radio Application Package format for which corresponding extensions are being proposed. Finally, a study is conducted on *AI workload placement for energy, knowledge sharing and trust optimization*. We introduce a close to optimal placement of AI workloads to the various network's physical nodes by minimising the energy consumption of the overall network towards sustainability, minimising traffic, and maximising the trust level. The AI workloads are assumed to be mostly inferencing related.

As a next step, **Intelligent E2E Management and Orchestration** are considered. One challenge addressed by the inclusion of AI in network management and orchestration is to avoid degradations in service caused by finite User Plane Function (UPF) resources while ensuring security, privacy and reducing data flow demands. This is being addressed through a study on *distributed AI for automated UPF scaling in low-latency network slices*. The agent is responsible for inferring traffic patterns associated with local UPFs and using this information to foresee opportunities for optimising resource allocations. Further improvements are achieved through *AI/ML based predictive orchestration*, which offers a wide range of approaches in order to ease the M&O-related operations within the network, being Predictive Orchestration is one of those approaches. This method aims at foreseeing future states of the network, using prediction/forecast AI/ML-techniques mainly based on time-series forecasting, to

be able to maximize the output of M&O operations in terms of flexibility, dynamicity, real-time, or resource placement, etc. among others. As a result, the forecasting algorithms have been split between the Management Functions block and the AI/ML Functions block at the Network Layer.

Finally, we consider the lower layers of the 7-Layers OSI model proposing **AI/ML solutions for RAN performance enhancements and sustainable 6G**. We start with a study on *ML-based end to end learning of RIS-assisted communication systems*. The numerical results show the potential of a CNN-autoencoder (CNN-AE) system for an RIS-assisted communication system which is capable of learning transmit signals and the optimal reflection coefficients for the RIS in an end-to-end manner to minimize the BER. Also, with the objective to further decrease channel decoding complexity, we study *NN/ML aided channel (de)coding for constrained devices* and investigate the use of Linear Block Codes (LBC) for short block lengths, typically in the range of few tens up to hundreds of bits, using Belief Propagation (BP) decoders, as used for LDPC codes. The goal is to provide a unified decoder architecture that would encompass both extrema in block regimes, and therefore suit both data and control planes requirements. This activity is complemented by a study on *AI based compressed sensing for beam selection in D-MIMO*, which present reduced overhead with certain random dictionaries, local environment statistics can also be utilized to further optimize the dictionary. The distribution of beam patterns received by UEs is not uniform and statistics on typical joint beam patterns can be exploited. Initial investigations indicated that the number of required measurements can indeed be further reduced if we apply AI/ML techniques to learn the dictionary for a given deployment [HEX-D42]. This dictionary training is performed with the help of an autoencoder architecture, where input is a representative set of beam channel vectors sampled from potential UE locations. To further improve transmitter efficiency, a study is being conducted on an *AI empowered receiver for PA non-linearity* approach to compensate the impact of PA non-linearity at the receiver side using a neural network-based demapper in operating regimes in which out of band emission requirements are fulfilled. The simulation results confirm that the proposed method can increase throughput (20% improvement) and/or extend the coverage of a communication link in the presence of PA non-linearity, and to enable enhancing energy efficiency (70% improvement of PA power-added efficiency) at the transmitter side. Another key trend for 6G systems lies in the usage of higher frequencies which is addressed in a study on *AI-Based Enhancements for Sub-THz*. Our results indicate that joint learning of sub-THz transmitter and receiver processing can lead to higher spectral efficiency and high resilience against hardware impairments. This extends our original findings reported in [HEX-D42], where similar results were obtained for a sub-6-GHz center frequency. In the transmitter, the constellation shape is being learned to facilitate pilotless detection, which greatly reduces transmission overhead. The BLER gain is approximately 2 dB, while the throughput improvement is in the order of 20-30%. New approaches to beamforming are addressed in the context of *Channel charting based beamforming*. Channel charting (CC) is the task of locating users relative to each other in an unsupervised way and can be viewed as a way to discover a low-dimensional latent space charting the channel manifold. The results indicate that, although lacking positional information about the UEs, the presented approach is capable of producing precoders of good quality.

In addition to the studies summarized above, a **demonstration setup** has been created including several instances of vehicular User Equipment (UE), connected to a B5G/6G network, receive (or send) a video stream, whose perceived quality is crucial for the availability of advanced driving assistance systems, such as see-through (or tele-operated driving). A Fed-XAI model is used to forecast the perceived video quality based on the available contextual, QoS and QoE metrics; furthermore, the Fed-XAI model is learned in a collaborative fashion and is highly interpretable by design.

Finally, we should understand that the solutions proposed and discussed in the present document come with a certain level of cost – the underlying compute and AI entities will require energy to operate. It is thus proposed that future 6G system design choices rely on a detailed benefit/cost analysis. It is only appropriate to introduce a corresponding new mechanism if the benefits in terms of system efficiency, security and other outweighs the cost in terms of energy consumption or similar.

References

[103 850] ETSI TS 103 850, "Reconfigurable Radio Systems (RRS); Definition of Radio Application Package", V1.1.1, October 2022. Online: https://www.etsi.org/deliver/etsi_ts/103800_103899/103850/01.01.01_60/ts_103850v010101p.pdf

[23.288] 3GPP TS 23.288, "Architecture enhancements for 5G System (5GS) to support network data analytics services (Release 16)", September 2022.

[28.533] 3GPP TS 28.533, "Management and Orchestration; Architecture Framework (Release 17)", March 2022.

[36.888] 3GPP TR 36.888, "Study on provision of low-cost Machine-Type Communications (MTC) User Equipments (UEs) based on LTE (Release 12)", June 2013.

[3GPP16] 3GPP, "TSG RAN1 86 and 87 Meetings - Finale Minutes Reports," 2016.

[5GAA20] 5GAA TR S-200137. Working Group Standards and Spectrum Study of Spectrum Needs for Safety Related Intelligent Transportation Systems—Day 1 and Advanced Use Cases, June 2020.

[AAR22] Alharbe, N.; Aljohani, A.; Rakrouki, M.A. A Fuzzy Grouping Genetic Algorithm for Solving a Real-World Virtual Machine Placement Problem in a Healthcare-Cloud. *Algorithms* **2022**, *15*, 128. https://doi.org/10.3390/a15040128

[AGE21] A. Aberdam, A. Golts and M. Elad, "Ada-LISTA: Learned Solvers Adaptive to Varying Models," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2021.3125041.

[AIA] Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCEACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS, April 2021, available at https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206

[ALK19] A. Alkhateeb, "DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications," in Proc. of Information Theory and Applications Workshop (ITA), San Diego, CA, Feb. 2019, pp. 1–8.

[AO+00] Abe, M. and Okamoto, T., 2000, August. Provably secure partially blind signatures. In Annual International Cryptology Conference (pp. 271-286). Springer, Berlin, Heidelberg.

[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.

[BCD+22] A. Bechini, J. L. Corcuera Bárcena, P. Ducange, F. Marcelloni and A. Renda, "Increasing Accuracy and Explainability in Fuzzy Regression Trees: An Experimental Analysis," 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2022, pp. 1-8, doi: 10.1109/FUZZ-IEEE55066.2022.9882604.

[BCDH+22] T. Borsos, M. Condoluci, M. Daoutis, P. Hága and A. Veres, "Resilience Analysis of Distributed Wireless Spiking Neural Networks," 2022 IEEE Wireless Communications and Networking Conference (WCNC), 2022, pp. 2375-2380, doi: 10.1109/WCNC51071.2022.9771543.

[BDD+19] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M. -S. Alouini and R. Zhang, "Wireless Communications Through Reconfigurable Intelligent Surfaces," in *IEEE Access*, vol. 7, pp. 116753-116773, 2019, doi: 10.1109/ACCESS.2019.2935192.

[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.

[BLSS+20] L. Barriga, M. Liljenstam, K. Seonghyun, and J. Sternby, https://www.ericsson.com/en/blog/2020/10/ai-security-mobile-networks, OCT 08, 2020.

[BNHGT+21] Bahramali, A., Nasr, M., Houmansadr, A., Goeckel, D., & Towsley, D. (2021, November). Robust adversarial attacks against DNN-based wireless communication systems. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (pp. 126-140).

[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.

[CDE05] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu, "Reduced Complexity Decoding of LDPC Codes," IEEE Transactions on Communications, vol. 53, no. 8, pp. 1288–1299, 2005.

[CDE+22] J. L. Corcuera Bárcena, P. Ducange, A. Ercolani, F. Marcelloni and A. Renda, "An Approach to Federated Learning of Explainable Fuzzy Regression Models," 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2022, pp. 1-8, doi: 10.1109/FUZZ-IEEE55066.2022.9882881.

[CDM+22] J. L. Corcuera Bárcena et al. Towards Trustworthy AI for QoE prediction in B5G/6G Networks, in: 1st Int'l Workshop on AI in beyond 5G and 6G Wireless Networks - AI6G2022, Vol.3189, 2022, pp. 1–9.

[CLR22] B. Chatelier, L. Le Magoarou, and G. Redieteab, "Efficient Deep Unfolding for SISO-OFDM Channel Estimation," Oct. 2022.

[CLW+18] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds," in NeurIPS 2018, pp. 9061–9071.

[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.

[CSD+17] J. W. Choi, B. Shim, Y. Ding, B. Rao and D. I. Kim, "Compressed Sensing for Wireless Communications: Useful Tips and Tricks," in IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1527-1550, thirdquarter 2017, doi: 10.1109/COMST.2017.2664421.

[DDF+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[DMR+20] D. Bega, M. Gramaglia, R. Perez, M. Fiore, A. Banchs, "AI-based Autonomous Control, Management, and Orchestration in 5G: From Standards to Algorithms". IEEE Network, vol. 34, no 6, p. 14-20, 2020.

[DMR+21] D. L. Dampahalage, K. B. S. Manosha, N. Rajatheva, and M. Latva-Aho, "Weighted-Sum-Rate Maximization for an Reconfigurable Intelligent Surface Aided Vehicular Network," IEEE Open Journal of the Communications Society, vol. 2, pp. 687–703, 2021.

[ETSI.34] ETSI Whitepaper No.#34: "Artificial Intelligence and future directions for ETSI". June 2020. [Online] Available at: Artificial Intelligence and future directions for ETSI [Accessed 12 of December 2022].

[F+16] Fang, J., Zhang, R., Fu, T. Z., Zhang, Z., Zhou, A., and Zhu, J. Parallel stream processing against workload skewness and variance. arXiv:1610.05121 (2016).

[FCD+18] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. Ten Brink, "OFDM-autoencoder for end-to-end learning of communications systems," in 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2018, pp. 1–5.

[FHS23] H. Farhadi, J. Haraldson, and M. Sundberg, "A deep learning receiver for non-linear transmitter," IEEE ACCESS, doi: 10.1109/ACCESS.2023.3234501

[FMI99] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation," IEEE Transactions on Communications, vol. 47, pp. 673– 680, 1999.

[FS20] H. Farhadi and M. Sundberg, "Machine learning empowered context-aware receiver for high-band transmission," 2020 IEEE Globecom Workshops (GC Wkshps, Taipei, Taiwan, 2020, pp. 1-6, doi: 10.1109/GCWkshps50303.2020.9367518.

[FSE+22] Foley, Patrick, et al. "OpenFL: the open federated learning library." Physics in Medicine & Biology 67.21 (2022): 214001.

[FSN+20] C. Fuchs, S. Spolaor, M. S. Nobile and U. Kaymak, "pyFUME: a Python Package for Fuzzy Model Estimation," 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020, pp. 1-8, doi: 10.1109/FUZZ48607.2020.9177565.

[FU98] G. D. Forney and G. Ungerboeck, "Modulation and coding for linear Gaussian channels", IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2384-2415, October 1998.

[G14] Gedik, B. Partitioning functions for stateful data parallelism in stream processing. The VLDB Journal 23, 4 (2014), 517–539.

[GL10] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in ICML 2010, pp. 399–406.

[GSR+21] N. Ginige, K. B. Shashika Manosha, N. Rajatheva, and M. Latva-aho, "Untrained DNN for channel estimation of RIS-assisted multi-user OFDM system with hardware impairments," in 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2021, pp. 561–566.

[GZC+20] J. Gao, C. Zhong, X. Chen, H. Lin, and Z. Zhang, "Unsupervised learning for passive beamforming," IEEE Communications Letters, vol. 24, no. 5, pp. 1052–1056, 2020.

[HEX-D12] Hexa-X, "Deliverable D1.2: Expanded 6G vision, use cases and societal values – including aspects of sustainability, security and spectrum", April 2021.

[HEX-D13] Hexa-X Deliverable D1.3, "Targets and requirements for 6G - initial E2E architecture". Online: https://hexa-x.eu/wp-content/uploads/2022/03/Hexa-X_D1.3.pdf

[HEX-D41] Hexa-X Deliverable D4.1, "AI-driven communication & computation co-design: Gap analysis and blueprint". Online: https://hexa-x.eu/wp-content/uploads/2021/09/Hexa-X-D4.1_v1.0.pdf

[HEX-D42] Hexa-X Deliverable D4.2, "AI-driven communication & computation co-design: initial solutions". Online: https://hexa-x.eu/wp-content/uploads/2022/07/Hexa-X_D4.2_v1.0.pdf

[HEX-D23] Hexa-X Deliverable D2.3, "Radio models and enabling techniques towards ultra-high data rate links and capacity in 6G". Online: <Link will be available in April 2023>

[HEX-D51] 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

[HEX-D62] Hexa-X Deliverable D6.2: Design of service management and orchestration functionalities. April 2022.

[HKH21] M. Honkala, D. Korpi, and J. M. J. Huttunen, “DeepRx: Fully convolutional deep learning receiver,” IEEE Transactions on Wireless Communications, vol. 20, no. 6, pp. 3925–3940, Jun. 2021.

[HWD+20] He W, Wu Y J, Deng L, et al., Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences. Neural Networks, 2020, 132: 108-120.

[KGB+17] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” ICLR Workshop, 2017. [Online]. Available: https://arxiv.org/abs/1607.02533

[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

[KKÇ+22] Karakoç, F., Karaçay, L., Çomak, P., Gülen, U., Fuladi, R. and Soykan, E.U., 2022. A Security-Friendly Privacy Solution for Federated Learning.

[KKH+19] S. Khan, K. S. Khan, N. Haider, and S. Y. Shin, “Deep-learning-aided detection for reconfigurable intelligent surfaces,” arXiv preprint arXiv:1910.09136, 2019.

[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).

[LDL+22] G. Larue, L-A. Dufrene, Q. Lampin et al. "Neural Belief Propagation Auto-Encoder for Linear Block Code Design”, IEEE Transactions On Communications, early access, 2022, DOI: 10.1109/TCOMM.2022.3208331.

[LEM21] L. Le Magoarou, “Efficient Channel Charting via Phase-Insensitive Distance Computation,” IEEE Wireless Commun. Lett., vol. 10, pp. 2634–2638, 2021.

[LKR+22] S. Lahmer, A. Khoshsirat, M. Rossi, and A. Zanella, “Energy consumption of neural networks on nvidia edge boards: an empirical model,” in 2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt), 2022, pp. 365–371.

[LMA+23] I. Labriji, M. Merluzzi, F. E. Airod, E. Calvanese Strinati, “Energy-efficient cooperative inference via adaptive deep neural network splitting at the edge,” accepted at IEEE ICC 2023, Rome, Italy

[LYP+22a] L. Le Magoarou, T. Yassine, S. Paquelet, and M. Crussière, “Deep Learning for Location Based Beamforming with Nlos Channels,” Singapore, Singapore, 2022, pp. 8812–8816.

[LYP+22b] L. Le Magoarou, T. Yassine, S. Paquelet, and M. Crussière, “Channel charting based beamforming,” Asilomar Conference on Signals, Systems, and Computers 2022.

[MAZ75] J. E. Mazo, “Faster-than-Nyquist signaling”, Bell System Technical Journal, vol. 54, no. 8, pp. 1451-1462, October 1975.

[MBS+21] Magoula, L., Barmpounakis, S., Stavrakakis, I., & Alonistioti, N. (2021). A genetic algorithm approach for service function chain placement in 5G and beyond, virtualized edge networks. Computer Networks, 195, 108157.

[MFB+22] M. Merluzzi, M. C. Filippou, L. G. Baltar and E. C. Strinati, "Effective Goal-oriented 6G Communications: the Energy-aware Edge Inferencing Case," 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Grenoble, France, 2022, pp. 457-462,

[MFB+23] M.Merluzzi, M. C. Filippou, L. Gomes Baltar, M. Muek, E. Calvanese Strinati, “6G goal-oriented communications: How to coexist with legacy systems?,” submitted to IEEE Transactions on machine learning in communications and networking, online at:https://www.techrxiv.org/articles/preprint/6G_goal_oriented_communications_How_to_coexist_with_legacy_systems_/22189879

[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.

[MLR22] Y. Matsubara, M. Levorato, and F. Restuccia, “Split computing and early exiting for deep learning applications: Survey and research challenges,” ACM Comput. Surv., mar 2022, just Accepted. [Online]. Available: https://doi.org/10.1145/3527155

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

[NCS+19] Noussan M, Carioni G, Sanvito FD, Colombo E. Urban Mobility Demand Profiles: Time Series for Cars and Bike-Sharing Use as a Resource for Transport and Energy Modeling. Data. 2019; 4(3):108. https://doi.org/10.3390/data4030108

[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.

[PKH+21] J. Pihlajasalo, D. Korpi, M. Honkala, J. M. J. Huttunen, T. Riihonen, J. Talvitie, A. Brihuega, M. A. Uusitalo, and M. Valkama, “HybridDeepRx: Deep learning receiver for high-EVM signals,” in Proc. IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Sep. 2021.

[PMGJCS+17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ser. ASIA CCS ’17. Association for Computing Machinery, 2017, p. 506–519.

[RDM+22] Renda, A.; Ducange, P.; Marcelloni, F.; Sabella, D.; Filippou, M.C.; Nardini, G.; Stea, G.; Virdis, A.; Micheli, D.; Rapone, D.; Baltar, L.G. Federated Learning of Explainable AI Models in 6G Systems: Towards Secure and Automated Vehicle Networking. *Information* **2022**, *13*, 395. https://doi.org/10.3390/info13080395

[RSP+20] N. Rathi, G. Srinivasan, P. Panda, K. Roy, Enabling Deep Spiking Neural Networks with Hybrid Conversion and Spike Timing Dependent Backpropagation, International Conference on Learning Representations, ICLR 2020

[RSR+21] N. Rajapaksha, K. B. Shashika Manosha, N. Rajatheva and M. Latva-Aho, "Deep Learning-based Power Control for Cell-Free Massive MIMO Networks," ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1-7.

[RU07] T. Richardson and R. Urbanke, Modern Coding Theory. Cambridge University Press, 2007.

[SAN18] M. Sandler et al., “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.[WWL+19] N. Wu, X. Wang, B. Lin, and K. Zhang, “A CNN-based end-to-end learning framework toward intelligent communication systems,” IEEE Access, vol. 7, pp. 110197–110204, 2019.

[SEV18] S. Makridakis, Spiliotis E., Assimakopoulos V., “Statistical and machine learning forecasting methods: concerns and ways forward”, PLoS ONE 13(3):e0194889.

[SMG+18] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, “Channel Charting: Locating Users Within the Radio Environment Using Channel State Information,” IEEE Access, vol. 6, pp. 47682–47698, 2018.

[SMSL+22] P. M. Santos, B. R. Manoj, M. Sadeghi, and E. G. Larsson, “Universal adversarial attacks on neural networks for power allocation in a massive mimo system,” IEEE Wireless Commun. Lett., vol. 11, no. 1, pp. 67–71, 2022

[SRBE+11] Sagduyu, Yalin Evren, Randall A. Berry, and Anthony Ephremides. "Jamming games in wireless networks with incomplete information." IEEE Communications Magazine 49.8 (2011): 112-118.

[SRI13] N. Srivastava, "Improving neural networks with dropout." University of Toronto 182, no. 566 (2013): 7.

[ST20] Santos, H. G., & Toffolo, T. A. (2020), "Mixed Integer Linear Programming with Python". URL: https://buildmedia.readthedocs.org/media/pdf/python-mip/latest/python-mip.pdf

[TAZ+13] A. Tzanakaki, M. P. Anastasopoulos, G. S. Zervas, B. R. Rofoee, R. Nejabati and D. Simeonidou, "Virtualization of heterogeneous wireless-optical network and IT infrastructures in support of cloud and mobile cloud services". IEEE Communications Magazine, vol. 51, no. 8, pp. 155-161, August 2013.

[TDL00] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, vol. 290, Art. no. 5500, Dec. 2000.

[TFS23] https://www.tensorflow.org/tfx/guide/serving

[TS38] "NR physical layer procedures for data," 3GPP TS38.214, 2018.

[TSP19] T. Jirsik, S. Trcka and P. Celeda, "Quality of Service Forecasting with LSTM Neural Network," 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, 2019, pp. 251-260.

[VIDBA+19] Vandikas, K., S. Ickin, G. Dixit, M. Buisman, and J. Åkeson. "Privacy-aware machine learning with low network footprint." Ericsson Research, Ericsson AB, Tech. Rep (2019).

[WZ20] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," IEEE Communications Magazine, vol. 58, no. 1, pp. 106–112, 2020.

[YBM20] Y. Yang, R. Bamler, and S. Mandt. "Variational Bayesian quantization." In International Conference on Machine Learning, pp. 10670-10680. PMLR, 2020.

[YL22] T. Yassine and L. Le Magoarou, "mpNet: variable depth unfolded neural network for massive MIMO channel estimation," IEEE Trans. Wireless Commun., vol. PP, p. 1, 2022.

[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.

[ZS17] M. Zhu, and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression." arXiv preprint arXiv:1710.01878 (2017).

[ZSBB21] Zvara, Z., Szabó, P., Balázs, B., & Benczúr, A. A. (2021, December). System-aware dynamic partitioning for batch and streaming workloads. In *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing* (pp. 1-10).

Annex A

In this section, we propose a technically detailed extension of the "Reserve" field (as illustrated in Figure 5-2) of the "Radio Application Package (RAP)" in order to accommodate for two additional requirements in the context of Hexa-X applications:

1) It is proposed to indicate the suitability of the RAP for usage in specific Hexa-X use cases. There are indeed different requirements depending on the target application – for example, code may require specific conformity checks prior to usage in an industrial environment where any malfunctioning may cause harm to persons.

Hexa-X Use Case Family	Suitability indication
Telepresence	Bit 1 set to "1" means suitability for this class of Use Cases. Suitability to specific Sub-Use-Cases are further detailed in a new bit-field (1st bit of new bit field indicates suitability to first sub-use-case as indicated by Fig. 1 for this class, etc.)
Robots to Cobots	Bit 2 set to "1" means suitability for this class of Use Cases. Suitability to specific Sub-Use-Cases are further detailed in a new bit-field (1st bit of new bit field indicates suitability to first sub-use-case as indicated by Fig. 1 for this class, etc.)
Massive Twinning	Bit 3 set to "1" means suitability for this class of Use Cases. Suitability to specific Sub-Use-Cases are further detailed in a new bit-field (1st bit of new bit field indicates suitability to first sub-use-case as indicated by Fig. 1 for this class, etc.)
Trusted embedded networks & Hyperconnected resilient network infrastructures	Bit 4 set to "1" means suitability for this class of Use Cases. Suitability to specific Sub-Use-Cases are further detailed in a new bit-field (1st bit of new bit field indicates suitability to first sub-use-case as indicated by Fig. 1 for this class, etc.)
Enabling Sustainability	Bit 5 set to "1" means suitability for this class of Use Cases. Suitability to specific Sub-Use-Cases are further detailed in a new bit-field (1st bit of new bit field indicates suitability to first sub-use-case as indicated by Fig. 1 for this class, etc.)

2) We furthermore consider the specific case that the RAP contains code the is considered to enable a "High Risk" application as defined by the draft AI Act [AIA]. In such a case, the application needs to fulfill a series of requirements and demonstrate compliance against those. It is thus proposed that the RAP contains a specific bit which indicates that the manufacturers guarantees compliance for specific AI "High Risk" categories. Only if the corresponding bit is set, the software component may be used in this specific context.

Bit 1 set to "1" means suitability for 1. Biometric identification and categorisation of natural persons: (a) AI systems intended to be used for the 'real-time' and 'post' remote biometric identification of natural persons;
Bit 2 set to "1" means suitability for 2. Management and operation of critical infrastructure: (a) AI systems intended to be used as safety components in the management and operation of road traffic and the supply of water, gas, heating and electricity.
Bit 3 set to "1" means suitability for 3. Education and vocational training: Bit 3 set to "1", sub-Bit 1 set to "1" means suitability for (a) AI systems intended to be used for the purpose of determining access or assigning natural persons to educational and vocational training institutions; Bit 3 set to "1", sub-Bit 2 set to "1" means suitability for (b) AI systems intended to be used for the purpose of assessing students in educational and vocational training institutions and for assessing participants in tests commonly required for admission to educational institutions.
Bit 4 set to "1" means suitability for 4. Employment, workers management and access to self-employment: Bit 4 set to "1", sub-Bit 1 set to "1" means suitability for (a) AI systems intended to be used for recruitment or selection of natural persons, notably for advertising vacancies, screening or filtering applications, evaluating candidates in the course of interviews or tests; Bit 4 set to "1", sub-Bit 2 set to "1" means suitability for (b) AI intended to be used for making decisions on promotion and termination of work-related contractual relationships, for task allocation and for monitoring and evaluating performance and behavior of persons in such relationships.

Bit 5 set to “1” means suitability for 5. Access to and enjoyment of essential private services and public services and benefits:

Bit 5 set to “1”, sub-Bit 1 set to “1” means suitability for (a) AI systems intended to be used by public authorities or on behalf of public authorities to evaluate the eligibility of natural persons for public assistance benefits and services, as well as to grant, reduce, revoke, or reclaim such benefits and services;

Bit 5 set to “1”, sub-Bit 2 set to “1” means suitability for (b) AI systems intended to be used to evaluate the creditworthiness of natural persons or establish their credit score, with the exception of AI systems put into service by small scale providers for their own use;

Bit 5 set to “1”, sub-Bit 3 set to “1” means suitability for (c) AI systems intended to be used to dispatch, or to establish priority in the dispatching of emergency first response services, including by firefighters and medical aid.

Bit 6 set to “1” means suitability for 6. Law enforcement:

Bit 6 set to “1”, sub-Bit 1 set to “1” means suitability for (a) AI systems intended to be used by law enforcement authorities for making individual risk assessments of natural persons in order to assess the risk of a natural person for offending or reoffending or the risk for potential victims of criminal offences;

Bit 6 set to “1”, sub-Bit 2 set to “1” means suitability for (b) AI systems intended to be used by law enforcement authorities as polygraphs and similar tools or to detect the emotional state of a natural person;

Bit 6 set to “1”, sub-Bit 3 set to “1” means suitability for (c) AI systems intended to be used by law enforcement authorities to detect deep fakes as referred to in article 52(3);

Bit 6 set to “1”, sub-Bit 4 set to “1” means suitability for (d) AI systems intended to be used by law enforcement authorities for evaluation of the reliability of evidence in the course of investigation or prosecution of criminal offences;

Bit 6 set to “1”, sub-Bit 5 set to “1” means suitability for (e) AI systems intended to be used by law enforcement authorities for predicting the occurrence or reoccurrence of an actual or potential criminal offence based on profiling of natural persons as referred to in Article 3(4) of Directive (EU) 2016/680 or assessing personality traits and characteristics or past criminal behaviour of natural persons or groups;

Bit 6 set to “1”, sub-Bit 6 set to “1” means suitability for (f) AI systems intended to be used by law enforcement authorities for profiling of natural persons as referred to in Article 3(4) of Directive (EU) 2016/680 in the course of detection, investigation or prosecution of criminal offences;

Bit 6 set to “1”, sub-Bit 7 set to “1” means suitability for (g) AI systems intended to be used for crime analytics regarding natural persons, allowing law enforcement authorities to search complex related and unrelated large data sets available in different data sources or in different data formats in order to identify unknown patterns or discover hidden relationships in the data.

Bit 7 set to “1” means suitability for 7. Migration, asylum and border control management:

Bit 7 set to “1”, sub-Bit 1 set to “1” means suitability for (a) AI systems intended to be used by competent public authorities as polygraphs and similar tools or to detect the emotional state of a natural person;

Bit 7 set to “1”, sub-Bit 2 set to “1” means suitability for (b) AI systems intended to be used by competent public authorities to assess a risk, including a security risk, a risk of irregular immigration, or a health risk, posed by a natural person who intends to enter or has entered into the territory of a Member State;

Bit 7 set to “1”, sub-Bit 3 set to “1” means suitability for (c) AI systems intended to be used by competent public authorities for the verification of the authenticity of travel documents and

supporting documentation of natural persons and detect non-authentic documents by checking their security features; Bit 7 set to "1", sub-Bit 4 set to "1" means suitability for (d) AI systems intended to assist competent public authorities for the examination of applications for asylum, visa and residence permits and associated complaints with regard to the eligibility of the natural persons applying for a status.
Bit 8 set to "1" means suitability for 8. Administration of justice and democratic processes: Bit 8 set to "1", sub-Bit 1 set to "1" means suitability for (a) AI systems intended to assist a judicial authority in researching and interpreting facts and the law and in applying the law to a concrete set of facts.

The full bit-field is summarised below considering all possible items 1…8 above:

High Risk AI Bits = 00, 01, 10, "11" (00 = no AI application, 01=AI application but no high risk AI application, 10 = high risk AI application, 11 = forbidden AI Application)							
Bit 1 = 0/1	Bit 2 = 0/1	Bit 3 = 0/1	Bit 4 = 0/1	Bit 5 = 0/1	Bit 6 = 0/1	Bit 7 = 0/1	Bit 8 = 0/1
No sub-Bit needed (only 1 sub-item)	No sub-Bit needed (only 1 sub-item)	Sub-Bits 1…2 = 0/1	Sub-Bits 1…2 = 0/1	Sub-Bits 1…3 = 0/1	Sub-Bits 1…7 = 0/1	Sub-Bits 1…4 = 0/1	Sub-Bit 1 = 0/1

The proposed framework on flexible compute workload assignment aims to enable intelligent and flexible workload delegation, factoring in 6G use case's requirements as well as end user/ developer capabilities. The proposed solution may be relevant to all Hexa-X 6G UCFs, as described in [HEX-D13].