Project Report

CELTIC-NEXT AIMM Project

WP4: AI for Radio Resource Optimisation

Deliverable D4.2

Authors: Stephan ten Brink, Florian Euchner (UST), Yejian Chen, Thorsten Wild (NBLS), Arman Shojaeifard, Deepa Jagyasi (InterDigital), Cliff Ellement (ThinkRF)

Project Acronym:	**AIMM**
Project Full Title:	**AI-enabled Massive MIMO**
Project Coordinator:	**Arman Shojaeifard (InterDigital)**
Project Duration:	**24 months (Oct. 2020 – Sep. 2022)**
Submission Date:	**31 September 2022 (M24)**
Dissemination Level:	**External**

Abstract

This report provides a summary of activities from the work within the 24 months of the CELTIC-NEXT AIMM project Work-Package 4 (WP4) on “AI for radio resource optimisation”. These include all updates and results to date from all topics of work within WP4, namely on “Channel Estimation & MIMO Detection”, ”User Localisation & Channel Charting”, and “RF Detection and Location”, all of which incorporate tools from AI/ML for design and operation.

Executive Summary

This report describes the activities that have taken place to date (M24) within the CELTIC-NEXT AIMM project Work-Package 4 (WP4) on “AI for Radio Resource Optimisation”.

The focus of this work package is the application of ML to various problems in radio resource optimisation. These comprise real-time physical layer problems, such as channel estimation, MIMO detection, and rate adaptation, as well as problems with less stringent timing requirements, such as user scheduling and localisation. Also, the advanced idea of inferring on the downlink channel state information for basestation (multiuser) precoding based on the observed uplink channel using deep learning will be investigated.

This deliverable provides information on the progress made against all these topics of work within WP4. Further, dissemination activities and future work plans are provided.

In summary, AIMM WP4 has been progressing in accordance with the project plan and the project partners will exploit the results of this work-package accordingly.

Table of Contents

Abbreviations

Abbreviation	Definition
3G	Third generation cellular
3GPP	Third Generation Project Partnership
4G LTE/LTE-A	Fourth generation cellular Long Term Evolution/Long Term Evolution Advanced
5G NR	Fifth generation cellular New Radio
A1	O-RAN interface between Non-RT RIC and Near-RT RIC
AAS	Active antenna system
AARX	Antenna array as receiver
AI/ML	Artificial Intelligence/Machine Learning
BS	Base station
CAPEX	Capital expenditure
CoMP	Coordinated multipoint
CPRI	Common Public Radio Interface
CQI	Channel Quality Indicator
CSI	Channel State Information
CS-RS	Cell-Specific Reference Signal
CU	Centralised unit
DCI	Downlink Control Indicator
DMRS	Demodulation Reference Signal
DPB	Dynamic Point Blanking
DPC	Dirty-paper-coding
DPD	Digital Pre-Distortion
DPS	Dynamic Point Selection
DSP	Digital signal processing
DU	Distributed unit
E2	O-RAN interface between Near-RT RIC and CUs/DUs
eCPRI	Enhanced Common Public Radio Interface
EM	Electromagnetic
eNB	eNodeB (4G LTE/LTE-A base station)
EVM	Error vector magnitude
F1	3GPP interface between CU and DU
FD MIMO	3D/full-dimension MIMO
FR1	Frequency range 1
FR2	Frequency range 2
FWA	Fixed wireless access
gNB	gNodeB (5G NR base station)
HBF	Holographic Beamforming
HLS	Higher-layer-split
IPR	Intellectual property rights
IRS	Intelligent Reflecting Surface
KPI	Key-performance-indicator
L#	Layer number # on the protocol stack
LLS	Lower-layer-split
LMS	Least Mean Squares
LOS	Line-of-sight
MAC	Medium Access Control
MDT	Minimisation of drive test
MIMO	Multiple-input multiple-output
ML	Machine Learning
MOBTX	Mobile transmitter
MRT	Maximum-ratio-transmission
M-TRP	Multi transmission/reception points
NMSE	Normalized Mean Squared Error
NN	Neural Network
Near-RT	Near-real-time
Non-RT	Non-real-time
OPEX	Operational expenditure

O-RAN	O-RAN Alliance
Open RAN	Ecosystem for open standardised interfaces implementation
OTA	Over-The-Air
PA	Power amplifier
PBCH	Physical Broadcast Channel
PDCP	Packet Data Convergence Protocol
PDSCH	Physical Downlink Shared Channel
PHY	Physical Layer
PRACH	Physical Random Access Procedure
PRB	Physical Resource Block
PSS	Primary Synchronisation Signal
PUSCH	Physical Uplink Shared Channel
QoE	Quality-of-experience
QoS	Quality-of-service
RAN	Radio access network
rApp	An application designed to run on the Non-RT RIC
REFTX	Reference transmitter
RIC	O-RAN RAN Intelligent Controller
RIS	Reconfigurable Intelligent Surfaces
RIT	Radio Interface Technology
RLC	Radio Link Control
RLS	Recursive Least Squares
RRC	Radio Resource Control
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RT	Real-time
RU	Radio unit
SE	Spectral efficiency
SINR	Signal-to-interference-plus-noise ratio
SISO	Single-input single-output
SNR	Signal-to-Noise Ratio
SON	Self-organising-network
SRIT	Set of Radio Interface Technologies
SSB	System synchronisation block
SSS	Secondary synchronisation Signal
TXRU	Transceiver chain
UE	User equipment
vRAN	Virtualised RAN
X2	3GPP interface between eNBs
xApp	An application designed to run on the Near-RT RIC
Xn	3GPP interface between gNBs
ZF	Zero-forcing

1 Introduction

MIMO is a key air-interface technology in nearly all modern communications systems [1]. MIMO, through utilization of multiple antennas at the radios, can provide several benefits including enhancing spectral efficiency and quality-of-service [2]. Despite significant performance improvements achieved through MIMO to date, there exists a significant gap between the theoretical versus practical performance of multi-antenna systems [3].

Motivated by the above, the AIMM project targets radical performance improvements and efficiency dividends for 5G and beyond MIMO systems through adoption of AI/ML capabilities in both link-level and system-level RAN domains, as well as alternative deployment methods including radio intelligent surfaces and cell-less antenna systems. To achieve the set targets, the AIMM project work is divided between six tightly coupled work-packages, as illustrated in Fig. 1 below.

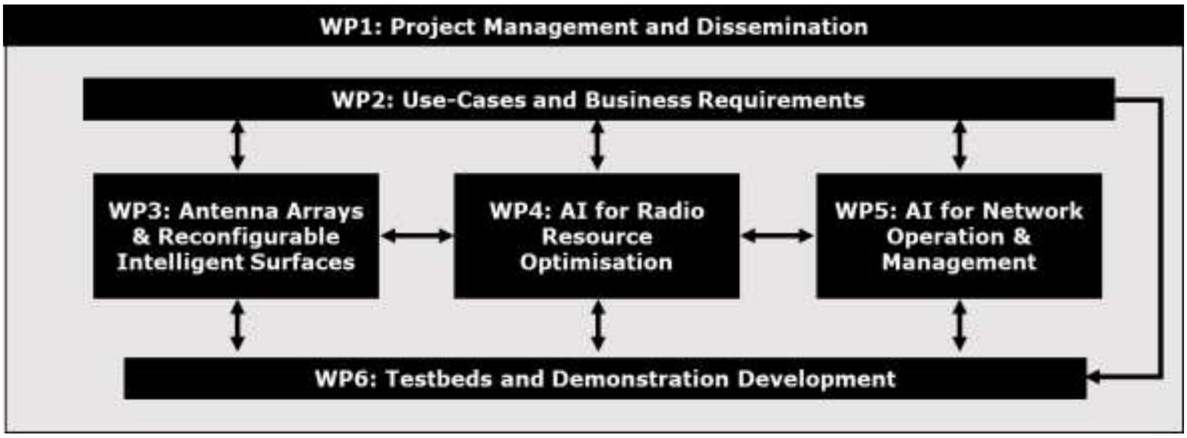

Figure 1: AIMM project work-package structure.

This document provides an outline of the work on AI for Radio Resource Optimisation as part of the AIMM Work-Package 4 (WP4). The technologies of interest target physical (PHY) layer based use cases on Channel Estimation and MIMO Detection, User Localisation and Channel Charting, Channel State Information (CSI) Acquisition in Frequency Division Duplex (FDD) Massive MIMO, and RF Anomaly Detection, all of which involving the adoption of AI/ML for systems operation and optimisation.

This deliverable provides a summary of the work that has been carried out to date within this AIMM project WP4. Further, dissemination activities and future work plan within this work-package are accordingly highlighted in this report.

2 Technical Work Progress

This section provides progress reports by WP4 participants around the technical work that has been carried out to date within this work-package.

2.1 Task 4.1: Channel Estimation & MIMO Detection

Channel estimation and symbol detection are of utmost importance for Massive MIMO systems. However, state-of-the-art algorithms for both problems come at tremendous implementation complexities which render their practical use prohibitive. ML provides a different approach to both problems with the potential to achieve close to or even beyond state-of-the-art performance with reduced complexity. The goal of this task is to develop new ML-based solutions to channel estimation and symbol detection which will be evaluated under realistic conditions, that is, channels generated through simulations (3GPP 3D MIMO Model), raytracing, or actual measurements. An emphasis will be put on the problems of online learning, that is, training ML models in the field, and transfer learning, that is, how ML models can be adapted from one cell to the other with minimal retraining.

2.1.1 Exploit GPU to Verify DMRS-based Channel Estimation with Machine Learning

Within the context of AIMM project, we reached our objective toward D4.1 by extending the existing one-dimensional Machine Learning (ML) based estimator [4] and conventional *Turbo-AI* [11], [12] to a Demodulation Reference Signal (DMRS) based channel estimator. The proposed approach, named as *DMRS-Turbo-AI* [13], can support the scenarios with ultra-high mobility, or ultra-sparse pilot structure equivalently.

The link level performance further motivates us to verify *Turbo-AI* in GPU implementation D4.2. In Fig. 2, the Over-The-Air (OTA) data is generated with Spirent VR5, a configurable RF channel emulator, by randomly capturing the two consecutive Physical Resource Blocks (PRB) as basic processing units. The DMRS of these PRB-pairs can be exploited to create labels and observations under different Signal-to-Noise Ratios (SNR) for training the Neural Network (NN) in frequency and time domains.

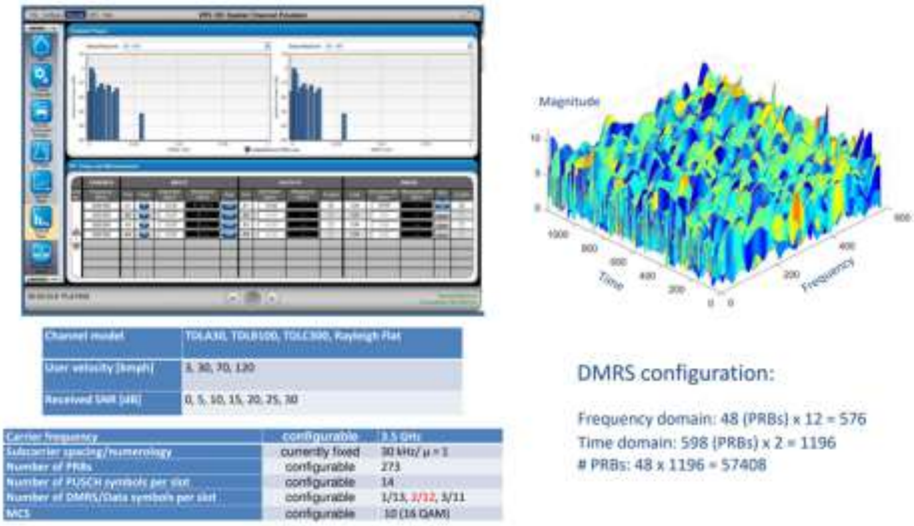

Figure 2: Capture the Over-The-Air (OTA) data for training the NNs.

In Fig. 3, it is illustrated how the observations for training are arranged out of the OTA data. Notice that the correlation in time domain within a PRB-pair can be obtained only by 2 symbols, and that in frequency domain is obtained through 12 consecutive symbols. We thus slightly modify the arrangement for the inputs of the time domain NN, which jointly introduce frequency domain components. Thus, both frequency and time domain NNs are with the same network architecture, which can not only guarantee the robust performance of the second iteration in time domain, but also allow hardware sharing between the first and second iteration.

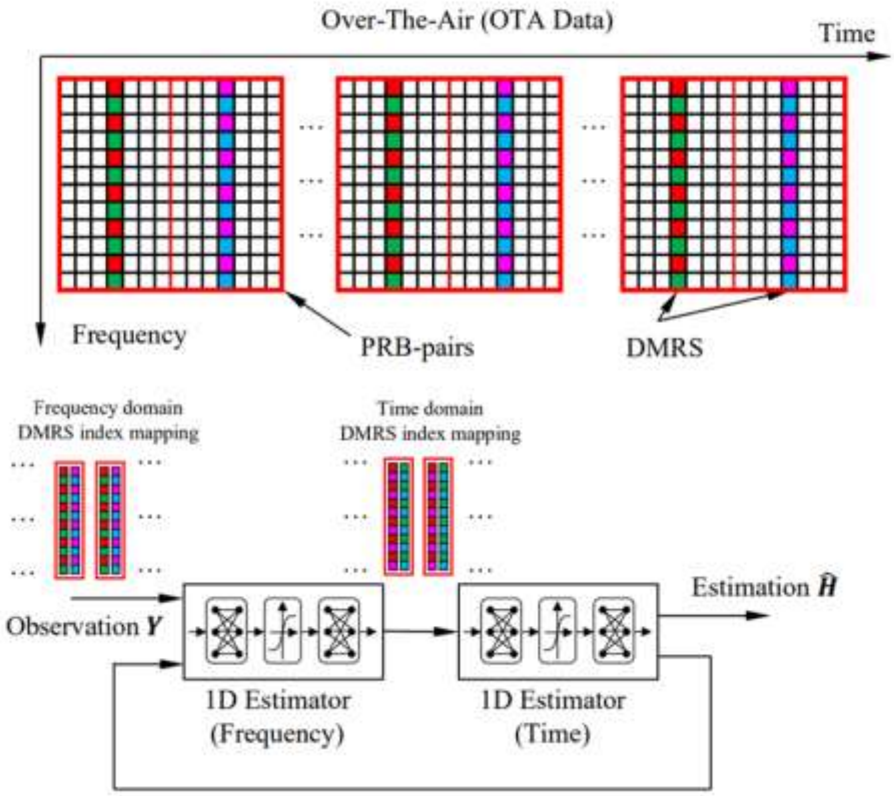

Figure 3: Train the NNs for frequency and time domains to 2D Turbo-AI.

Fundamentally, the NNs are trained, based on the TDLA30 and TDLC300 channels, and later on the pre-trained NN models will be verified for TDLB100 channel. Since the observations are collected for randomly selected SNR values, the trained NN model should exhibit universal availability within a wide range of SNRs.

In Fig. 4, the trained NNs are verified at 0dB SNR, by randomly selecting a PRB-pair out of a testing data set to carry out *2D Turbo-AI* and repeat this procedure for 10000 independent realizations. The measurement of the channel estimation Normalized Mean Squared Error (NMSE) demonstrates that the channel estimates can be iteratively improved, even by testing such small data pieces.

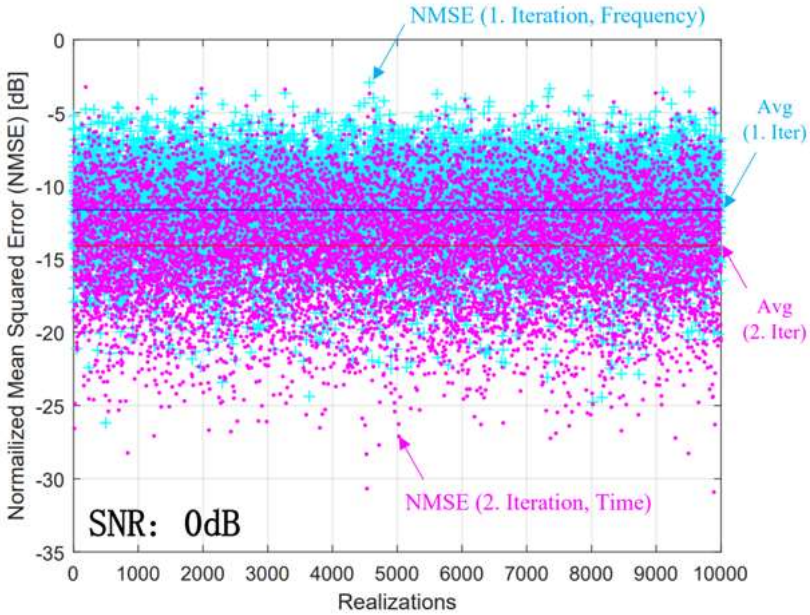

Figure 4: Verify the NNs in 2D Turbo-AI for 10000 realizations at SNR 0dB, TDLB100 channel, 3km/h.

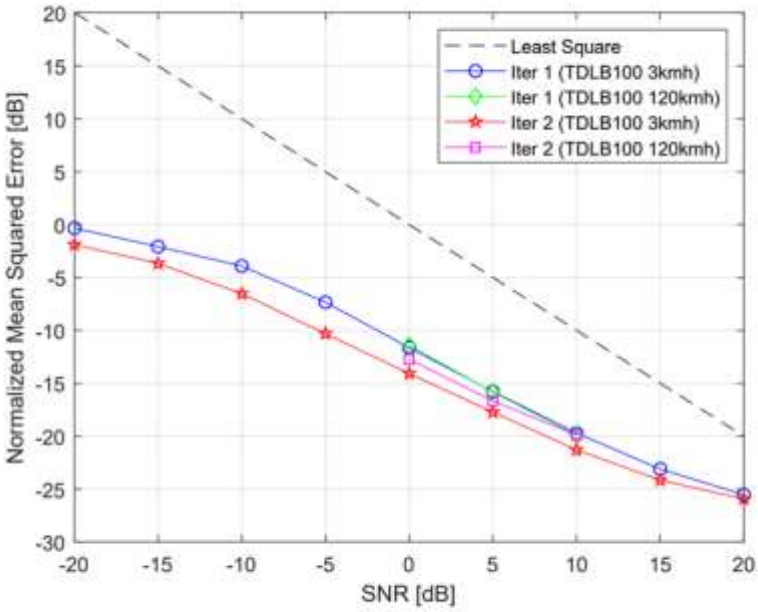

Figure 5: NMSE versus SNR for 2D Turbo-AI.

In Fig. 5, the NMSE versus SNR performance curves are summarized for verifying *2D Turbo-AI* in TDLB100 channel. The least square estimator provides an upper bound performance as raw channel estimation. For the low-speed scenario at 3km/h, 1dB to 3dB estimation gain through time domain can be clearly observed. For the high-speed scenario at 120km/h, the time domain gain will be reduced, which is mainly caused by loss of temporal correlation at high speed.

The link level results summarized above can be regarded as a kind of preparation for a cross WP activity internally in Nokia between WP4 and WP6, in which WP4 transfers the existing pre-trained NN-models of frequency and time domains to WP6. And WP6 integrates the pre-trained NNs to a GPU-based Soft gNB for realizing the whole *2D Turbo-AI* procedure. With the real-time generated OTA data, the quality of communication can be instantaneously demonstrated and evaluated by diverse system metrics, based on the equipment, presented in a separate deliverable D6.2 of AIMM project.

2.1.2 Deep Learning-based DMRS Channel Estimation

Despite the great success of MIMO to date, there exists a significant gap between the theoretical versus practical performance of these systems. As a result, significant efforts are ongoing within both cellular (5G NR Rel-17 feMIMO WI [6]) and Wi-Fi (IEEE 802.11be EHT MIMO Protocol Enh. [7]) standards to further enhance MIMO operations.

Today's MIMO systems utilise user-specific reference signals (pilots) in order to allow for coherent demodulation of the precoded/beamformed signals at the receiver. For example, 5G NR standards use user-specific demodulation reference signals (DMRS) for the purpose of composite channel estimation (CCE), achieved through applying the same precoding/beamforming weights to DMRS that are used on the downlink and uplink data and control physical channels, as illustrated in Fig. 6. Current 5G NR specifications allow for some flexibility in DMRS configuration to cater for different UE capabilities and use cases. For example, for NR PDSCH DMRS, there are Configuration Type 1 vs Type 2, Mapping Type A vs Type B, Starting Symbol for Mapping Type A, Single vs Double Symbol DMRS, DMRS Additional Positions, and Duration [8].

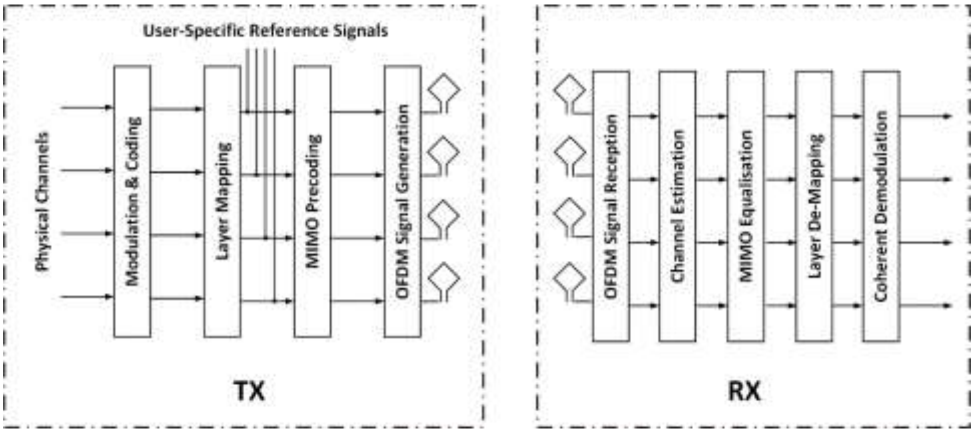

Figure 6: Transceiver chain for MIMO systems depicting the use of user-specific reference signals accompanying physical channels for the purpose of composite channel estimation.

Under existing DMRS configuration settings, the channel estimation is a non-linear problem at the traditional estimation methods (e.g., Wiener filter) are strictly sub-optimal. Applying tools from AI/ML, in particular deep learning, to the channel estimation framework as well as channel estimation algorithms can provide benefits by reducing DMRS overhead, enhancing channel estimation performance, reducing receiver computational complexity, or a combination of these. Looking further ahead, entirely new features and air-interface design for channel estimation can be achieved through the applications of AI/ML (e.g., autoencoders). These aspects involve deep learning-based DMRS channel estimation in beyond 5G NR MIMO systems, with a specific focus around the impact on the terminal side.

Here, we provide a proof-of-concept implementation to demonstrate the core principle for the adaptive user-specific reference signal configuration using tools from AI/ML. The various assumptions and modes of operation described in this section are to facilitate the demonstration of the core principle, and indeed these can be relaxed and extended through proper adjustments.

Consider a scenario where the network provides the terminal with both the anchor DMRS configuration in addition to the actual pilots accompanying transmission of precoded/beamformed physical channels. The anchor configuration can be based on current 5G NR specifications, in terms of the available patterns and variables for the corresponding type of physical channel. The terminal utilises the DMRSs to proceed with CCE and coherent demodulation. The channel estimation measurements from the anchor reference signal configuration are then fed into an (offline) AI/ML engine where the output of the model can provide inference on the preferred configuration of the DMRS for subsequent payloads. For the configuration of DMRS, in terms of position and density in time, frequency, and code domains, we consider two systems, a baseline (non-adaptive) system based on 5G NR specifications, and a proposed adaptive system where the configuration is dynamically decided and signalled back to the network by an AI/ML engine residing at the terminal. A high-level diagram of the proposed framework is provided in Fig. 7.

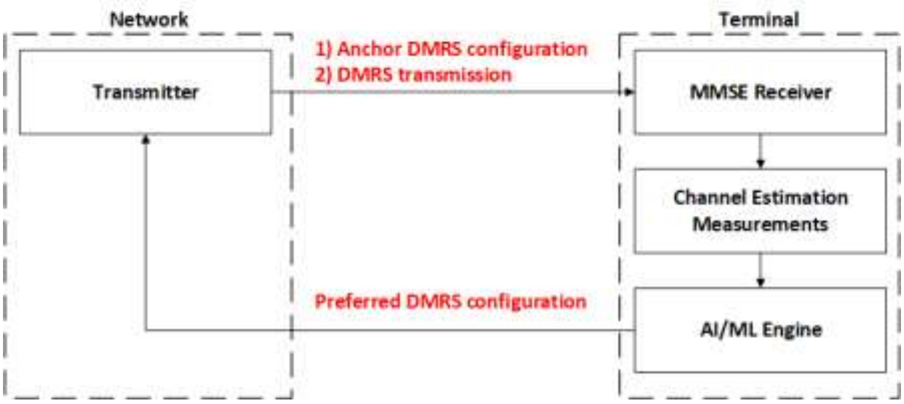

Figure 7: Proof-of-concept implementation of AI/ML-based adaptive reference signal configuration for MIMO CCE.

It is important to note that both systems utilise the same receiver algorithm (here, a practical MMSE solution). MMSE aims to minimize the MSE between the estimate and actual composite channel values of the resource elements (REs) carrying DMRSs. We assume that the second-order channel statistics (including expected value and channel covariance matrix) are available as priori information for the MMSE receiver. These estimated channel values are used to perform interpolation/extrapolation to estimate the missing values from the channel estimation grid for each antenna port After MIMO CCE, the receiver proceeds with the coherent demodulation operation. The accuracy of the MIMO CCE process is here assessed by the MSE performance. The choice of a conventional receiver algorithm (MMSE) here provides a key benefit in terms of allowing the leveraging of well-known models from information theory for the collection of labelled data for the AI/ML engine.

Some evaluation results based on a proof-of-concept implementation of the proposed deep learning-based adaptive DMRS configuration framework are provided next. Specifically, we use a 5G NR-compliant link-level simulator (LLS) for capturing the end-to-end physical downlink shared channel (PDSCH) transceiver chain for a 2×2 downlink MIMO system between a base station and a terminal. Synthetic standards-compliant 5G NR waveforms and channel models are used for the training of the proposed adaptive system. In order to assess the model under a range of scenarios, we collect and feed data to a deep neural network based on a wide range of signal-to-noise (SNR) values (between -10 to 10 dB), Doppler shift values (between 1 to 600 Hz), and delay spread values (between 10 and 500 ns). To prevent over-fitting, we split the the generated synthetic data into training and validation groups, where the latter is used to assess the performance of the trained deep learning model at certain intervals. The scheduled transmission resource grid for each slot is considered to consist of 10 resource blocks of 12 subcarriers in frequency, and 14 OFDM symbols in time. The modulation scheme used is 16-QAM, and the baseline pilot configuration is single-symbol PDSCH DMRS Configuration Type 1, Mapping Type A (starting symbol 3), with one additional position (at symbol 11).

Fig. 8 depicts a sample performance gain of the trained deep learning-based adaptive DMRS configurator over the baseline case for the settings of system parameters described in this section. It can be seen that for a given antenna port, the proposed system has achieved a better MSE performance with a reduced DMRS overhead.

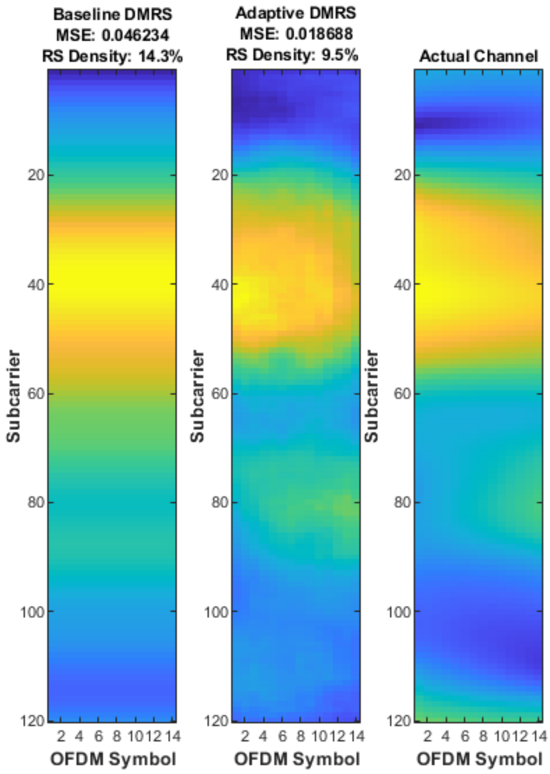

Figure 8: Sample performance of the proposed deep learning-based adaptive DMRS configurator over the NR baseline scheme for antenna port 1000.

2.1.3 Evolutionary Beamforming for Massive MIMO

Massive MIMO systems consist of many transmitting and/or receiving antenna elements. This gives rise to a high dimensional joint optimization problem when performing optimal beamforming. Without simplified assumptions about channel conditions, rigorously solving such problems consumes significant computation power and incurs high processing delay. In small-scale MIMO systems, a widely recognized near-optimal beamforming method is vector perturbation, which performs sphere encoding to solve the embedded integer least squares problem. Unfortunately, properties of classical sphere encoding algorithms are not yet well understood, besides the fact that their performances tend to degrade in higher dimensional signal space, where the end-to-end performance gap between suboptimal and optimal beamforming techniques is no longer negligible.

To mitigate the impaired performance-complexity trade-off of sphere encoding in massive MIMO system, we draw inspiration from recent progresses in practical lattice algorithms. In a prior investigation of *joint transmission in wired systems* [9], an evolutionary algorithm enabled stochastic sphere encoding strategy was studied as a counterexample to the presumed performance advantage of classical sphere encoding frameworks (Fig. 9). However, due to the exponentially increasing size of the search space (w.r.t. the dimension of the integer least squares problem) and the relative shortage of computation memory in practical network hardware, frequently processing a large population of intermediate solutions holistically (as performed by most evolutionary algorithms) is memory inefficient.

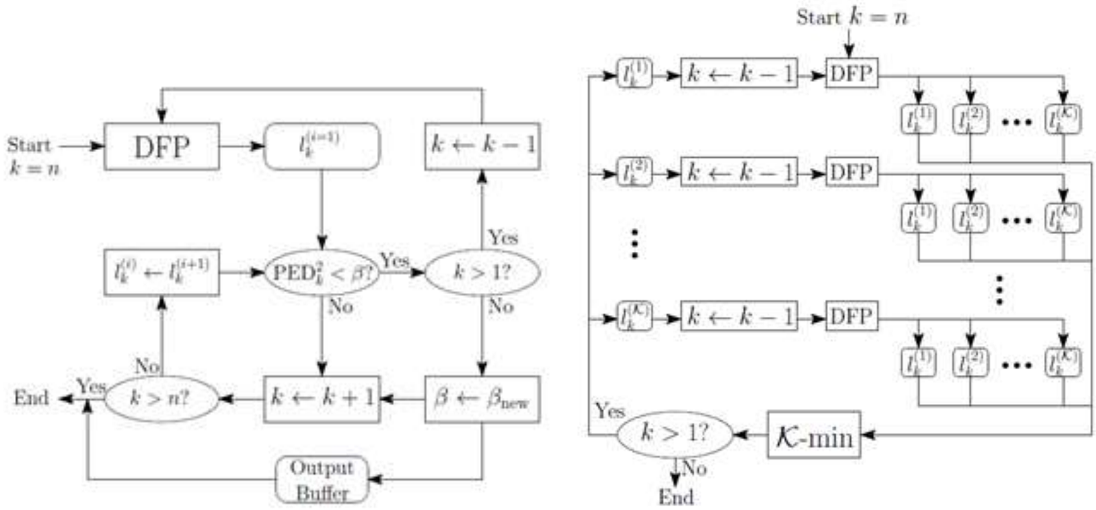

Figure 9: Depth first (left) and K-best (right) sphere encoder [9].

For improving the performance complexity trade-off of stochastic sphere encoding over the prior art conceived in [9], a novel iterative technique termed as cellular evolution has been submitted for patent application. The new art abandons the conventional large-population-oriented mutation-crossover-selection workflow in favour of a more mutually independent architecture, such that the parallelism and memory efficiency are both improved. Further technical details regarding this patent application will become available around October 2023.

2.2 Task 4.2: User Localisation & Channel Charting

User localisation through radio signals is becoming increasingly important and it is expected that location accuracy will be a key performance requirement of Beyond 5G systems, similar to latency and throughput. While model-based approaches work well in line-of-sight propagation conditions, their precision typically suffers in complex indoor and urban outdoor environments. For this reason, various groups have leveraged ML for user localisation based on CSI that is already available at Massive MIMO receivers.

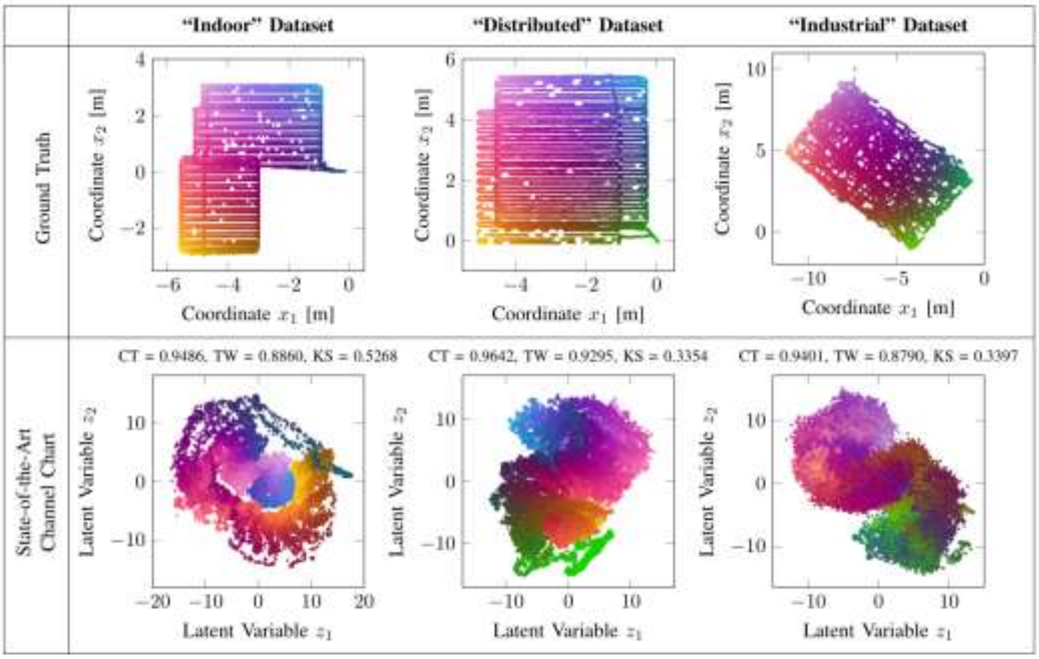

Figure 10: Channel charts for three different datasets, generated using a state-of-the-art triplet neural network-based Channel Charting pipeline.

Own experiments as well as experiments carried out by other groups have shown that localization based on CSI data is possible using neural networks trained in a supervised manner. Unfortunately, supervised training is not practical due to its need for large quantities of labelled training data, that is, CSI with precise "ground truth" position information, which might be difficult to acquire in large or fast-changing environments. Channel Charting [10] has been proposed as an alternative, which can create a map of *relative* UE positions without requiring any labels. That is, except for timestamps, which are almost certainly available anyway. Channel Charting requires even larger quantities of unlabelled training data, which, thanks to a newly developed massive MIMO channel sounder dubbed DICHASUS (<u>Di</u>stributed <u>Cha</u>nnel <u>S</u>ounder by <u>U</u>niversity of <u>S</u>tuttgart) [14], we are able to capture.

Initially, the focus in this task is the application of the state-of-the-art triplet neural network-based channel charting pipeline from literature to our own datasets. This pipeline relies on a *triplet neural network*, that is, a special kind of neural network architecture that can learn the structure of dataset from a large set of triplets consisting of an anchor sample, a far sample and a near sample. For successful training, it only has to be ensured that anchor and near sample are close to each other, whereas the far sample must be located at a great distance from the anchor sample. In literature, triplets are selected entirely based on the timestamps of datapoints.

Fig. 10 shows ground truth position data and channel charts for three different datasets captured by DICHASUS. To draw the points in channel charts, the individual datapoints in the dataset were assigned a color according to their "ground truth" position according to a color gradient. This color of the point in physical space is then preserved for the channel chart so similar coloring indicates datapoints which where actually measured in proximity of each other. Clearly, the channel charts preserve the local geometry of space, but fail to capture the global structure of the dataset. Also, even though all datasets contain a similar number of datapoints, the channel chart of the dataset measured in an "Industrial" environment resembles the ground truth much more closely. This raises the question of whether this is due to the available amount of data, due to properties of real-world datasets or a result of shortcomings of the current feature engineering or dimensionality reduction methods.

To address these issues, several changes and improvements to training methods and triplet selection were implemented and evaluated. Most importantly though, it has been shown that almost perfect reconstruction of even the global geometry is possible just as long as enough training data is available. This can be seen when performing *genie-aided* triplet selection: Instead of relying purely on timestamps for triplet generation, the "ground truth" position data (which would not be available in a real-world system) is taken into account. Thereby, the near sample is guaranteed to lie within a certain radius of the anchor sample. The critical advantage of the genie-aided method, however, is not just that it can ensure actual physical proximity, but that now, a much larger variety of close samples can be found for each anchor point. Where previously, only points that lie on the physical trajectory of the UE could be considered as potential near samples, now all even datapoints that were captured at much earlier or later points in time, but that happen to be close are taken into account. The channel charts generated using genie-aided triplet selection clearly preserve both global and local structure of the dataset.

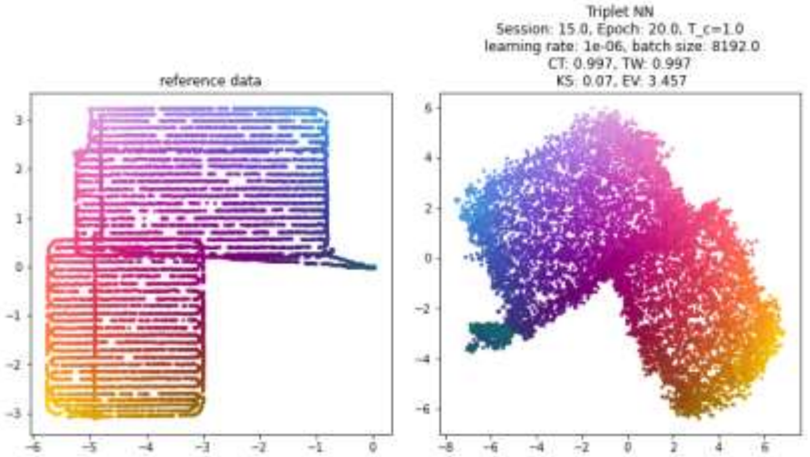

Figure 11: Ground truth position data (left) and channel chart generated from CSI (right) with "simulated trajectory"-aided triplet generation.

An additional genie-aided method that also bears practical ramifications is genie-aided generation of "simulated trajectories". Here, the idea is to generate realistic trajectories through the dataset, that the UE could, in theory, have taken. Experiments have shown that trajectories along straight lines perform no worse than curved trajectories with regards to channel charting, so only trajectories along straight lines are considered here. After trajectories are generated, virtual timestamps are assigned to trajectories and triplet selection is performed as in the state-of-the-art approach from literature. Fig. 11 shows a typical channel chart obtained through "simulated trajectory"-based triplet selection, for the "Indoor" dataset. Compared to the corresponding channel chart in Fig. 10, the chart in Fig. 8 clearly shows much more similarity to the ground truth locations. This shows that channel charting performance improves with larger quantities of training triplets, even if these triplets are simulated.

Of course, a practical channel charting system will not have access to ground truth data for genie-aided learning, so it is important to be able to demonstrate the feasibility of channel charting even under those circumstances. The promising results with "simulated trajectories" motivated an additional measurement campaign with the objective of generating a dataset which might replicate a similar performance enhancement with more and more diverse trajectories through space. This time, data was captured in an industrial environment, but with a distributed antenna setup.

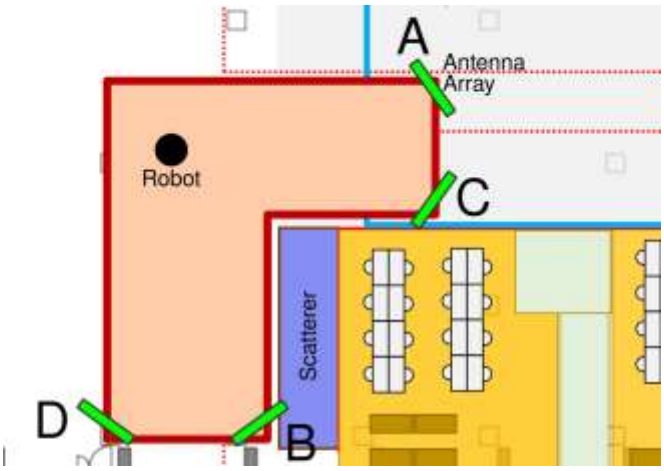

Figure 12: Distributed antenna setup in industrial environment.

The four receiver antenna arrays marked in green in Fig. 12 are placed in the corners of the area that the robot equipped with the transmitter antenna moves around in. From every point inside the area, and hence every transmitter position, there is at least one line-of-sight path to some receiver antenna array. From the experiments on “simulated trajectories”, we learned that the dataset should contain a large number of trajectories that are distributed over the whole considered area and that point in different directions in order to achieve good channel charting performance. Therefore, the new dataset consists of three meandering paths, each of them covering the whole area. These paths mainly differ in the orientation of the meanders: looking at the area in a top view map as in Figure 12, the three paths contain horizontal, vertical and diagonal meanders.

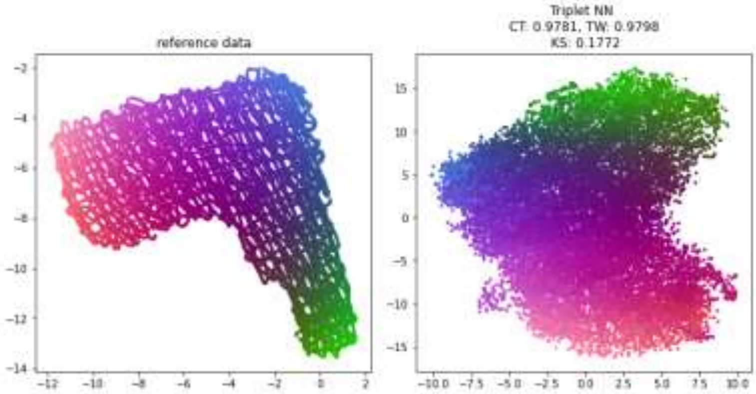

Figure 13: Ground truth position data (left) and channel chart generated from CSI (right) with new dataset.

While these six different directions (horizontal / vertical / diagonal orientations and forward / backward for each orientation) represent just some of the possible trajectories, they appear to be sufficiently diverse to allow for a reconstruction of a map of the area. With this more diverse dataset, the state-of-the art method using time-based triplet selection is expected to deliver better channel charts. Indeed, the resulting channel chart in Fig. 13 does not only preserve the local, but also the global structure of the ground truth positions, which was not the case for some of the previous datasets. The slightly inferior performance compared to the “simulated trajectory”-approach can be explained by the limited number of different trajectories and directions. However, this experiment shows that channel charting with time-based triplet selection is able to create meaningful channel charts under the right circumstances. In practice, it is unlikely for users to follow such idealized trajectories. Thus, further investigation on similarity-based triplet selection could play also a role for practical channel charting.

To enable as many research groups as possible to easily apply channel charting on DICHASUS data, a comprehensible tutorial on triplet-based channel charting has been published on the DICHASUS website: https://dichasus.inue.uni-stuttgart.de/tutorials/tutorial/channelcharting/

In addition to our own experiments based on DICHASUS datasets, we also invite the scientific community at large to participate in our research efforts. To this end, we have set up a website as a distribution channel for position-labelled CSI datasets: https://dichasus.inue.uni-stuttgart.de/ (see Fig. 8). A few datasets have already been published, with many more to come in the course of the next several months.

2.3 Task 4.3: Reduction of CSI Feedback for Massive MIMO FDD Operation

The advantages of Massive MIMO are best leveraged in the time division duplex (TDD) mode of operation, where uplink (UL) and downlink (DL) channel use the same frequency band. In the TDD case, due to channel reciprocity, no explicit reporting of DL channel state information (CSI) is required (from the mobile terminal through the UL to the basestation), to enable basestation multiuser/massive MIMO precoding. However, in many regulatory domains and also due to technical reasons, UL and DL are put on separate (yet nearby) frequency bands (frequency division duplex, FDD), thus putting a huge burden on the terminal to report DL CSI to the basestation through the UL channel.

Since, barring transceiver impairments, both UL and DL CSI are determined by the physical environment surrounding transmitter and receiver, it stands to reason that, for a static environment, a mapping from UL CSI to DL CSI may exist. The objective of this task is to investigate whether this mapping can be learned using various neural networks with different architectures. Fig. 14 illustrates this scenario in the frequency domain.

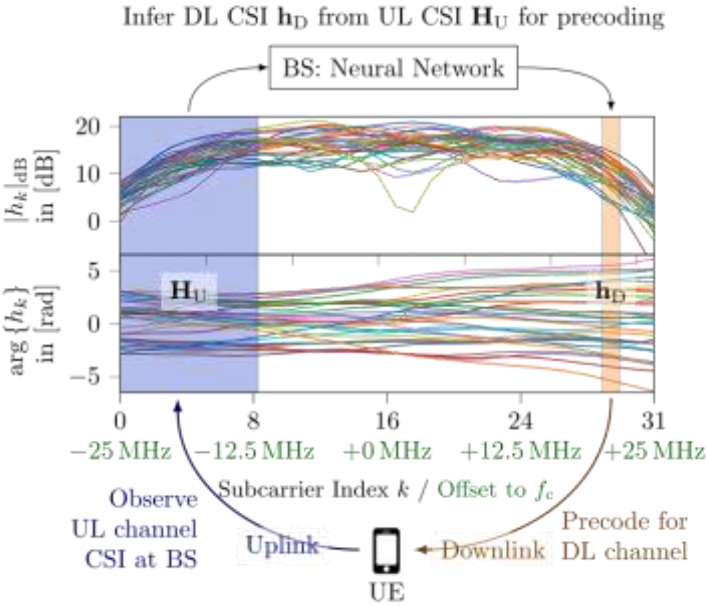

Figure 14: Principle of Operation: DL CSI is inferred from observed UL CSI on a different (but adjacent) frequency band.

While it has been conjectured that downlink CSI estimation from uplink CSI may be possible [14], to the best of our knowledge, the idea has never been practically verified on real-world measurement data. In particular, two open questions are the choice of the neural network architecture and the ability of a neural network to generalize to previously unseen data. Two possible network architectures, illustrated in Fig. 15, are a simple deep neural network with fully connected layers and an Autoencoder-like Encoder/Decoder structure. The latter architecture is motivated by the idea that a sparser representation of CSI should exist, justified by the fact that CSI is determined by geometrical properties of the physical environment, such as the location and orientation of the transmitter.

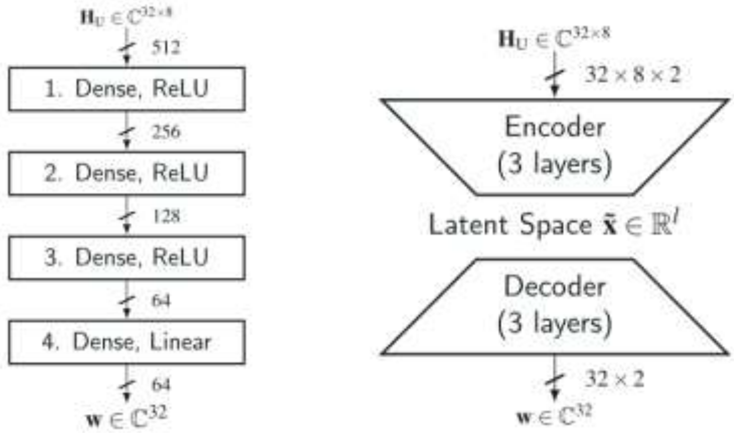

Figure 15: Different Neural Network Architectures for Downlink CSI Estimation (Dense Network or Encoder/Decoder Structure)

All experiments were carried out on an indoor dataset measured with DICHASUS [15]. Several baselines to benchmark neural network-based downlink CSI estimation against were defined. As a worst-case baseline, the precoding vector is chosen randomly, leading to average received powers that are 15dB below the optimum in our case. Secondly, a baseline using a single optimal precoding vector for the whole dataset is defined, which generalizes well, but leads to an average receive power that is 8dB below the optimum. All tested neural network-based downlink CSI estimation techniques significantly outperform these baselines, as shown in Fig. 16. To evaluate the quality of generalization, all datasets were split into training set and test set according to a checkerboard pattern. In Fig. 10, smaller circles represent a smaller checkerboard grid size.

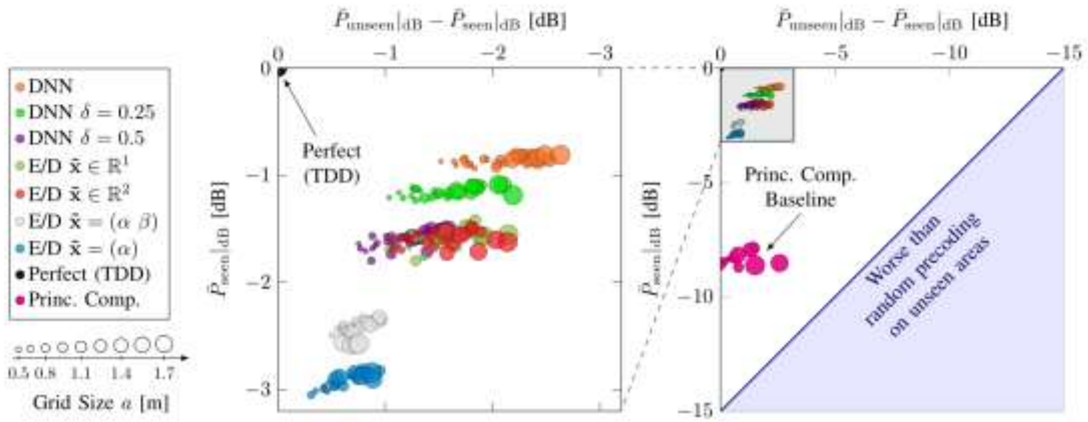

Figure 16: Performance of different neural network architectures. The horizontal axis indicates the quality of generalization of the precoding vector prediction to previously unseen areas of physical space, whereas the vertical axis shows the expected precoding performance on previously seen data. The size of the circle indicates the grid size of the checkerboard pattern (from 0.5m to 1.8m) that was used for evaluation.

A simple Dense Neural Network (DNN) exhibits the best performance on areas of space which are represented in the training set, but does not generalize well to areas that the network was not trained on. Adding dropout layers with dropout rate δ improves the quality of generalization somewhat. Autoencoder-like encoder / decoder (ED) structures perform worse on previously seen data, but generalize much better, especially if the latent space is constrained to represent either the azimuth angle α or both azimuth and elevation angles α and β. Regardless of the architecture, smaller grid sizes lead to better performance on previously unseen areas, i.e., an improved ability to generalize to unseen areas.

In summary, the results show that NN-based downlink channel estimation from available uplink CSI significantly outperforms the baselines and that generalization to physical areas not represented in the training set is one of the major challenges of the approach. Thanks DICHAUS datasets being publicly available, anyone may reproduce this research on the same dataset or compare the results to other datasets captured in different types of environments or with different antenna configurations, carrier frequencies and/or bandwidths.

2.4 Task 4.4: Advanced AI Machine Learning Algorithms for RF Detection and Location

Using Artificial Intelligence to detect and locate adversarial devices and attacks is a relatively news field in the Cyber Security and Defence domains with several Governments supporting local companies to develop a much needed capability.

Using advanced AI/Machine Learning and RF signal processing feature engineering techniques to design an automated RF interference detector to address both unintended EMI (electromagnetic interference) or nefarious transmissions (jammers) within 4G & 5G service bandwidths.

The AI based Interference Detection system required the creation of Machine Learning training data from both Real-world and RF/Channel Simulation sources in order to provide Training data for the detection algorithm. RF Data represented multiple 4G & 5G signal and transmission formats, multiple channel impairments as well as RF noise levels were used to be a robust machine Learning model.

Significant effort was applied to machine learning feature engineering (dimensionality reduction) methods which improved training processing time and accuracy of ML detection algorithms these included Segmented FFT windowing & re-sampling / Spectral density estimation (Welsh Method) with Envelope modelling and normalization. In addition, AI hyper parameters were tuned, tested, and optimized for the application.

As a result of a successful AI algorithm running at 95% + accuracy across, we deployed the algorithm on an embedded processor to be deployed on the network Edge with the ThinkRF Realtime Spectrum Analyser. Fine tuning of the machine learning algorithm and feature engineering was required to improve performance specifications.

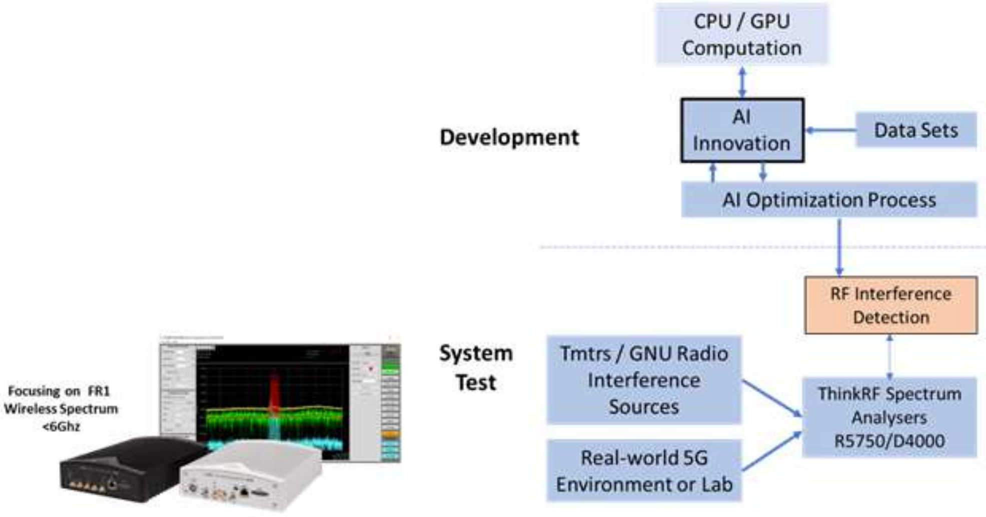

***Figure 17: Interference Detection Development and Test Environment*.**

ThinkRF AI based Interference Detection Testbed
Detection of RF interference sources - unintended and nefarious attacks

- Data Collection Phase
  - Sources: Synthetics (MATLAB) and Real-world transmissions (Canada & Europe if possible)
  - Interference sources – MATLAB Toolkits & test gear
- AI Innovation Phase
  - Optimization AI (anomaly Detection) algorithms / Parameters

We have developed unique knowledge and SW in the AI based interference detection algorithms that allows us to detection low power interference sources with minimal edge-wise processing power for 4G & 5G signals. A block diagram overview of the system is given in Fig. 11. The AI based algorithm methods provide us with the ability to extend this solution to other wireless services with minimal effort (ie Training). The solution detects jammer transmissions, harmonic and spurious noise sources as well as co-channel and adjacent channel interference even when the interference power levels are significantly less than the intended 5G signals and thus addressing a significant issue in the 4G/5G marketplace.

Technical Overview

ThinkRF's research on AI based Interference Detection capabilities with detection high-accuracy across a number of Interference sources including Narrow band (jamming), adjacent Channel and Co-Channel interference. The initial Machine Learning design was trained using Simulated MatLab 4G/5G signal samples that comprised of clean signal as well as samples with Physical RF channel impairments of various degrees. Later stages of testing included real-world 4G/5G signals from local Base-stations transmitters with lab based injection of interference sources.

A significant amount of assessment in both feature engineering design and AI algorithm hyper parameter modification was required. In addition, the latter testing scenario require the integration of ThinkRF's real-time Spectrum Analyser, embedded processor (for real-time detection inference), live antenna for spectrum signal capture as well as Interference lab set-up.

This project is the first step in developing a commercial product offer. The ongoing project task aims at developing an edge AI-based solution to detect the presence of anomaly or interference in the 5G PHY downlink, with the minimum false positive and negative rates. To this end, we are exploring digital signal processing and machine learning (ML) tools to analyse 5G spectrum for robust interference detection in a completely unsupervised way. The objective behind employing unsupervised learning for interference detection is to detect previously unseen interference events without any prior knowledge about these.

The developed interference detection approach is initially trained and tested on synthetic data generated using MATLAB 5G/LTE toolboxes. These 5G waveforms are contaminated with background noise and two sources of interference at different SNR and SINR levels, namely co-channel and adjacent channel interferences. SNR and SINR stand for signal to interference ratio and signal to interference plus noise ratio, respectively.

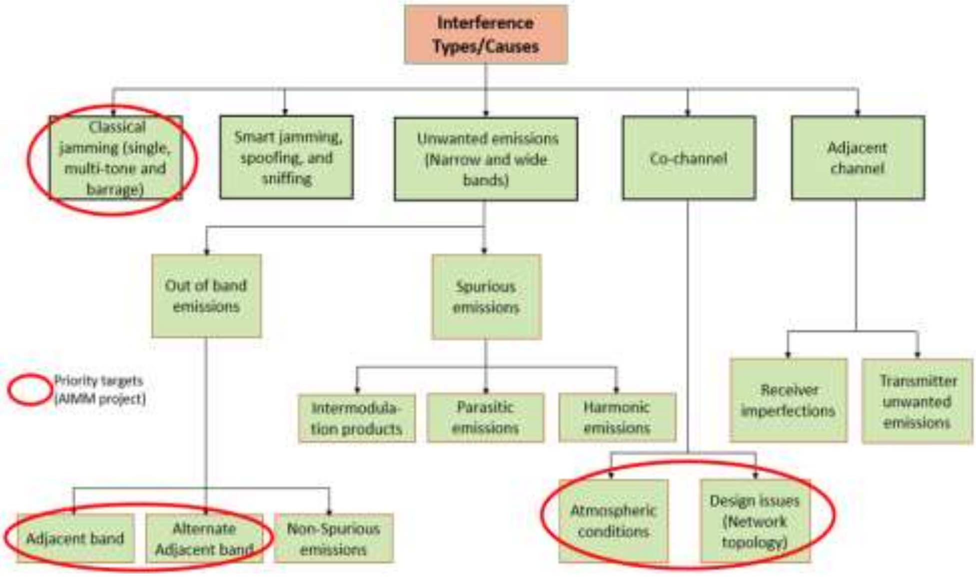

Fig. 18: Interference types and causes in wireless networks.

Overview of Interference in Wireless Networks

Interference is one of the most performance-limiting factors in wireless networks, which isoften used to refer to the addition of unwanted signals to a signal of interest. There are several determinants of interference; one can mention (i) the network geometry or problems related to spatial distribution of concurrently transmitting nodes, (ii) the path loss law or signal attenuation with distance, and (iii) equipment malfunctions. As shown in Figure 63, the most common interferences are due to unintentional adjacent or co-channel emissions, or any other sources of unwanted emissions. There are also so-called intentional interferences, also known as jamming, which are potentially threatening public safety.

Autoencoder-based anomaly detection

Autoencoders are a specific type of feedforward neural networks where the input is the same as the output. They comprise of the input into a lower-dimensional code and then reconstruct the output from this representation. The code is a compact “summary” or “compression” of the input, also called the latent-space representation. An autoencoder consists of 3 components: encoder, code and decoder. The encoder compresses the input and produces the code, the decoder then reconstructs the input only using this code. To build an autoencoder we need 3 things: an encoding method, decoding method, and a loss function to compare the output with the target.

The autoencoder’s dimensionality reduction technique can be applied to many problems include image denoising and anomaly detection. The latter is achieved by training the autoencoder in a “normal” condition environment. Once that is achieved, the trained autoencoder is now used to reconstruct the operation RF signal environment, where it is unable to do so, may indicate an RF anomaly is present. The anomaly detection process is generally based on the analysis of the MSE error (see Fig. 19).

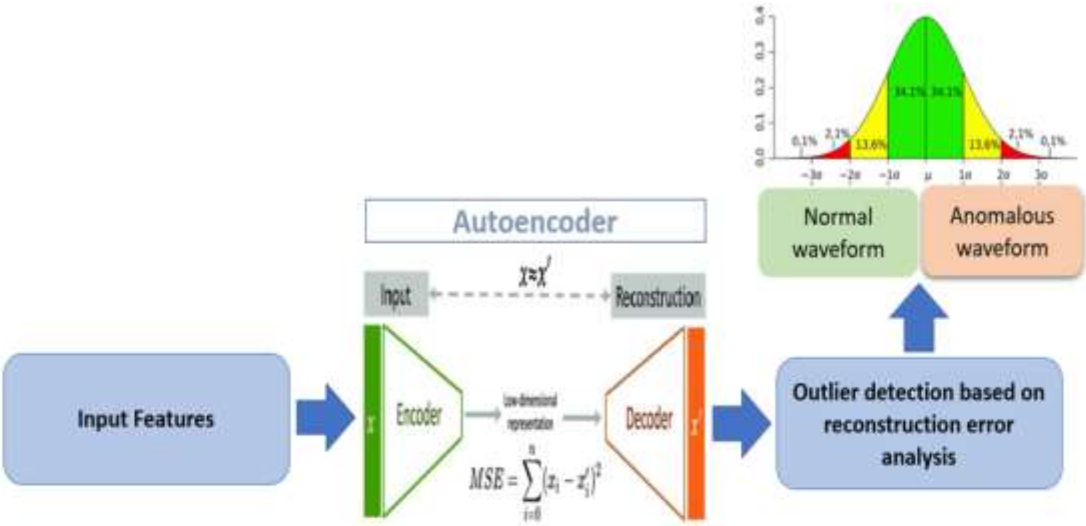

Figure 19: Autoencoder-based architecture for anomaly detection.

The ThinkRF testbed, shown in Figure 20, comprises of several components, including embedded processor (GPU) to run the advanced AI machine Learning algorithms and techniques to assess and evaluate Training parameters, ThinkRF Spectrum Analyzer with Omni-directional antenna. Network access and PC to control and monitor the set-up and provide remote access to users.

The testbed set-up will provide the ability to Detect and locate RF sources for interference from both unintended and nefarious sources. Although, in the short-term only interference detection is support as geolocation capabilities are currently under development.

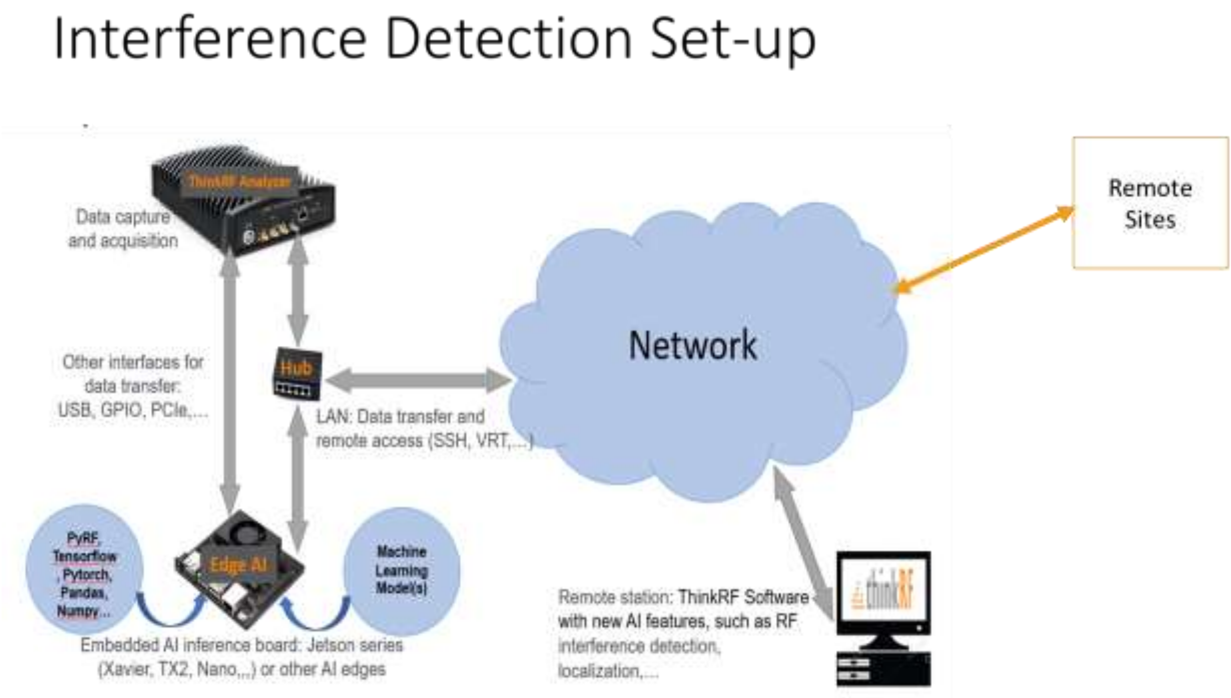

Figure 20: Wireless Security Testbed.

3 Conclusions and Future Work

WP4 has produced significant technical innovation in all tasks, be it channel estimation and MIMO detection (Task 4.1), user positioning and channel charting (Task 4.2), reduction of channel feedback in FDD massive MIMO (Task 4.3) and AI for RF detection and localisation (Task 4.4), and WP4 is on track with respect to its anticipated objectives. The WP4 participants have also been conducting testing, verification and fine-tuning of the developed AI-based algorithms using real-world data as measured in WP3 (distributed channel sounder, in particular for Tasks 4.2, 4.3) and WP6 (prototypes and testbeds, Tasks 4.1, 4.4).

The work in WP4 has been disseminated in a variety of ways including publication of channel data sets, patents applications, paper publications, and standards participation. The reader is referred to the AIMM dissemination report in D1.6 for further details.

References

[1] T. L. Marzetta, “Noncooperative cellular wireless with unlimited numbers of base station antennas,” IEEE Transactions on Wireless Communications, vol. 9, no. 11, p. 3590–3600, 2010.

[2] P. v. Butovitsch, D. Astely, C. Friberg, A. Furuskär, B. Göransson, B. Hogan and J. L. E. Karlsson, “Advanced antenna systems for 5G networks,” Ericsson White paper, 2019.

[3] A. Shojaeifard (InterDigital), “AI for Massive MIMO”, Cambridge Wireless Event, 2020.

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

[5] Yejian Chen; Jafar Mohammadi; Stefan Wesemann; Thorsten Wild; “Turbo-AI: Iterative Machine Learning Based Channel Estimation for 2D Massive Arrays,” Preprint arXiv: 2011.03521 [eess.SP], https://arxiv.org/abs/2011.03521, Nov. 2020.

[6] 3GPP Further Enhancements on MIMO for NR, WID, RP-202024.

[7] L. Cariou, “802.11 EHT Proposed PAR,” IEEE 802.11-18/1231r6.

[8] 3GPP 5G NR, Multiplexing and Channel Coding, TS 38.212.

[9] Y. Zhang, J. Zhang and A. A. Rawi, "Evolutionary Random Walk Aided Stochastic Sphere Encoder for Broadband G.mgfast," *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 1894-1899.

[10] C. Studer, S. Medjkouh, E. Gonultas¸, T. Goldstein, und O. Tirkkonen, ,,Channel charting: Locating users within the radio environment using channel state information,” IEEE Access, Vol. 6, S. 47 682–47 698, 2018.

[11] Chen, Y.; Mohammadi, J.; Wesemann, S.; Wild, T.; “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.

[12] Chen, Y.; Mohammadi, J.; Wesemann, S.; Wild, T.; “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.

[13] Chen, Y.; Mohammadi, J.; Wesemann, S.; Wild, T.; Turbo AI, Part IV: Estimating Uplink Channels for Ultra High Mobility With Sparse Pilots,” in *Proc. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC’22)*, Sep. 2022.

[14] Yang, Y., Gao, F., Li, G. Y., & Jian, M. (2019). Deep learning-based downlink channel prediction for FDD massive MIMO system. IEEE Communications Letters.

[15] F. Euchner, M. Gauger, S. Doerner, S. ten Brink, "A Distributed Massive MIMO Channel Sounder for "big CSI data"- driven Machine Learning", ITG/IEEE Workshop on Smart Antennas, Nov. 2021.

BT

CEMWorks

IMST

interdigital

Loughborough University

NOKIA

thinkRF

University of BRISTOL

Universität Stuttgart

Vilicom
signalling the future

Bundesministerium für Wirtschaft und Energie

UKRI Innovate UK

Canada NRC·CNRC