UNIVERSITY OF CAMBRIDGE

Applications of Artificial Intelligence to Electromagnetism and Indoor Wireless Networks

Stefanos Sotirios Bakirtzis

Homerton College

May 2024

This thesis is submitted for the degree of Doctor of Philosophy

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the preface and specified in the text. It is not substantially the same as any work that has already been submitted, or, is being concurrently submitted, for any degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Stefanos Sotirios Bakirtzis
May 2024

Abstract

Fifth-generation (5G) and beyond (B5G) mobile networks are anticipated to revolutionise the structure of the wireless ecosystem and support a set of highly heterogeneous applications, handing a substantially larger volume of mobile traffic with diverse requirements in terms of bandwidth, throughput, latency, quality of service and reliability. Remarkably, although the preponderance of this traffic is generated in indoor environments, the largest part of the legacy communication systems radio access network is deployed outdoors. As a result, next-generation networks are anticipated to align with the emerging need to serve indoor users, and therefore, major vendors and mobile network operators expect indoor cellular networks (ICNs) to be extensively deployed in 5G/B5G systems.

Consequently, the installation of new components within the heart of the radio access network, in conjunction with the increasing network complexity and user requirements, calls for expedient electromagnetic wave propagation modelling tools and optimisation frameworks that can assist the design of ICNs, ensuring the orchestrated and prosperous operation of the wireless ecosystem. Furthermore, to enable their optimal functionality it is necessary to comprehend the characteristics of the mobile service demands generated by ICNs.

Aiming at tackling these open problems, this thesis probes how artificial intelligence (AI) can be leveraged in the context of computational electromagnetics and indoor wireless networking. To this end, first, an innovative solution to Maxwell's equations through graph neural network (GNN) message passing is introduced by establishing a causal connection between GNNs and the finite-difference time-domain method. Consequently, an expedient and credible data-driven radio propagation modelling tool, named EM DeepRay, is developed by coupling a high-performance ray-tracer with a convolutional encoder-decoder. Then, the computational efficiency and robustness of EM DeepRay are exploited to develop an AI-assisted indoor wireless network planning framework, and to assess the uncertainty in network's performance. Finally, a characterisation of the mobile service usage across a nationwide ICN deployment is provided, indicating that ICNs inherently manifest a limited set of mobile application utilisation profiles, which are not present in conventional outdoor macro base stations.

Εἰς τὸ ῥεῦμα τῆς ζωῆς μου

Acknowledgements

I would like to express my deepest gratitude towards my supervisor, Dr. Ian Wassell, for his valuable guidance throughout my PhD. He provided me with the opportunity to join an exciting project and implement my research ideas while sharing his technical expertise and insightful feedback to enhance the quality of my work. In addition, his kindness and willingness to endorse and promote any of my endeavours have been remarkable, and I am grateful for his contribution to my growth as an individual and an academic. I would also like to sincerely thank my co-supervisor, Professor Jie Zhang, for his constant support, for sharing his knowledge and ideas with me, and for offering me the chance to work under his guidance at Ranplan Wireless. His mentality and strive for excellence helped me develop a strong work ethic and sense of responsibility. Additionally, I am highly grateful to Professor Marco Fiore for his multiple technical insights, active mentoring, genuine cordiality, and never-ending eagerness to help. It was a great honour and privilege for me to work with all these great people and researchers.

I would also like to thank the members of my thesis committee, Professor Miguel Rodrigues, and Dr. Michael Crisp, for their time and their valuable feedback, which has been very helpful in improving my thesis.

Kudos to all the BANYAN ESRs, Kehai Qiu, Andre Felipe Zanella, Akem Aristide, and Gabriel Ferreira, for our work during the project; I am very happy for the moments we shared, the conversations and jokes we had, along with the ideas we exchanged. Furthermore, I would like to thank my colleagues from Ranplan Wireless for their help in the development of DeepRay, all the members of the Network Data Science Group at the IMDEA Networks, and the SENSE Group at Orange Innovation for our work on indoor cellular network traffic analysis. It was a pleasure for me to meet them and work with them.

I would also like to acknowledge the financial support of the European Commission through the Horizon 2020 Framework Program for the first three years of my research, and the support of the Onassis Foundation and the Foundation for Education and European Culture during the last year of my studies.

Finally, quoting the same passage I wrote for my Master's thesis, I would like to thank all these people (no need to be named) with whom our lives are bound in a peculiar, yet beautiful way. I acknowledge this bond and unconditional love as the most important thing

in my life. In the years to come, I can only hope that this bond will be preserved, as each one of us will be walking their own unique path in this life.

Νὰ μὲ θυμόσαστε - εἶπε.
Χιλιάδες χιλιόμετρα περπάτησα χωρὶς ψωμί, χωρίς νερό, πάνω σὲ πέτρες κι ἀγκάθια,
γιὰ νὰ σᾶς φέρω ψωμὶ καὶ νερὸ καὶ τριαντάφυλλα.
Τὴν ὀμορφιὰ
Ποτές μου δὲν τὴν πρόδωσα ...

Γιάννης Ρίτσος, "Ἐπιλογικό"

Stefanos Bakirtzis,
University of Cambridge, 2024

Contents

1 Introduction **27**

1.1 Motivation 27

1.2 Contribution 29

1.3 Organisation 32

1.4 Publications 33

2 Preliminaries **35**

2.1 Next-Generation Wireless Networks 35

2.2 Radio Propagation Modelling 35

2.2.1 Empirical Propagation Models 36

2.2.2 Deterministic Propagation Models 37

2.2.2.1 Ray Tracing 37

2.2.2.2 The Finite-Difference Time-Domain method 38

2.3 Deep Learning Architectures 42

2.3.1 Multilayer Perceptrons 42

2.3.2 Convolutional Neural Networks 43

2.3.3 Graph Neural Networks 45

2.4 Explainable Machine Learning 46

2.4.1 The SHAP Framework 47

2.5 Summary 48

3 Solving Maxwell's Equations with Graph Neural Networks **49**

3.1 Expedient Solutions to Maxwell Equations 50

3.2 A Graph-Driven Solution to Maxwell Equations 51

3.2.1 Equivalent Graph Neural Network Representation of Discretised Maxwell's Equation 52

3.2.2 Graph-Driven Solution with Absorbing Boundaries 54

3.3 Numerical Results 55

3.4 Summary 59

4 Radio Propagation Modelling in the Era of Artificial Intelligence 61
4.1 EM DeepRay: An Expedient, Generalisable, and Realistic Data-Driven Indoor Propagation Model . . . 62
4.1.1 Input features and target output . . . 63
4.1.2 Workflow . . . 65
4.1.2.1 Convolutional encoder-decoder . . . 67
4.1.2.2 Evaluation metrics . . . 70
4.1.3 Comparison with simulated data . . . 70
4.2 Generalisation to 3D and Antenna Radiation Pattern . . . 75
4.2.1 Modification in EM DeepRay Architecture . . . 77
4.2.2 Augmented Data Set and Assessing Generalisation . . . 77
4.2.3 Evaluation with Measurements in Real World-Deployments . . . 81
4.3 Calibration via Transfer Learning . . . 82
4.4 Summary . . . 87

5 Applications of Data-Driven Propagation Models in Indoor Wireless Networks 89
5.1 Indoor Wireless Network Performance Metrics . . . 89
5.2 UDN Planning with Data-driven Propagation Models . . . 92
5.2.1 Optimisation Framework Overview . . . 93
5.2.2 Computational Intelligence Optimisers . . . 94
5.2.3 Evaluation in Practical Use Cases . . . 97
5.3 Stochastic Wireless Channel Evaluation . . . 105
5.3.1 Uncertainty Quantification . . . 105
5.3.2 Numerical Results . . . 107
5.4 Summary . . . 111

6 Characterising Indoor Cellular Network Mobile Service Demands 113
6.1 Internet Measurement Data . . . 114
6.2 Classifying ICN Behaviours . . . 115
6.2.1 Measuring relative ICN service usage . . . 115
6.2.2 Clustering ICN antennas . . . 117
6.3 Understanding ICN Traffic . . . 120
6.3.1 Interpreting Clustering Results . . . 121
6.3.1.1 Interpreting Clustering with SHAP . . . 121
6.3.2 Indoor Environment Analysis . . . 124
6.4 Comparison with Outdoor Antennas . . . 129
6.5 Temporal Analysis . . . 130
6.6 Summary . . . 134

7 Conclusion and Future Work **135**
7.1 Conclusion 135
7.2 Future Work 136

Bibliography **139**

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
ASDU-Net	Asymmetric Stacked Dilated Convolution U-Net
AU-Net	Asymmetric U-Net
B5G	Fifth-Generation and Beyond
BO	Bayesian Optimization
BS	Base Station
BTS	Base Transceiver Station
BWP	Building Wireless Performance
CDF	Cumulative Distribution Function
CEM	Computational Electromagnetics
CI	Confidence Intervals
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CQI	Channel Quality Indicator
DAS	Distributed Antenna System
DE	Differential Evolution
DL	Deep Learning
EPS	Evolved Packet System
fBS	femto Base Station
FCN	Fully Convolutional Networks
FDTD	Finite-Difference Time-Domain

FSPL	Free Space Path Loss
GA	Genetic Algorithm
GDPR	General Data Protection Regulation
GEM	Graph Neural Network-based Electromagnetics
GGSNs	Gateway GPRS Support Nodes
GNN	Graph Neural Network
GTP-C	GPRS Tunnelling Protocol control plane
GPU	Graphics Processing Unit
KPI	Key Performance Indicator
LoS	Line of Sights
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MEC	Mobile Edge Computing
MLP	Multilayer Perceptrons
MIMO	Multiple-Input Multiple-Output
MNO	Mobile Network Operator
MPNN	Message Passing Neural Network
MST	Multiscale Training
MWM	Multiwall Model
NR	New Radio
OMP	Orthogonal Matching Pursuit
ICN	Indoor Cellular Network
IWNP	Indoor Wireless Network Performance
RIS	Reconfigurable Intelligence Surfaces
QoE	Quality of Experience
QoI	Quantity of Interest
QoS	Quality of Service
PCE	Polynomial Chaos Expansion

PDN	Public Data Networks
PDP	Packet Data Protocol
PGW	Packet Data Network Gateway
PL	Path Loss
PML	Perfectly Matched Layer
PSO	Particle Swarm Optimization
RAN	Radio Access Network
RCA	Revealed Comparative Advantage
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Networks
RSCA	Revealed Symmetric Comparative Advantage
RSS	Received Signal Strength
Rx	Receiver
SAR	Specific Absorption Rate
SDU-Net	Stacked Dilated Convolution U-Net
SBR	Shoot and Bouncing Rays
SHAP	Shapley Additive Explanations
SINR	Signal to-Interference-Plus-Noise Ratio
TE	Transverse Electric
TM	Transverse Magnetic
TS	Technical Specifications
Tx	Transmitter
UDNs	Ultra-Dense Networks
ULI	User Location Information
UQ	Uncertainty Quantification
XML	Explainable Machine Learning

List of Tables

4.1 EM DeepRay performance for each case study; known and unknown indicates included or not included in the training data set. 74
4.2 Test set error metrics for different DL backbones. 79
4.3 Test set error metrics with different data augmentation techniques using ASDU-Net as a backbone. 80
4.4 Comparison with Measured Data. 81
4.5 Error of EM DeepRay compared to measurements before and after calibration. 86
4.6 Calibration results using a smaller number of random training points, and estimating the error with respect to the remaining measurement points. . . . 87

5.1 Mapping estimated SINR to CQI and spectral efficiency values. 92
5.2 Optimisation Algorithm Parameters. 96
5.3 Optimisation Performance Statistics. 99
5.4 Input sampling distributions. 107
5.5 Statistics of the LoS and nLoS point. 109
5.6 Statistics of the IWNP metrics. 111

List of Figures

2.1 Novel technologies employed in next-generation wireless networks. 36

2.2 Illustrative example of two ray-based deterministic propagation models showing (a) the rays generated using the SBR method, and (b) the images and ray trajectories generated through the image method. 38

2.3 grid layout of the TE polarisation electromagnetic field components using the FDTD method. 41

2.4 Multilayer perceptron. 43

2.5 Typical operations in a CNN. The orange box depicts the kernel of a convolutional filter slid over an RGB image. The different coloured squares represent the kernels of a max-pooling filter used to extract the maximum point from a sub-area of each channel. 44

3.1 Graph-driven solver of Maxwell's Equations; the E_y, H_z, and H_x components are shown in green, purple, and dark orange colour, respectively. **a**, Discretised TE Maxwell equations along with **b**, a corresponding example 2D TE grid representation of the electromagnetic field component arrangement in space. **c**, The grid implicitly assumes an equivalent graph topology, and consequently, **d**, the electromagnetic field update equations can be expressed in a graph-driven manner via mathematical operations between the graph nodes. 53

3.2 Computational emulation of electromagnetic wave propagation for three use cases (top, centre and bottom rows); **a** shows a snapshot of the electric field E_y evolution simulated via GEM, whilst **b** depicts the snapshot at the same time instance as computed with FDTD. The simulated domain is shown in **c**, whereas **d** illustrates the waveform of the E_y component sampled at a random point of the simulated grid. 56

3.3 Estimation of SAR with GEM for a biological tissue. **a**, Electric permittivity for the air, skin, fat, and muscle layers of the biological tissue. **b**, Snapshot of the electric field intensity dissipated throughout the tissue layers. **c**, Computed SAR with GEM and FDTD along a vertical cross-section passing through the point source. 57

3.4 Wave propagation in a cavity waveguide. **a** shows the geometry of the waveguide, while **b** depicts a snapshot of the electric field intensity across the waveguide. **c** Simulated waveform via GEM and FDTD sampled at the end of the waveguide. 58
3.5 Performance comparison. **a**, Simulation time for GEM, our in-house FDTD code and Meep. **b**, Performance gain achieved by GEM and Meep compared to the in-house FDTD code using CPU parallelization. 59

4.1 Pipeline of a data-driven propagation model. 62
4.2 Example of the input tensor channels for a random sample at an operating frequency of 2 GHz. 64
4.3 EM DeepRay architecture. 66
4.4 Convolutional encoder-decoder used in EM DeepRay to encode a physics-based input tensor and decode it as a PL heatmap. 68
4.5 Details of the cascaded atrous convolutions used in the SDU-Net block and an example of convolutional kernels with different atrous rates, r, applied to the image of an indoor environment. 69
4.6 Comparison between the PL heatmap simulated through (a) ray tracing and predicted by (b) EM DeepRay for a validation set sample at 433 MHz (known building and frequency, but unknown Tx position). The (c) error map depicts the absolute error between (a) and (b). 72
4.7 Comparison between the PL heatmap simulated through (a) ray tracing and predicted by (b) EM DeepRay for a Test set 1 sample at 2 GHz (known frequency band, but unknown indoor geometry). The error map (c) depicts the absolute error between (a) and (b). 73
4.8 Comparison between the PL heatmap simulated through (a) ray tracing and predicted by (b) EM DeepRay for a Test set 2 sample at 2.6 GHz (known indoor geometry, but unknown frequency band). The error map (c) depicts the absolute error between (a) and (b). 73
4.9 Comparison between the PL heatmap simulated through (a) ray tracing and predicted by (b) EM DeepRay for a Test set 3 sample at 868 MHz (unknown indoor geometry and frequency band). The error map (c) depicts the absolute error between (a) and (b). 73
4.10 Example of the angles between the Tx and a Rx. 75
4.11 Schematic of EM DeepRay, using an asymmetric SDU-Net with more encoding than decoding layers, where the fourth channel is modified to consider the impact of the antenna radiation pattern. 76

4.12 The Data augmentation process followed to increase the number of training samples and thus enhance the model's performance. The initial input and target tensors are flipped by $90^o, 180^o$, and 270°, while an MST strategy is followed focusing on a box area of 200, 400, 600, and 800 pixels around the Tx, and consequently flipping the zoomed samples to create additional samples. 78
4.13 Simulated via ray tracing and predicted by EM DeepRay PL heatmaps for samples with directive antennas. 79
4.14 Hospital heatmaps for ray tracing and EM DeepRay and measured RSS. . . 81
4.15 Workspace heatmaps for ray tracing and EM DeepRay and measured RSS. . 82
4.16 Calibration framework; the dots in the target tensor correspond to points in the indoor geometry at which the RSS is measured, and their colour represents the RSS intensity. 83
4.17 Predicted RSS, and comparison with measured data. 85

5.1 Complete framework for AI-assisted indoor wireless network planning with a data-driven propagation model, comprising (A) an expedient data-driven propagation model leveraged to readily emulate wireless signal propagation (green box), (B) evaluation of the metric describing the performance of the network (purple box), and (C) a $black - box$ computational intelligence optimiser that guides the optimisation space point selection by observing the harvested cost function for each point of the optimisation space, *i.e.*, possible deployment (orange box). Starting from a randomly selected deployment (left), the network performance is measured iteratively for deployments indicated by the computational intelligence algorithm, ultimately leading to a nearly-optimal network design (right) 94
5.2 Digital twin of an urban environment used to conduct an outdoor-to-indoor ray tracing simulation, providing the signal heatmaps of four macro BSs (black squares) within the digital replicas of three buildings, B1, B2 and B3 (black dots), where fBSs will be deployed. 97
5.3 Building layout and example of identified nearly-optimal network deployment for B1, along with the evolution of T_{eff} during the optimization with BO, indicating the cost function value achieved the at each iteration, as well as its best value up until that iteration, while using ray tracing and EM DeepRay to emulate signal propagation 100
5.4 Building layout and example of identified nearly-optimal network deployment for B2, along with the evolution of T_{eff} during the optimisation with BO. . . 101
5.5 Optimal site selection for B2 at the end of the optimization process, determined via (a) BO, (b) GA, (c) DE, and (d) PSO 102

5.6 Building layout and example of identified nearly-optimal network deployment for B3, along with the evolution of T_{eff} during the optimisation with BO. . . 103
5.7 Geometry under consideration, along with the simulated PL and throughput CDFs for a LoS and nLos point. 108
5.8 CDFs and CIs of the various IWNP metrics for the indoor geometry shown in Fig. 5.7a. 110

6.1 Distribution of the Normalized traffic, the RCA, and the RSCA for the M mobile services of a single antenna. 117
6.2 Silhouette score and Dunn index versus the number of clusters, serving as a stopping criterion to select the optimal number of clusters. 119
6.3 Dendrogram illustrating the iterative merging of antennas into clusters as returned by the hierarchical agglomerative algorithm run on RSCA features of individual antennas. Distance thresholds for $k_c = 6$ and $k_c = 9$ are highlighted. Colours tell apart the 9 clusters identified by the second threshold. 120
6.4 Heatmap of the mobile service RSCA (y-axis) for the clustered ICN antennas (x-axis), when $k = 9$. 121
6.5 SHAP values beeswarm plots for clusters 0 to 8, depicting the importance of each application in the clustering decision. The group colour refers to the branch colour of the dendrogram in Figure 6.3. Positive Shapley values matched with high feature values imply that for an antenna to belong to that cluster it should be overutilising the respective application, while low feature values with positive coefficients suggest under-utilisation. 123
6.6 Sankey diagram depicting how the clusters flow into different environment types. 126
6.7 Types of indoor environments per cluster 127
6.8 Cluster distribution per indoor environment type. 128
6.9 Distribution among the identified clusters, for 22,000 outdoor antennas located in close proximity of the ICN antennas included in the data set. 130
6.10 Normalised median traffic heatmaps per hour for a period between 04/01/2023 and 24/01/2023. Each heatmap represents the median traffic of all antennas that belong to the specified cluster at a specific hour and day, while the light grey dashed lines indicate weekends. 131
6.11 Heatmaps of per hour normalised median traffic between 04/01/2023 and 24/01/2023, for the antennas of each cluster and for a selected set of services chosen based on their importance according to the SHAP values presented in Figure 6.5. The light grey dashed lines indicate weekends. 132

Chapter 1

Introduction

1.1 Motivation

During the past decades, wireless communication systems have experienced immense growth and they are now an indispensable part of our everyday life. With the advent of the Internet of Things (IoT), and the establishment of fifth-generation (5G) and beyond systems (B5G), the landscape in the wireless ecosystem is expected to change radically [1, 2, 3]. Unlike legacy communication systems, next-generation wireless networks are expected to host a vast number of technologies and applications with diverse requirements. For example, 5G/B5G mobile networks will see the use of massive multiple-input multiple-output (MIMO) antennas, millimetre-wave (mm-wave) frequencies, reconfigurable intelligent surfaces (RISs), mobile edge computing (MEC), and ultra-dense networks (UDNs) [2]. Hence, efficient network planning and optimisation are becoming a much more convoluted and manifold problem, rendering conventional communication network design techniques obsolete [4].

Artificial intelligence (AI) is a key enabler in the reshaping of wireless networks and the shift from a reactive to a proactive network design approach [5]. In particular, the potential of machine learning (ML) algorithms, and especially deep learning (DL), to process large volumes of data and extract purposeful information, can revolutionise wireless network operation [6]. Until now, ML has been employed to tackle a wide range of wireless network design-related problems, such as resource allocation, user mobility analysis, localisation, and wireless channel modelling [7, 8, 9]. The latter case has recently attracted significant interest, giving rise to the novel category of ML-based or data-driven propagation models [10, 11] (note that in this work the terms ML-based propagation model and data-driven propagation model are used interchangeably), since radio propagation simulators are an indispensable tool for telecommunication operators and radio frequency engineers whilst designing new or expanding existing wireless communication systems.

In addition to fusing real-world-level accuracy to existing propagation simulators, the distinct advantage of ML-based propagation models is that once trained offline, they can

be used as standalone propagation models that replicate the results of complex propagation solvers with dramatically reduced computational time. This renders these models appealing for applications where a large number of radio propagation simulations is required; searching for efficient network topologies, *e.g.*, in UDNs, is a problem that precisely falls in this category, *i.e.*, it needs many iterations over different radio propagation estimations. Consequently, propagation solvers have a substantial impact on the computational cost and the search space process, as they are invoked a large number of times by iterative optimization algorithms in order to gradually adapt the network topology towards meeting some target key performance indicators (KPIs). This is the dominant reason why legacy computationally expensive solvers, *e.g.*, solvers based on ray tracing [12], that take minutes or even hours for a single simulation [13, 14] cannot support thorough explorations of the deployment space.

This problem can be severe to the point that network planning with conventional radio propagation solvers becomes computationally unfeasible under common optimization techniques such as genetic, particle swarm, or evolutionary algorithms, even in relatively small-scale scenarios with a limited number of wireless cells. This could slow the widespread deployment of indoor cellular networks (ICNs) which will be an integral component of UDNs owing to their potential to improve the coverage, spectral, and energy efficiency of 5G/B5G systems. Such improvements can be achieved by deploying a large number of small cells in target indoor environments where the signal coverage from outdoor base stations (BSs) deteriorates significantly [15]. These are highly relevant scenarios in 5G/B5G systems, and therefore several mobile network operators (MNOs) are preparing to migrate from the conventional "outside-in" coverage approach, *i.e.*, having outdoor BSs providing coverage to indoor users, to the establishment of ICNs and other indoor wireless network technologies [16, 17, 18, 19].

Furthermore, the emergence of such pervasive networks makes it important to comprehend how they will be used, since the dynamics and distinguishing features of the traffic data generated by in-building radio access network components have not yet been studied. Unlike outdoor BSs that tend to observe a diverse set of uses, *i.e.*, concurrently serving numerous subscribers engaging in a wide range of activities during the course of daily life, ICNs are expected to target more specific use cases. For instance, ICNs are deployed in underground subway and train stations to compensate for the limited coverage of the outdoor wireless network. Likewise, corporate offices are equipped with indoor BSs to provide enhanced and reliable communications to support the work of their employees [20], whilst their installation is necessary in exhibition centres and stadiums in order to accommodate the concentrated high-traffic demands intertwined with social events. Therefore, it is reasonable to expect that ICN traffic is influenced by the context in which it is generated, which is highly dependent on the indoor environment type and, by extension, on the kind of activities in which users are engaged. Consequently, the traffic dynamics of ICNs are expected to differ significantly from those of legacy communication system outdoor BSs.

1.2 Contribution

Aiming at tackling the open problems mentioned in Section 1.1, the following contributions stem from the work presented in this thesis:

- A full-wave transient solution to Maxwell equations is proposed via a proof-of-concept graph neural network (GNN)-based computational electromagnetics (CEM) framework, referred to as GEM. Specifically, it is demonstrated that the update equations derived by discretising Maxwell's differential equations can be natively expressed as a two-layer graph neural network with non-trainable pre-determined edge weights. The proposed framework solves Maxwell's equations numerically and exhibits precisely the same accuracy as the finite-difference time-domain (FDTD) method, *i.e.*, it is second-order accurate, yet, building on the native GPU parallelisation of graph neural networks it can dramatically reduce computational time of conventional FDTD implementations by a factor of up to 40 times. The immense gain in terms of computational time ultimately allows the simulation of arbitrarily large and complex problems related to electromagnetism. More importantly, it is anticipated that applying a similar treatment to systems of differential equations arising in other scientific disciplines. For example, computational fluid dynamics, neuroscience, and quantum physics can benefit from this approach, enabling the emulation of physical phenomena that are currently not feasible to do via conventional computational methods.

- While urban and sub-urban ML-propagation models have been widely explored [21, 22, 23, 24, 25, 26, 27], ML-based indoor propagation models have received less attention, despite their importance for 5G/B5G systems. Due to the pervasive deployment of indoor wireless networks in 5G/B5G systems, it's imperative to build data-driven models that consider the particularities of wave propagation in indoor environments. For instance, in indoor environments, it is common to encounter a larger variety of building materials than in outdoor spaces. Moreover, in outdoor ray tracing simulations, the building walls are modelled as semi-infinite spaces because of the high losses. However, the walls found in indoor environments are typically thinner, and for materials such as plasterboard or wood, the losses are considerably smaller. Hence, it is necessary to consider wave attenuation when propagating through walls. Aspiring to close this gap, this thesis overcomes two fundamental limitations of indoor ML-based propagation models: (*i*) how to forge generalisable data-driven propagation models that can effectively make accurate predictions for unknown simulation scenarios (materials, building layout, antenna position, and radiation pattern), (*ii*) how the fidelity of such models can be enhanced by also considering actual measured data, instead of using simulated data only.

- To this end, first it is outlined how a convolutional encoder-decoder can be trained to predict the path loss (PL) for an arbitrarily complex indoor environment over different frequency bands, and antenna radiation patterns. In particular, a modified version of the U-Net [28] architecture using stacked dilated convolutions is employed (SDU-Net) [29], which learns to transform an input tensor comprising information related to the indoor simulation scenario to a PL heatmap. Unlike [30, 31, 32], the predictions of the proposed data-driven framework, named EM DeepRay [33], rely on physics-based information, such as the permittivity and the conductivity of the walls, and other parameters that affect wave propagation, *e.g.*, the distance and the frequency. It is shown that such treatment enables the problem of limited generalizability to be overcome, as it is directly applicable to multiple indoor geometries, frequencies, and antenna radiation patterns without any further training. Once trained, EM DeepRay can be used as a fully-fledged propagation solver to predict the PL for unknown simulation scenarios, accurately mimicking the results of a ray-tracer with the additional advantage of a substantially reduced computational time (less than a second in a graphics processing unit (GPU)).

- EM DeepRay is designed to replicate the results of a ray-tracer. Yet, ray tracing results intrinsically exhibit errors with respect to the real channel behaviour due to the finite accuracy of the propagation tool, e.g., approximation of electromagnetic wave with rays, the way that the digital environment is simulated (wall thickness, material properties), and details that cannot be captured in the simulation (small objects and human motion). Consequently, the predictions of EM DeepRay, as a surrogate of ray tracing, will also be subject to such errors. To overcome this limitation this thesis presents an approach that allows the data-driven propagation model to make predictions that match measured data. Specifically, transfer learning is employed to calibrate EM DeepRay, by adjusting its pre-estimated weights computed using simulated data, and provide realistic estimates of the signal level using only a small quantity of measured data.

- To highlight the importance of data-driven propagation models, this work explores their applicability in tasks related to the wireless ecosystem. Specifically, two applications of EM DeepRay are considered: (*i*) UDN planning and (*ii*) uncertainty quantification (UQ) [34] of the wireless channel and indoor wireless network performance (IWNP). In the first case, it is shown how data-driven propagation models can be leveraged for the automated and expedient deployment of small cells in indoor environments, introducing a new class of indoor wireless network planners that combine generalisable data-driven propagation models with AI-based optimisers to determine optimal network topologies with respect to a target KPI. One specific instance of the proposed approach is implemented, proving that the use of data-driven propagation models can reduce

the computational time of wireless indoor network design by two to three orders of magnitude. Thus, accurate planning is enabled, which would otherwise be extremely expensive to conduct using conventional indoor propagation tools, yielding significant gains in the resulting indoor planning quality and performance. In the second case, we explore how the uncertainties related to the simulated environment affect the estimates of the wireless channel characteristics and the IWNP evaluation. This uncertainty can be ascribed to the lack of knowledge regarding the geometry itself or regarding the properties, *i.e.*, the permittivity and the conductivity, of the construction materials that are found within the geometry [35]. Again, the UQ analysis with EM DeepRay and the ray-tracer yields similar results, but in a substantially reduced computational time.

- Unlike previous research that focused on understanding and predicting the dynamics of the macro antenna traffic [36, 37, 38, 39, 40], this thesis provides the first comprehensive study on the characteristics of ICN traffic, specifically considering its intrinsic particularities. From the analysis of a countrywide dataset of measured ICN Internet traffic data, an appropriate transformation of the traffic data is defined, enabling the range of different Internet mobile service utilisation profiles to be probed at indoor antennas. Then, employing an unsupervised learning approach it is designated that distinct service utilisation clusters are inherent in indoor communication systems. This rich and diverse behaviour of Internet services has not been unveiled before, and it is also demonstrated that it does not align with that of outdoor legacy communication systems. To interpret the clustering results and delve into the essence of the different clusters, techniques from the field of explainable machine learning (XML) are leveraged. This enables the identification of the most important features for each cluster, and consequently allows the most important, as well as the least used, Internet service types at each cluster to be established.

- It is exposed that there is a strong connection between the ICN clusters identified by the aforementioned analysis and the indoor environment type. In particular, it is shown that the same Internet applications manifest heterogeneous behaviours, even for antennas in close proximity, due to the determining influence of the environment type on user activities. This phenomenon is rooted in the ICN environment and has not been highlighted before. Finally, it is revealed that in the various clusters the total Internet traffic data, as well as the traffic generated by the individual applications, exhibit different activity peaks and temporal patterns. This paves the way for the proactive and successful management of ICN traffic by MNOs.

1.3 Organisation

The outline of the thesis is as follows:

Chapter 2 provides a brief overview of the wireless ecosystem structure, and it recites the categories of the existing radio propagation models. Furthermore, it covers the functionality of some basic artificial neural network (ANN) architectures, such as multilayer perceptrons (MLPs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and GNNs. Finally, Chapter 2 discusses existing XML techniques that enable interpreting the decisions of black-box ML frameworks. In particular, the Shapley additive explanations (SHAP) framework [41] is presented, which attempts to explain the response of an ML model through a set of linear polynomial functions with binary random variables, and in the context of this thesis, it is used to interpret the results of the clustering of ICN antennas.

Chapter 3 expounds on how Maxwell's equations can be natively solved with GNN message passing. These equations are of paramount importance as they depict the laws that govern electromagnetic wave propagation and represent the foundations of modern communication and electrical technologies. Hence efficient solutions to Maxwell's equations can streamline various applications in diverse scientific fields. It is shown that a network of neurons with specific connections and weights exists, for which the message passing between its nodes is equivalent to the solution of the discretised Maxwell equations. Consequently, the exchange of messages between the network neurons resembles the spatiotemporal evolution of the electromagnetic fields. Unlike, data-driven CEM approaches, which are ill-fated in light of the fast error accumulation in their predictions, GEM yields no error due to the proper setting of the connections between the graph nodes and the corresponding edge weights. The minimal computational complexity entailed by the properly tuned message-passing neural networks along with their native GPU implementation leads to a substantial acceleration over the standard FDTD technique using central processing unit (CPU) parallelisation, while also outperforming previous state-of-the-art transient response solvers.

Chapter 4 introduces the proposed data-driven indoor propagation modelling framework, detailing the structure of the input and the output tensor and that of the selected DL architecture used to implement EM DeepRay. Additionally, Chapter 3 displays numerical results comparing the predictions of EM DeepRay with synthetic data generated through a ray tracing simulator, studying scenarios where the proposed framework is trained over multiple geometries and frequency bands, and its generalisability is explored for new indoor environments, frequencies, and antenna radiation patterns. Finally, it is outlined how transfer learning can be exploited to calibrate EM DeepRay; then, the accuracy of its predictions with respect to measured data is benchmarked against that of other commonly used propagation models.

Chapter 5 discusses how data-driven propagation models can be leveraged for wireless network-related applications, eventually replacing conventional propagation solvers. Specifi-

cally, by coupling EM DeepRay with AI optimizers to test multiple network topologies and eventually identifying the one that maximizes the IWNP, it is shown that such models can be exploited for the rigorous and fully automated AI-based indoor wireless network planning according to a KPI of interest, The analysis highlights how the once cumbersome procedure of finding the optimal network topology now takes only a few minutes. More importantly, it paves the way for the shift from conventional and obsolete network planning techniques, that rely on human experience or on simplistic mathematical models, to a fully automated AI-powered network design.

Chapter 6 unveils the unique behaviours that are intrinsic to the traffic generated by ICNs, demonstrating that a set of clusters, with distinct mobile service utilisation profiles, exists across a nationwide ICN deployment. Through XML, a detailed analysis of the services that characterise particular profiles is provided, and consequently, it is shown that ICN traffic demands strongly relate to the type of indoor environment. These profiles are more diverse compared to those encountered in the traffic generated by conventional outdoor BSs. In addition, they exhibit distinctive overall and per-application utilisation temporal patterns, influenced by the kind of activities that take place in the different indoor environments.

Finally, Chapter 7 concludes this thesis, discussing its contributions and some open remaining challenges.

1.4 Publications

The contributions summarised in Section 1.3 have led to following publications:

- Stefanos Bakirtzis, Jiming Chen, Kehai Qiu, Jie Zhang, Ian Wassell, "EM DeepRay: An Expedient, Generalizable and Realistic Data-Driven Indoor Propagation Model", IEEE Transactions on Antennas and Propagation, Special Issue on Artificial Intelligence in Radio Propagation for Communications, vol. 70, no. 6 [33]

- Stefanos Bakirtzis, André Zanella, Stefania Rubrichi, Cezary Ziemlicki, Zbigniew Smoreda, Ian Wassell, Jie Zhang, Marco Fiore, "Characterizing Mobile Service Demands at Indoor Cellular Networks", ACM Internet Measurement Conference 2023, Montreal, Canada [42]

- Stefanos Bakirtzis, Ian Wassell, Marco Fiore, and Jie Zhang, "AI-assisted Indoor Wireless Network Planning with Data-Driven Propagation Models", IEEE Network [43]

- Stefanos Bakirtzis, Kehai Qiu, Jie Zhang, Ian Wassell, "Deep Learning-based Multivariate Time Series Classification for Indoor Outdoor Detection", IEEE Internet of Things Journal, vol.9, no.23, pp. 24529-24540 [44]

- Stefanos Bakirtzis, Kehai Qiu, Jie Zhang, Ian Wassell, "DeepRay: Deep Learning Meets Ray Tracing", 16th European Conference on Antennas and Propagation, Madrid, Spain, IEEE, March 2022, pp. 1-5 [32]

- Stefanos Bakirtzis, Ian Wassell, Marco Fiore, and Jie Zhang, "Stochastic Evaluation of Indoor Wireless Network Performance with Data-Driven Propagation Models", IEEE 2022 Globecom, Brazil, Rio de Janeiro, December 2022 [34]

- Stefanos Bakirtzis, Marco Fiore, and Ian Wassell, "Towards Graph Neural Network-based Computational Electromagnetics", IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting, Florence, Italy, 2024 [45]

- Stefanos Bakirtzis, Kehai Qiu, Jiming Chen, Hui Song, Ian Wassell, and Jie Zhang, "Rigorous Indoor Wireless Communication System Simulations with Deep Learning-based Propagation Models", IEEE Journal on Multiscale and Multiphysics computational techniques

The following manuscripts are under review:

- Stefanos Bakirtzis, Marco Fiore, Ian Wassell "Solving Maxwell Equations with Non-Trainable Graph Neural Network Message Passing", Journal of Computational Physics [46]

- Stefanos Bakirtzis, Çağkan Yapar, Marco Fiore, Jie Zhang, and Ian Wassell, "Empowering Wireless Network Applications with Deep Learning-based Radio Propagation Models", IEEE Wireless Communications [47]

Chapter 2

Preliminaries

2.1 Next-Generation Wireless Networks

The structure of the wireless ecosystem has been constantly evolving over the last decades. From the 2G networks, supporting voice calls and text messages there has been a giant leap to the 5G/B5G networks that are expected to provide an umbrella to IoT, machine-to-machine communications, virtual reality, and other emerging applications. Indeed, as shown in Fig. 2.1, next-generation wireless networks will deploy a plethora of new technologies to provide ubiquitous coverage, combined with increased data rates, reduced round trip latency, and higher energy efficiency. On one hand, 5G networks aim at interconnecting different structural services and entities of the modern world, *e.g.*, transportation systems or industrial IoT, and create a networked society [48]. On the other hand, B5G networks will have as a primary objective to embed intelligence in this interconnected environment and establish an autonomous self-orchestrated ecosystem [49]. For instance, the deployment of RISs will transform the radio propagation environment from a passive medium to an active entity that is reconfigured to assist communications, whilst MEC can be exploited for the smart and optimal handling of the information distributed through the network. Consequently, embedding AI within network functionalities, or leveraging it to process large volumes of data and guide the core network expansion, is indispensable for the successful orchestration and management of the evolutionary complex wireless ecosystem [4].

2.2 Radio Propagation Modelling

Radio propagation modelling is the cornerstone of the efficient and successful design of any wireless communication system. Indeed, having reliable estimates of the signal propagation in space enables proactive wireless network design, aiming at attaining the highest signal-to-interference-plus-noise ratios (SINR), which in turn, allows the use of the appropriate modulation and coding schemes that yield higher data rates and can provide an enhanced user

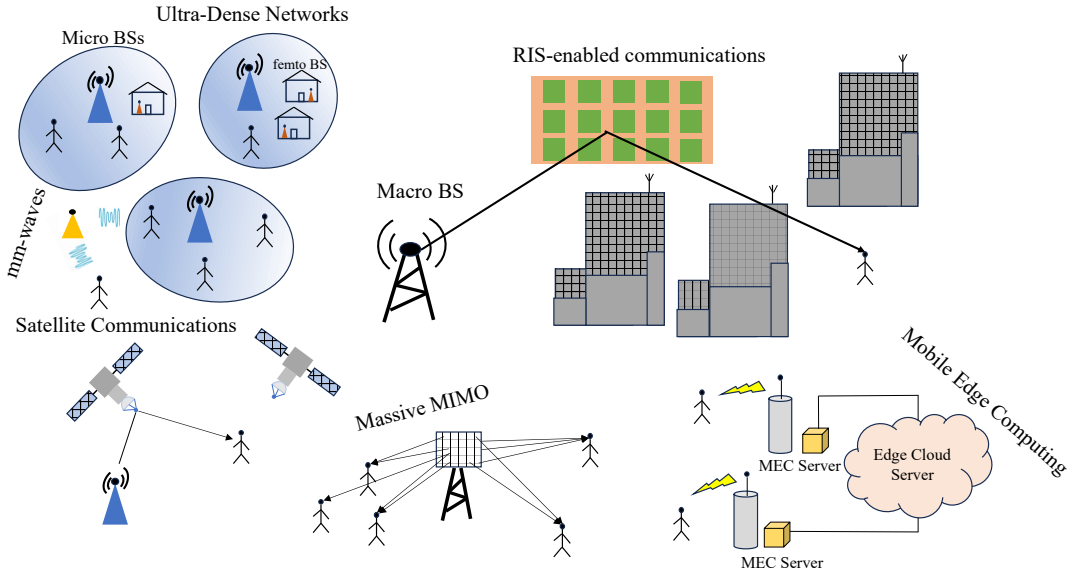


Figure 2.1: Novel technologies employed in next-generation wireless networks.

quality of service (QoS) and experience (QoE). However, although the nature of electromagnetic wave propagation can be captured by Maxwell's equations, due to the wireless channel complexity, deriving an analytical solution can be impossible, even for simple propagation environments. For that reason, conventionally, empirical or deterministic models are used to assess wireless channel characteristics [50].

2.2.1 Empirical Propagation Models

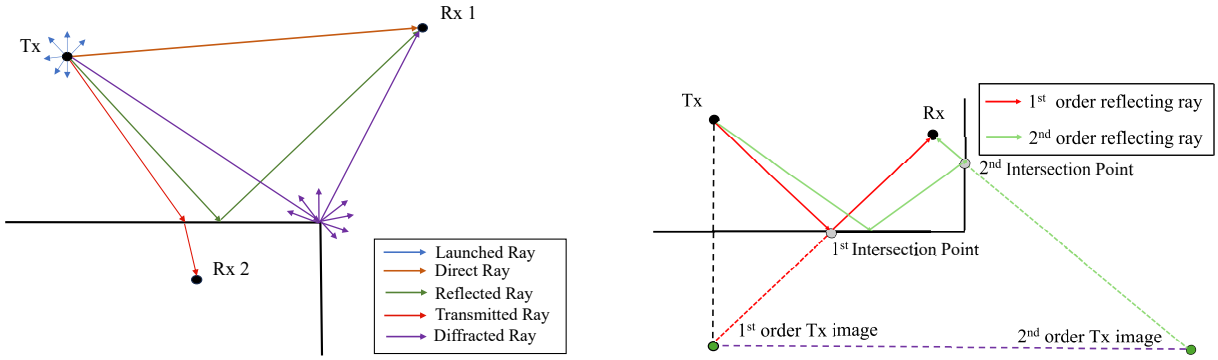
Empirical channel models are derived by fitting simplistic mathematical models to measurement campaign data, and they can be time dispersive or non-time dispersive [51, 52, 53]. Non-time dispersive models evaluate the mean attenuation of the propagating signal as a function of some communication scenario parameters, *e.g.*, frequency and BS height. Some commonly used non-time dispersive empirical propagation models for outdoor propagation environments include the Hata model, the COST-231 model, or the Walfisch-Ikegami model, while the multiwall model (MWM) is a popular tool used for indoor environments [54]. The time dispersive models describe the time-varying nature of the channel link capturing measures such as the multipath delay profile of the channel. The Stanford University Interim channel model and the ITU Path Loss Model constitute some indicative examples of empirical time dispersive models [55]. Overall, although empirical models are practical in use, they can demonstrate significant deviations from the actual channel parameters [53], potentially rendering them unreliable.

2.2.2 Deterministic Propagation Models

Deterministic propagation models furnish approximate solutions to Maxwell's equations by solving them under certain approximations. These models are site-specific, considering the distinct particularities of the propagation environment, such as the geometry and material electromagnetic properties. Typically, the stricter the approximation used by the model, the higher the accuracy, but also the higher the computational cost. Hence, although deterministic propagation models yield accurate estimates of the wireless channel characteristics, their computational complexity makes them unappealing for large-scale network deployment or real-time applications. In the following two subsections, an asymptotic and a full-wave deterministic propagation model are described. These models will be used later in the thesis to forge a data-driven radio propagation solver and pioneer a *graph-driven* CEM tool that can also be used in the context of radio propagation modelling.

2.2.2.1 Ray Tracing

Ray tracing has been widely adopted to conduct link-budget simulations, and due to its ability to simulate multi-path effects of time-varying channels, it is expected to have an even more important role in the design of 5G/B5G systems, UDNs, and in the installation of femto-cells in indoor environments [12, 56]. In ray tracing the electromagnetic waves are approximated as straight lines or tubes that traverse the propagation environment interacting with the objects found within their path. Under this assumption, the total received field at a point of the simulated space can be computed as the superposition of all the fields/rays that can reach that point. These can be rays that reach that point without obstruction, *i.e.*, direct line of sight (LoS) rays, or rays generated by the interaction (reflection, refraction, diffraction, or scattering) of a ray with an object of the propagation environment. The most common methods to trace the trajectories of the rays are the shoot and bouncing rays (SBR) and the image method [57, 58]. The first assumes that a finite number of rays are launched by a transmitter (Tx) and consequently the trajectory of each ray is traced forwardly, detecting the interactions of each ray with objects of the simulated space, from which subsequent reflected, transmitted, diffracted or scattered rays are generated, as shown in Fig. 2.2a. The latter uses the image theory to estimate the ray paths in a backward manner. Specifically, it's necessary to generate the images, *i.e.*, the geometrical projections, of the Tx on each facet of the simulated space. Consequently, for the straight lines connecting these images with the Rx, one must find the points at which they intersect with the respective facet. Then, the emerging line path between the Tx, the intersection points, and the Rx constitutes the reflecting ray trajectory. An example of this process is presented in Fig. 2.2b, depicting the two images of the Tx on the parallel and perpendicular facets. The lines connecting the images with the Rx are shown in red and green color, respectively (the line part up to the intersection point is dashed). The first-order ray trajectory is found by connecting the Tx to



(a) Rays generated using the SBR method.

(b) Arrangement of electromagnetic field components in FDTD grid.

Figure 2.2: Illustrative example of two ray-based deterministic propagation models showing (a) the rays generated using the SBR method, and (b) the images and ray trajectories generated through the image method.

the intersection point (gray dot) of the first image and then to the Rx. For the second order ray, *i.e.*, reflected by two facets, the intersection point of the first image is connected to that of the second image and then to the Rx.

2.2.2.2 The Finite-Difference Time-Domain method

Unlike ray tracing which treats the electromagnetic fields as rays, full-wave methods, such as the FDTD technique, can be used to derive a numerical solution of Maxwell's equations. The third and fourth equations, also known as the Ampere and Faraday laws, capture the impact of a time-varying electric field on the magnetic field and vice versa, specifically:

$$\nabla \times \mathbf{H} - \epsilon \frac{\partial \mathbf{E}}{\partial t} = \mathbf{J}, \tag{2.1}$$

$$\nabla \times \mathbf{E} + \mu \frac{\partial \mathbf{H}}{\partial t} = -\mathbf{M_f}, \tag{2.2}$$

where $\mathbf{E}$ and $\mathbf{H}$ represent the electric and magnetic field intensity vectors, respectively, whilst $\mathbf{J} = \sigma \mathbf{E}$ and $\mathbf{M_f} = \sigma^* \mathbf{H}$ denote the electric and magnetic current density vectors, and ϵ, μ, σ, and σ^* are the permittivity, the permeability, the electric conductivity and the magnetic conductivity of the medium, respectively.

Considering that $\mathbf{E}$ and $\mathbf{H}$ are three-dimensional (3D) field vectors, *i.e.*, $\mathbf{E} = [\ E_x(x,y,z), E_y(x,y,z), E_z(x,y,z)]$ and $\mathbf{H} = [\ H_x(x,y,z), H_y(x,y,z), H_z(x,y,z)]$, in the general case (2.1) and (2.2) will provide a system of six coupled partial differential equations relating the components of the magnetic and the electric fields. However, in many cases, a physical problem can be simplified and reduced into a two-dimensional planar version, if the modelled structure can be assumed to extend to infinity in one direction without exhibiting changes in the shape or position of its transverse cross-section, *e.g.*, propagation within a

waveguide or the normal incidence of a wave onto an interface between two media. The oblique incidence of a plane wave onto an interface (which is interwoven with radio propagation, where waves impinge onto walls and objects) can also be perceived as a 2D problem by expressing the incident wave as the summation of two waves with particular polarisations. Specifically, for the first wave the electric field is perpendicular to the plane of incidence, while for the second the magnetic field is perpendicular to the plane of incidence. These cases are referred to as transverse electric (TE) and transverse magnetic (TM), respectively.

For instance, assuming a plane wave propagating in the z-direction, then the TE polarisation will comprise the E_y, Hx, and Hz components, with the electric field E_y being perpendicular to the direction of propagation. The TM will include the E_x, E_z, and H_y components, with the magnetic field H_y being perpendicular to the direction of propagation. In these cases, (2.1) and (2.2) reduce to a set of three coupled partial differential equations for each polarisation. Indicatively, for non-magnetic material, *i.e.*, $M_f = 0$, the differential equations emerging from (2.1) and (2.2) for the TE polarisation are:

$$\sigma E_y + \epsilon \frac{\partial E_y}{\partial t} = \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x}, \tag{2.3}$$

$$\mu \frac{\partial H_x}{\partial t} = \frac{\partial E_y}{\partial z}, \tag{2.4}$$

$$\mu \frac{\partial H_z}{\partial t} = -\frac{\partial E_y}{\partial x}. \tag{2.5}$$

Despite the fidelity of Maxwell's equations, computing analytically the electromagnetic field components through them is impossible for most physical world problems, which motivated the development of CEM methods. To solve Maxwell's equations numerically the FDTD technique employs a centred finite-difference approximation scheme to evaluate the derivative terms, discretising the spatial and the temporal derivatives of a field component f_c around a point x_0 over an infinitesimal interval Δh as follows:

$$\partial f_c / \partial x \big|_{x=x_0} \approx \frac{f_c\left(x_0 + \frac{\Delta h}{2}\right) - f_c\left(x_0 - \frac{\Delta h}{2}\right)}{\Delta h}. \tag{2.6}$$

Applying this treatment in the equations of the TE polarisation case, (2.3) - (2.5), centred around the points $[x, z, t] = [i + \frac{1}{2}, j + \frac{1}{2}, n + \frac{1}{2}]$, yields the following update equations:

$$H^{n+\frac{1}{2}}_{x_{i+\frac{1}{2},k}} = H^{n-\frac{1}{2}}_{x_{i+\frac{1}{2},k}} + \frac{\Delta t}{\Delta z \cdot \mu_{i+\frac{1}{2},k}} \left(E^n_{y_{i+\frac{1}{2},k+\frac{1}{2}}} - E^n_{y_{i+\frac{1}{2},k-\frac{1}{2}}} \right), \tag{2.7}$$

$$H^{n+\frac{1}{2}}_{z_{i,k+\frac{1}{2}}} = H^{n-\frac{1}{2}}_{z_{i,k+\frac{1}{2}}} - \frac{\Delta t}{\Delta x \cdot \mu_{i,k+\frac{1}{2}}} \left(E^n_{y_{i+\frac{1}{2},k+\frac{1}{2}}} - E^n_{y_{i-\frac{1}{2},k+\frac{1}{2}}} \right), \tag{2.8}$$

$$E^{n+1}_{y_{i+\frac{1}{2},k+\frac{1}{2}}} = \frac{A_-}{A_+} E^n_{y_{i+\frac{1}{2},k+\frac{1}{2}}} + \frac{\Delta t}{A_+ \cdot \epsilon_{y_{i+\frac{1}{2},k+\frac{1}{2}}}} \left(\frac{H^{n+\frac{1}{2}}_{x_{i+\frac{1}{2},k+1}} - H^{n+\frac{1}{2}}_{x_{i+\frac{1}{2},k}}}{\Delta z} - \frac{H^{n+\frac{1}{2}}_{z_{i+1,k+\frac{1}{2}}} - H^{n+\frac{1}{2}}_{z_{i,k+\frac{1}{2}}}}{\Delta x} \right), \quad (2.9)$$

where $A_{-,+} = 1 \mp \frac{\sigma_{i+\frac{1}{2},k+\frac{1}{2}} \Delta t}{2\epsilon_{i+\frac{1}{2},k+\frac{1}{2}}}$, Δt is the time step, Δx and Δz are the spatial steps in the horizontal and vertical direction, respectively, and i, k and n are integers. Similar equations can be straightforwardly derived for the transverse magnetic (TM) polarisation or the 3D case. As indicated by (2.7) – (2.9) and shown in Fig 2.3, the field components are arranged into a mesh and are sampled at spatial locations which are offset by half a pixel, whilst, by convention, the electric field components are evaluated at time steps that are multiples of n and the magnetic field components at time step multiples of $(n + \frac{1}{2})$. Magnetic field values at time steps multiples of n can be found by interpolating their values between the $(n - \frac{1}{2})$ and $(n + \frac{1}{2})$ time step. Apparently, one could also discretise the derivatives around the point $[x, z, t] = [i + \frac{1}{2}, j + \frac{1}{2}, n]$, in which case the magnetic fields would be evaluated at time steps that are multiples of n and the electric field components at time step multiples of $(n + \frac{1}{2})$. That would not impose any conceptual difference to the method and the field at other time steps would again be possible to compute via interpolation.

As can be seen, the FDTD update equations (2.7) – (2.9) relate the electric fields at discrete points in space and time with the neighbouring magnetic fields, and vice-versa. A field at a given time instance is expressed as a summation between its weighted value at the previous iteration and a weighted difference of the neighbouring field values. In particular, for the TE polarisation, the FDTD update scheme entails first evaluating each magnetic field component using the neighbouring electric field node values at the current time step and the values of the respective magnetic field at the previous time step. Then, the computed values of H_x and H_z along with the values of E_y at the previous time step are employed to estimate the electric fields at the next iteration, which are consequently used to update again the magnetic fields and so on. Ultimately, the evolution of the electromagnetic fields over space and time is posed as a two-step iterative procedure.

This procedure will converge and result to an accurate emulation of the electromagnetic fields, as long as the selected spatial and time steps satisfy the Courant–Friedrichs–Lewy condition and numerical dispersion equation. A detailed discussion on these conditions can be found in Chapter 4 of [59]. Aside from stability and dispersion constraints, the solution of Maxwell's equations via (2.7) – (2.9) will introduce numerical artifacts, and in particular artificial reflections, when the electromagnetic waves reach the boundary of the grid. To overcome this limitation and terminate the grid successfully, various types of absorbing

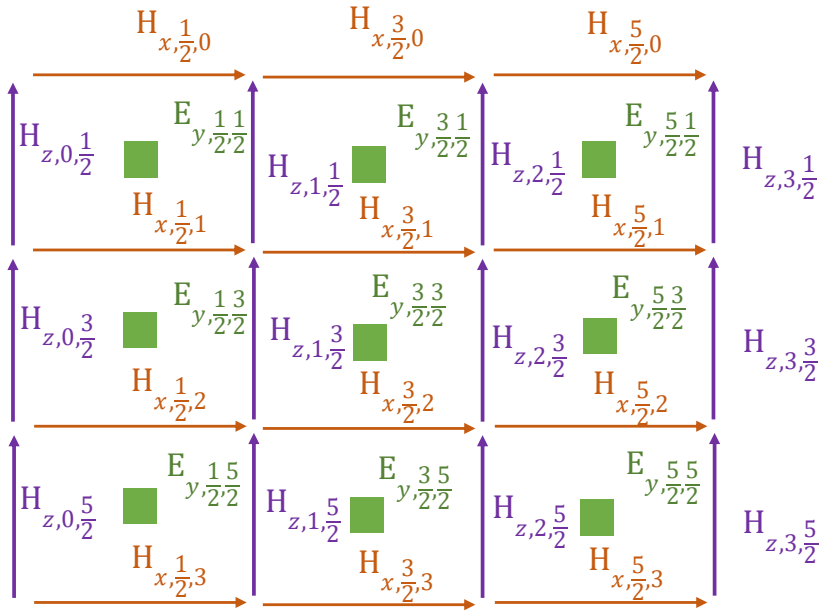


Figure 2.3: grid layout of the TE polarisation electromagnetic field components using the FDTD method.

boundary conditions have been proposed in the literature [59]. Among them, the perfectly matched layer (PML) method constitutes the most rigorous and expedient grid termination method [60]. According to this approach, the main grid is extended by placing at the end of the mesh an additional lossy layer of cells. Thus, the propagating waves in the PML region gradually attenuate and ultimately completely vanish. The parameters of the PML area are selected such as that the impedance of the additional lossy cells matches perfectly the impedance of the main grid at all angles and frequencies of an incident wave, thus ensuring that there are no spurious reflections due to the material parameter discontinuity.

For the implementation of the PML in the TE case, the electric field component is split into two sub-components denoted by E_{yx} and E_{yz}, where $E_y = E_{yx} + E_{yz}$. Furthermore, at the added lossy cells the material is assumed to be magnetic and the relationship $\sigma/\epsilon = \sigma^*/\mu$ should hold (note that the magnetic conductivity affects only the PML cells). Then, taking into consideration the application of the split field formulation, Maxwell's differential equations (2.3) – (2.5) can be replaced by four equations [60]:

$$\sigma_x E_{yx} + \epsilon \frac{\partial E_{yx}}{\partial t} = -\frac{\partial H_z}{\partial x}, \tag{2.10}$$

$$\sigma_z E_{yz} + \epsilon \frac{\partial E_{yz}}{\partial t} = \frac{\partial H_x}{\partial z}, \tag{2.11}$$

$$\sigma_x^* H_x + \mu \frac{\partial H_x}{\partial t} = \frac{\partial (E_{yx} + E_{yz})}{\partial z}, \tag{2.12}$$

$$\sigma_z^* E_z + \mu \frac{\partial H_z}{\partial t} = -\frac{\partial (E_{yx} + E_{yz})}{\partial x}, \tag{2.13}$$

where σ_x, and σ_x^* indicate the existence of electric and magnetic conductivity towards the x direction at the boundary of the grid, whilst σ_z and σ_z^* denote the same in the z direction. The four split equations (2.10) – (2.13) can be discretised to derive four update equations, following the same methodology described earlier.

2.3 Deep Learning Architectures

Propelled by the continuous advances in achievable computational power, the adoption of DL in numerous scientific disciplines has surged over recent years. DL constitutes a subset of ML, hinging on ANNs which seek to imitate human intelligence by allowing simple learning components to perform basic computational operations and connect to other learning components. The learning components are referred to as nodes, artificial neurons, or just neurons and some of their indicative use cases include (i) supervised learning, (ii) unsupervised learning, and (iii) reinforcement learning tasks [61].

In supervised learning the goal is to identify a mapping between some input and output data. Formally, this implies finding a function f that transforms some input data $\mathbf{X}$ to some target output data $\mathbf{Y}$ that constitutes a quantity of interest (QoI) for a real-world problem, *i.e.*, $f : \mathbf{X} \rightarrow \mathbf{Y}$ (note that the dimensionality of the input and out space depends on the problem and the DL architecture as it will be discussed in the next subsections). On the other hand, in unsupervised learning, there is no target output, and the goal is to distinguish distinctive motifs within the input data, allowing them to be split into different clusters. Finally, reinforcement learning aims at training autonomous agents to learn some optimal behaviour via a trial-and-error approach, while adhering to a certain reward and punishment policy.

The following subsections will discuss the functionality of some common ANN categories used in this thesis, namely (i) MLPs, (ii) CNNs, and (iii) GNNs.

2.3.1 Multilayer Perceptrons

In MLPs, the neurons are arranged into layers, where each neuron has input connections originating from the previous layer and output connections pointing towards the next layer. For regression problems, a typical MLP consists of an input layer, several hidden layers, and an output layer, as shown in Fig 2.4, where the vectors $\mathbf{X} = [x_1,\ x_2,\ ..., x_M] \in \mathcal{R}^{\mathcal{M}}$ and $\mathbf{Y} = [y_1,\ y_2,\ ..., y_{M_o}] \in \mathcal{R}^{M_o}$ are the input features and the outputs of the MLP, respectively, with M and M_o being the number of the input features and the output dimensionality, correspondingly. The term input features refers to the characteristic physical quantities that

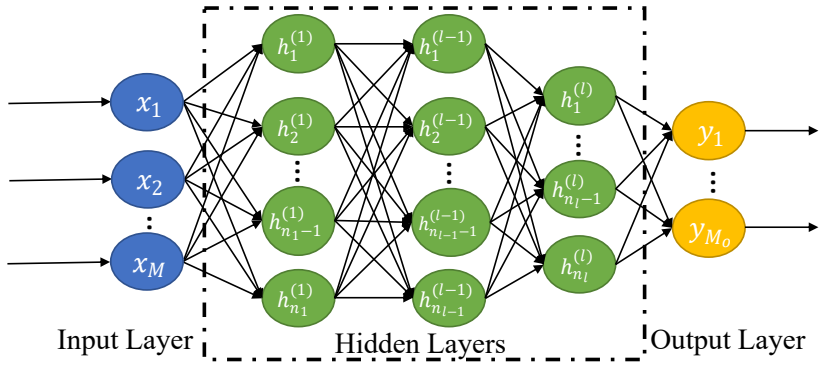


Figure 2.4: Multilayer perceptron.

affect the target QoI, which is the output of the MLP. The hidden layers consist of neurons that apply non-linear transformations to their input data. Specifically, the output, $h_j^{(l)}$, of the j-th neuron in the l-th layer is computed by applying a non-linear function, g, to the weighted sum of the previous layer neuron outputs, plus a bias term, $b_j^{(l)}$:

$$h_j^{(l)} = g\left(\sum_{i}^{n_{MLP,l-1}} \theta_{i,j} h_i^{(l-1)} + b_j^{(l)}\right), \tag{2.14}$$

where $h_i^{(l-1)}$ is the output of the i-th neuron of the previous layer, $\theta_{i,j}$ is the weight associated with $h_i^{(l-1)}$, and $n_{MLP,l-1}$ is the number of neurons in the previous layer. Some commonly used activation non-linear functions, g, are the rectified linear unit (ReLU), the tanh, and the sigmoid function [61]. The output of the MLP is estimated in a similar manner, and the QoI can assume discrete categorical (using a sigmoid or softmax activation function) or continuous values (using a linear activation function) for classification and regression problems, respectively.

2.3.2 Convolutional Neural Networks

This type of ANNs has been widely used in computer vision and image processing-related applications. In CNNs, the input data are represented as tensors, and they have three dimensions: the width, W, the height, H, and the number of channels, C, *i.e.*, $\mathbf{X} \in \mathcal{R}^{\mathcal{W}\times\mathcal{H}\times\mathcal{C}}$. A CNN typically may comprise three different layer types: convolutional, pooling, and MLPs. An example of a CNN is shown in Fig. 2.5, where an RGB image, representing the geometry of an indoor environment, is fed to a convolutional layer.

In a convolutional layer, the input tensor is convolved with n_f different filters. Each filter identifies different hidden features, and it consists of stacked two-dimensional (2D) kernels, *i.e.*, 2D square $n_w \times n_h$ weight matrices. The number of the stacked kernels is equal to the number of the input channels C, and the number of the output channels is equal to the number, n_f, of the individual filters convolved with the input tensor. The result of these convolutions, *i.e.*, the layer output, is known as the feature map. To estimate the elements

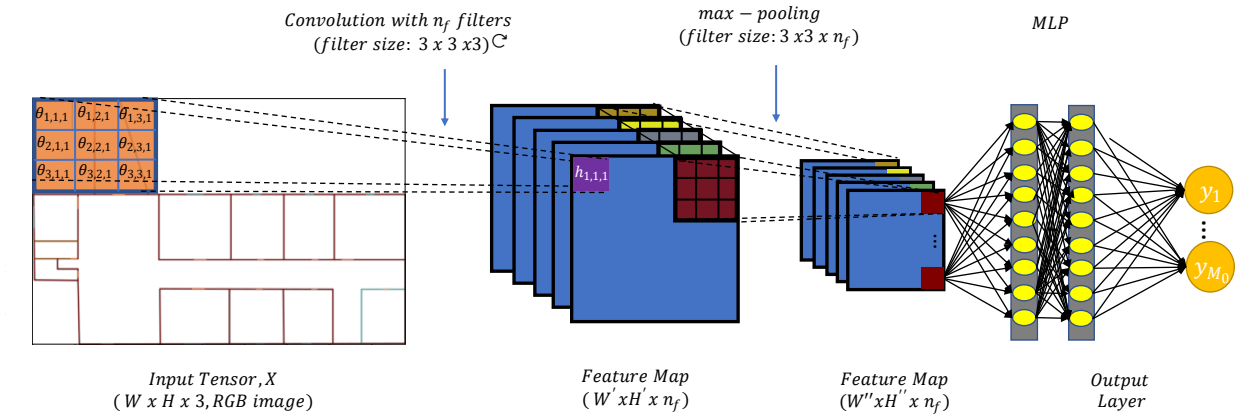


Figure 2.5: Typical operations in a CNN. The orange box depicts the kernel of a convolutional filter slid over an RGB image. The different coloured squares represent the kernels of a max-pooling filter used to extract the maximum point from a sub-area of each channel.

of the feature map, each kernel is slid with a stride, s, over the respective channel of the input tensor, and the dot product between the filter kernel weights and the points of the corresponding overlapping channel sub-area is computed. Consequently, the elements of the output feature map are computed as the cross-channel summation of the results of the aforementioned dot products. With respect to Fig. 2.5, this corresponds to computing the dot product between the orange weight boxes and the pixel intensity values of the overlapping sub-area and then adding the dot products estimated for each one of the 3 RGB channels. Indicatively, let θ_o be the weights of the o-th filter, then the elements h of the corresponding o-th channel of the feature map are computed as [61]:

$$h_{x,y,o} = \sum_{k}^{n_f} \sum_{i}^{n_w} \sum_{j}^{n_h} \theta_{o_{i,j,k}} \cdot X_{x \times s - i,\ y \times s - j,\ k}, \tag{2.15}$$

where the summation over i and j captures spatial correlations between the elements of the same channel, while the summation over k enables unveiling correlations between the different channels of the input tensor (cross-channel correlations). The convolutional layers of a CNN are typically followed by pooling layers used to simplify the representation of the encoded feature maps through downsampling and to reduce the number of the model's trainable parameters. The most common pooling operation is max-pooling. In a max-pooling layer, a single $n_w \times n_h \times n_c$ filter, where n_c is the number of the input tensor channels, is slid over the received input tensor, and the elements of the output feature map are estimated as the maximum element of the overlapping sub-area of each channel. For instance, the different coloured squares in Fig. 2.5 correspond to the kernels of the filter used to extract the maximum point from a sub-area of each channel. Finally, CNNs are commonly terminated with MLPs, which are used to estimate the final network output.

The main advantages of CNNs are the weight sharing and the connection sparsity properties. The first means that the same filter weights are applied to different parts of the

input tensor, *i.e.*, the same orange box is slid over the entire indoor geometry in Fig. 2.5. Thus, each convolutional filter can identify certain kinds of hidden features, and more importantly, it is not required to compute different weights for every point of the input tensor. The second property signifies that unlike MLPs, where each output neuron receives information from all the neurons of the previous layer, in CNNs, each neuron considers information originating only from a small sub-area of the previous layer. Thus, the numerical operations required to compute the output value of a neuron decrease substantially. For instance, for the estimation of $h_{1,1,1}$ in Fig. 2.5 (purple box), only the top-left side of the indoor geometry is considered.

The connection sparsity property also gives rise to the concept of the receptive field, which depicts the information region of the input tensor that affects the output response of a neuron. Evidently, since each convolutional layer encodes and compresses information of the previous layer, the neurons of deeper layers have indirect access to a larger area of the input tensor, and thereby a larger receptive field [61]. Hence, the shallower layers within a CNN can detect low-level features of the input tensor, whereas the deeper layers, due to their augmented receptive field, can identify more complicated abstract high-level features.

2.3.3 Graph Neural Networks

GNNs can operate on arbitrarily structured data whose interconnections form a graph-like network. Social networks, biological networks, or transportation systems constitute indicative examples falling into this category. According to the taxonomy presented in [62], GNNs can be distinguished into (*i*) recurrent graph neural networks, (*ii*) convolutional graph neural networks (ConGNNs), (*iii*) graph autoencoders, and (*iv*) spatial–temporal graph neural networks, and they can be used to perform node-level or graph-level supervised learning, as well as for unsupervised learning for graph embeddings. The following paragraphs will discuss convolutional graph neural networks, and in particular, spatial-based ConvGNNs that will be used later in the thesis, and their fundamental principle is that they update the hidden state of a central node by convolving its previous hidden state with those of the neighbouring nodes.

Generally, a graph $\mathcal{G}$ is represented by a set of n_v vertices or nodes, $\mathcal{V}$, and a set of n_e edges, $\mathcal{E}$, connecting the graph nodes, *i.e.*, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Hence, an edge $e_{i,j} = (v_i, v_j) \in \mathcal{E}$ designates that there is a connection between the nodes v_i and $v_j \in \mathcal{V}$. The set of edges emanating from a node v is called neighbourhood and it is formally defined as $N(v) = \{v' \in \mathcal{V} | (v, v') \in \mathcal{E}\}$. Both the graph nodes and edges may be associated with some features, in which case one can assume a node and edge feature tensor for the graph, $\mathbf{X}^v \in \mathcal{R}^{n_v \times M_v}$ and $\mathbf{X}^e \in \mathcal{R}^{n_e \times M_e}$, respectively, where M_v and M_e are the number of nodes and edge features, correspondingly. Apparently, one can also obtain a feature vector for a node and or an edge, denoted as $\mathbf{x}^v \in \mathcal{R}^{M_v}$ or $\mathbf{x}^e \in \mathcal{R}^{M_e}$, respectively, which constitute rows of the node and edge feature tensor.

The operation of an n_l-layer ConvGNNover over the graph nodes will induce consecutive changes to the input node features until a state of equilibrium is reached at the output. Let $\widetilde{\mathbf{H}}^{(l)} = [\mathbf{h_1}^{(l)}, \mathbf{h_2}^{(l)}, ..., \mathbf{h_n}^{(l)}] \in \mathcal{R}^{n \times M_h^{(l)}}$ be the hidden state feature matrix at the l-th layer, with $\widetilde{\mathbf{H}}^{(0)} = \mathbf{X}^v$ (note that the tilde over the matrix is used to distinguish it from the magnetic field intensity vector discussed in Section 2.2.2.2). Then, using a conventional convolutional graph neural (GCN) network [63] the hidden layer node values can estimated as:

$$\mathbf{h_i}^{(l+1)} = g\left(\sum_{j \in N(v_i)} \frac{1}{\sqrt{N_i N_j}} \mathbf{\Theta}^{(l)} \mathbf{h_j}^{(l)}\right), \tag{2.16}$$

where N_i and N_j are the neighbourhood sizes of the i-th and j-th node, respectively, $\mathbf{\Theta}$ represents the trainable weights for the l-th layer, and g is an activation function. A more general and robust implementation, referred to as message-passing neural networks (MPNNs) was introduced in [64]. Instead of applying a shared node transformation, $\mathbf{\Theta}^{(l)}$, at each layer, in MPNNs the updated node values are non-linear functions of themselves, their neighbouring node values, and the values of the corresponding edge features. Consequently, that enables applying much more versatile and complex transformations to the graph features. Specifically, the graph node values are updated through a message and a readout phase, both implemented via learned and differentiable functions. In the message-passing phase, the values of each hidden state are estimated based on the information accrued from its graph neighbourhood via two learnable non-linear operations, f_m and f_u, implemented via MLPs and employed to compute the conveyed messages, $\mathbf{m}_v$, and update the node, respectively:

$$\mathbf{m}_{v_i}^{(l+1)} = \sum_{j \in N(v_i)} f_m(\mathbf{h}_{v_i}^{(l)}, \mathbf{h}_{v_j}^{(l)}, \mathbf{x}^{e_{i,j}}), \tag{2.17}$$

$$\mathbf{h}_{v_i}^{(l+1)} = f_u(\mathbf{h}_{v_i}^{(l)}, \mathbf{m}_{v_i}^{(l+1)}), \tag{2.18}$$

where $\mathbf{x}^{e_{i,j}}$ is the edge feature vector of the edge connecting the i-th and the j-th nodes. Then, in the readout phase, an output feature vector is estimated for the entire graph via readout function f_r:

$$\mathbf{y}_{v_i}^{(l+1)} = f_r(\mathbf{h}_{v_i}^{(n_l)} | v \in \mathcal{V}). \tag{2.19}$$

2.4 Explainable Machine Learning

Although ML models have found application in various tasks, a major criticism against them is the lack of intuition behind their decisions. Indeed, their decisions are a result of consecutive complex non-linear operations applied to some input data, which renders

their comprehension by humans extremely difficult or, in most cases, impossible. This has motivated a substantial research effort to create frameworks that enable the interpretation of ML models [65]. Among the various approaches, the SHAP framework has shown to be advantageous with respect to differentiating among the various output classes, but more importantly, also with respect to conforming better with human intuition when compared to other frameworks [66].

2.4.1 The SHAP Framework

The SHAP framework is an additive feature attribution method that decomposes the response of an ML model, f_{ML}, to a set of linear explanation polynomial functions with binary input variables, $\mathbf{X}_b = [x_{b_1}, x_{b_2}, ..., x_{b_M}] \in \{0,1\}^M$ [41]. Let the binary variable polynomial expansion, p, be of the form:

$$p(\mathbf{X}_b) = \alpha_o + \sum_{i=1}^{M} \alpha_i \cdot x_{b_i}, \tag{2.20}$$

where $\alpha_i \in \mathcal{R}$ are the explanation coefficients, x_{b_i} are the binary variables that indicate if a feature is observed or not, and M is the number of the input features. The goal of additive feature attribution methods is to select the coefficients, α, of the explanation function such that p approximates locally the ML mode response, *i.e.*, $p(\mathbf{X}_b) \approx f_{ML}(\phi(\mathbf{X}_b))$, where ϕ is a function that maps the binary variables to the actual input features, $\mathbf{X}$, *i.e.*, $\mathbf{X} = \phi(\mathbf{X}_b)$. For instance, in a binary classification problem, assuming an input feature vector $\mathbf{X}$ for which the output is equal to 1, *i.e.*, $f_{ML}(\phi(\mathbf{X}_b)) = f_{ML}(\mathbf{X}) = 1$, the values of a_i multiplied by the respective binary variables x_{b_i} should add up to approximately 1. Then, due to the simplicity of the linear model, the polynomial coefficients quantify the contribution of each feature towards the output. As discussed in [41], the explanation functions must satisfy a local accuracy, missingness, and consistency property (see Section 3 of [41] for more details). To abide by these criteria, it was shown that there is a unique explanation function, for which the polynomial coefficients are equal to the Shapley values, defined as [67]:

$$\alpha_i = \sum_{\mathbf{X}'_b \subset \mathbf{X}_b \backslash \{i\}} \frac{|\mathbf{X}'_b|!(M - |\mathbf{X}'_b| - 1)!}{M!} [f(\phi(\mathbf{X}'_b)) - f(\phi(\mathbf{X}'_b \backslash \{i\}))], \tag{2.21}$$

where the coalition $\mathbf{X}'_b \in \mathbf{X}_b$ denotes all the possible binary feature vectors, $\mathbf{X}'_b$, whose non-zero entries are a subset of the non-zero entries of $\mathbf{X}_b$, with a zero entry indicating the absence of an input feature. The notation $|\cdot|$ is used to represent the number of non-zero entries in a binary vector, while the notation $\mathbf{X}'_b \backslash \{i\}$ implies setting x'_{b_i} equal to 0 and keeping intact the remaining binary values of $\mathbf{X}'_b$. Hence, the difference between the ML model response with and without considering the i-th feature, *i.e.*, $f(\phi(\mathbf{X}'_b)) - f(\phi(\mathbf{X}'_b \backslash \{i\}))$, expresses the contribution of the feature in the model's response. Finally, the term multiplying

the difference is used to weigh each contribution based on the number of non-zero features included in $\mathbf{X}'_b$.

As can be seen by (2.21), the computation of the Shapley values considers all the possible feature absence combinations in $\mathbf{X}_b$ by selecting instances of $\mathbf{X}_b$ with different binary variables set to zero. That allows capturing the interplay of the i-th feature with all the other features and enables quantifying its influence on the ML model response. However, the absence of a feature, *i.e.*, having its binary value equal to 0, cannot be straightforwardly conveyed to an ML framework. The same holds for the evaluation of, $\mathbf{X}'_b \backslash \{i\}$ that entails estimating the model's response while removing the i-th feature from the input tensor, *i.e.*, evaluating $f(\mathbf{X}'_b \backslash \{i\})$. Indeed, typically ML models use constant-size input tensors, and thus it is not possible to entirely remove an input feature.

A remedy to this problem is to replace the value of the feature that needs to be excluded with a peer feature random value coming from a backgroud data set, *e.g.*, part of the training dataset. By applying that ploy for all the possible vectors, one can anticipate that the impact of the feature sought to be removed will be ultimately sampled out, and thus it is implicitly removed. That is implemented through the mapping function ϕ: (i) when the j-th feature is present, *i.e.*, x_{b_j} =1, ϕ will return the respective feature value, x_j from $\mathbf{X}$, while (ii) when the j-th feature is absent, *i.e.*, x_{b_j} =0, ϕ will return a randomly selected value from the background data. There are different approaches to sample these random values from the background data and approximate the Shapley values. Some are model-agnostic, such as the Shapley sampling values [68] or the Kernel SHAP [41], that can be used to interpret any ML model , and others are model-specific approximations, *e.g.*, TreeShap that is employed for tree-based ML algorithms such as random forests or XGBoost [69]. The model-specific approximations are commonly dramatically faster than the model-agnostic implementations, especially for large data sets and high-dimensional input spaces, as they are designed to explicitly optimise the Shapley value computation process for a specific ML architecture.

2.5 Summary

This Chapter outlined emerging challenges in next-generation wireless networks and provided a brief overview of conventional radio propagation tools, such as the asymptotic ray tracing and the full-wave FDTD techniques, that underpin wireless network applications. Consequently, state-of-the-art deep learning architectures were discussed, detailing their operation and the type of non-linear transformation they can apply. Finally, a framework that can be leveraged to comprehend the predictions of ML models through simplistic polynomial additive feature attributions was described.

Chapter 3

Solving Maxwell's Equations with Graph Neural Networks

CEM provides numerical solutions to Maxwell's equations and it has been applied in various fields including biomedical engineering, geophysics, nanophotonics, wireless communication, and antenna and radio frequency component design [70, 71]. Existing CEM methods can be categorised based on whether they solve Maxwell's equations in a differential or integral form, or according to whether the solution is retrieved in the frequency or the time domain [72]. Some of the commonly used CEM full-wave analysis models include the finite element method [73], the method of moments [74], and the FDTD method [59], presented in Section 2.2.2.2. All these approaches necessitate high computational resources, which emerge as their main bottleneck. Among them, FDTD constitutes the most rigorous technique for high-fidelity transient (time-domain) solutions to Maxwell's equations, while it also enables broadband frequency domain responses to be acquired via a Fourier transformation of the sampled waveforms [75].

The computational burden in the FDTD method stems from the discretisation of Maxwell's equations both in space and time via a centred finite-difference approximation [76]. Once discretised, the electromagnetic fields at a given point in space and time are computed iteratively through a set of numerical equations, *e.g.*, via equations (2.7)–(2.9). The values of the spatial and time discretisation steps used in the centred finite-difference approximation are subject to constraints that must be met in order to satisfy certain stability and numerical dispersion requirements [59, 76]. This yields spatial step sizes that are a subdivision of the wavelength, and values of the time step that are smaller than the spatial step by a factor proportional to the speed of light. Inevitably, simulating electromagnetic propagation in large physical environments and very high frequencies can lead to overextended simulation times and exhaustive memory requirements, hindering the wide applicability of such models.

3.1 Expedient Solutions to Maxwell Equations

To accelerate the numerical full-wave solution of the discretised Maxwell equations, domain decomposition parallelisation schemes have been studied, aiming at partitioning the computational domain into several subdomains [77, 78]. The solution of the update equations at each subdomain is allocated to a different processing unit and subsequently, the individual solutions are combined to derive a solution for the entire domain. Typically, the decomposition is programmed so that the individual solutions are implemented using GPUs, thus speeding up the iterative update process [59, 79, 80, 81]. These approaches entail cumbersome transformations of the legacy discretised Maxwell's equations and they introduce several challenges that if not properly addressed can jeopardise the accuracy, stability, and efficiency of FDTD [82]. For instance, to avoid artificial reflections and numerical errors in domain decomposition, one must ensure reliable inter-subdomain information exchange between the neighbouring subdomains at their shared boundaries. Apart from inter-subdomain communication, it's also necessary to devise boundary conditions at the artificial boundaries between the subdomains, which if not properly implemented can produce numerical instabilities and artefacts. Furthermore, GPU CUDA-based FDTD implementations mandate advanced software expertise and knowledge of GPU architectures in order to support efficient data parallelism and memory management.

Lately, and in light of the recent advancements in AI and especially in DL, the idea of data-driven CEM has become increasingly popular. Such models proffer a unique opportunity to overcome the computational efficiency limitations associated with full-wave analysis [83, 84]. Specifically, low computational complexity DL algorithms can be trained offline to learn the underlying laws of electromagnetism, approximated through numerical methods, and they can be subsequently employed to replicate the results of these methods for new scenarios. For instance, approximate electromagnetic methods, such as ray tracing, have been coupled with DL algorithms to forge generalisable data-driven propagation models [21, 33, 85, 86]. Similarly, time-invariant full-wave methods, such as finite elements and method of moments, have also been coupled with DL to emulate steady-state results [87, 88, 89]. However, all these models are static in time, whereas transient numerical solutions to Maxwell's equations entail simulating wave propagation simultaneously in space and time. Thus, DL algorithms entwined with a time-domain method should be able to capture both the spatial dependency of the electromagnetic field intensity and its variation over time.

In this direction, the use of CNNs and sequence models has been explored [82, 90, 91, 92, 93]. Despite being able to capture spatiotemporal correlations, the scalability of such models as standalone solvers of Maxwell's equations is innately limited. Indeed, all these DL architectures invariably exhibit low versatility since conventional CNNs can be applied to fixed-size input spatial domains. More recently, GNNs were employed for the inference of electromagnetic fields [94, 45], due to their potential to respect the topological properties of

the geometry, and exploit the intrinsic graph-like structures that appear in CEM problems. Unlike CNNs which are designed to operate with structured data, GNNs can handle varying size topologies with arbitrary structure. In [94, 45], the GNN received as its input the electromagnetic properties of the simulated domain and the field components of the previous time steps and then inferred the electromagnetic field evolution. Although the trained data-driven model could predict the fields at the next time iteration with a small error, using it as a standalone CEM model was infeasible; indeed, the authors observed a fast error accumulation leading to a severe deterioration of the prediction accuracy and a breakdown of the solver after just a few iterations.

3.2 A Graph-Driven Solution to Maxwell Equations

To effectively overcome the performance barrier of the conventional transient solutions to Maxwell's equations, a graph-driven approach is proposed. This approach is founded on a very simple, yet fundamentally innovative and elegant observation: *the spatial locality of GNNs, i.e., the fact that graph node values depend on their own previous values and the values of their neighbouring graph nodes, is semantically aligned with the mechanisms underpinning the FDTD method, where the electromagnetic field values are iteratively computed based on the previous time step and the neighbouring electromagnetic field components.* From this observation springs an equivalency between the discretised Maxwell's equations and a specific GNN topology. Hence, a numerical solution to discretised Maxwell's equations can be achieved via a fixed-weight, non-trainable GNN, namely GEM, that can faithfully replicate the results of the full-wave method.

GEM consists only of two MPNN layers [64] with non-trainable and pre-determined edge weights: one responsible for updating the electric fields and the other for updating the magnetic fields in the same manner as in the FDTD method. However, exploiting the native GPU implementations of GNNs [95], GEM can solve Maxwell's equations up to 40 times faster than conventional FDTD CPU implementations, while yielding exactly the same results; interestingly, it is also observed that GEM is *at least* twice as fast as state-of-the-art FDTD implementations that exploit advanced optimisations and parallelisation, *e.g.*, grid chunking [59], even if GEM does not yet adopt those same techniques. Therefore, contrary to some recent trends in the literature [82, 90, 91, 92, 93, 94, 96], GEM proves that striving to identify appropriate input features or deep learning architectures aimed at training data-driven CEM models is no longer necessary. Instead, Maxwell's equations can be intrinsically transformed into a GEM model that fundamentally outperforms the legacy full-wave solver in terms of computational performance, without any need for training.

Furthermore, GEM can be exploited in the context of PDE resolution with neural operators, where the aim is to find operators that map an input space onto a target output

space[97, 98, 99]. In such cases, the input space may comprise all the parameters that affect the solution of a problem, such as a source excitation, initial and boundary conditions, coefficients related to the PDE, or characteristics of the physical space, while the output space portrays the solution of a PDE. However, instead of pursuing solely data-driven approaches where neural networks need to be trained to solve a system of PDEs arising in physics, one can adopt graph-driven approaches, where a GNN with properly defined edge weights can rigorously solve the exact target PDE system. Such models can effectively replace conventional scientific computing techniques since they provide identical results and are substantially faster. Consequently, to contrive learnable operators, non-linearities and trainable components can be included in the graph-driven solvers, to either account for real-world measured data (in cases where the fidelity of partial differential equation is not guaranteed) or to operate outside the stability limits that pose strict requirements for the spatial and temporal discretisation steps of conventional scientific computing techniques.

3.2.1 Equivalent Graph Neural Network Representation of Discretised Maxwell's Equation

The discretised Maxwell equations, *e.g.*, (2.7)–(2.9), enshrine an underlying connection between the electromagnetic fields which becomes more evident by inspecting the layout of the electromagnetic field components in Fig. 3.1b: *the grid of magnetic and electric field nodes resembles a graph whose vertices correspond to the electromagnetic field components appearing in the discretised Maxwell equations.* Based on this intuition, by properly arranging the edge connections between the graph nodes and the corresponding edge weights, one can forge a network of neurons such that the exchange of information between the nodes faithfully reenacts the iterations leading to the solution of the discretised Maxwell equations. Specifically, the electric field nodes have loop connections to themselves as well as direct connections to their neighbouring magnetic field nodes, whilst the magnetic field nodes are connected to the adjacent electric fields and they also have a self-loop connection, as shown indicatively in Fig. 3.1c for the TE polarisation case. For each connection, suitable edge weights can be straightforwardly derived by observing the counterpart coefficients of the respective field components appearing in the discretised Maxwell equations, as also illustrated in Fig. 3.1a and Fig. 3.1d.

Then, the two-step iterative numerical solution to Maxwell's equations can be achieved through a two-layer GNN comprising two MPNNs [64]. As mentioned in Section 2.3.3, in MPNNs the graph node values are updated through a message and a readout phase, both implemented via learned and differentiable functions. However, GEM proves that to solve differential equations the message function does not need to be learnable since the weighting coefficients can be readily deduced from the discretised equations, while the readout function

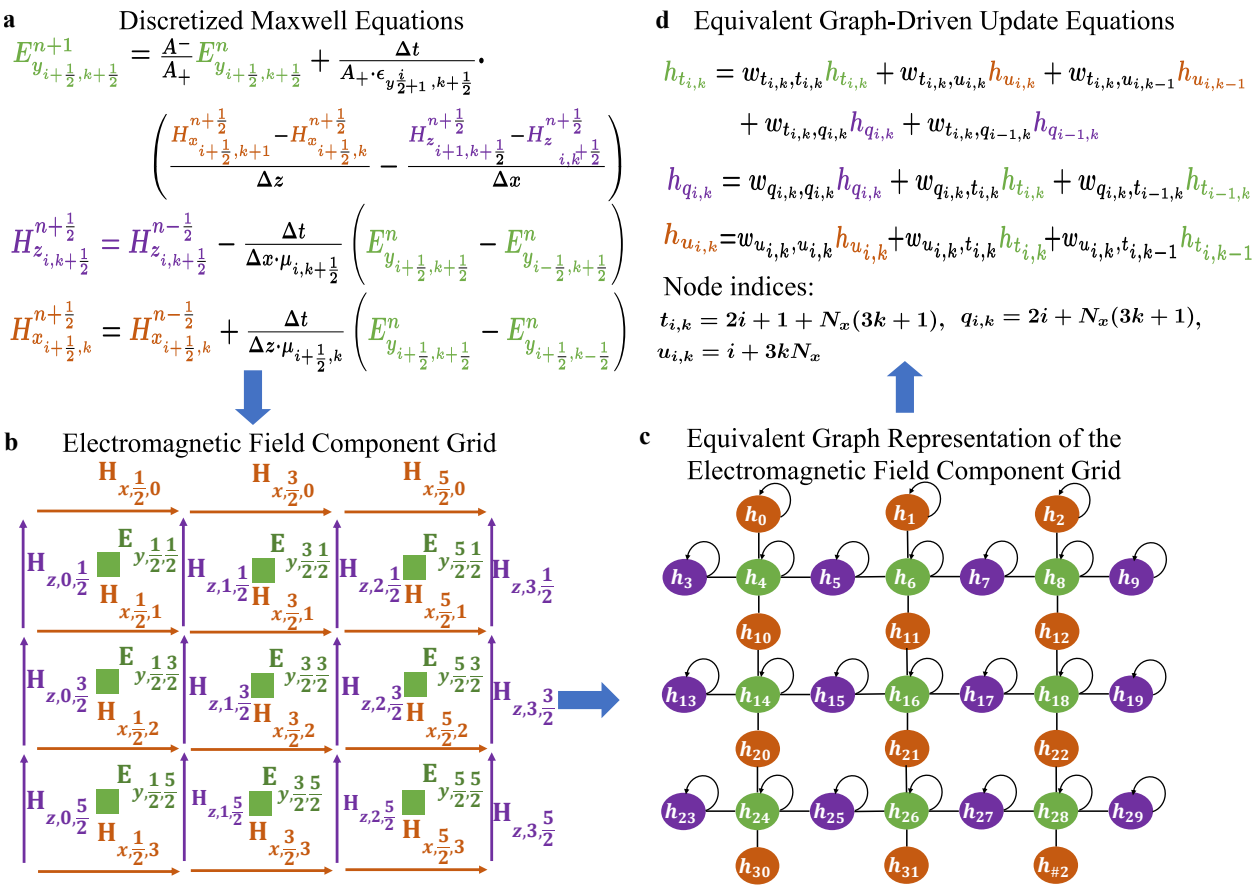


Figure 3.1: Graph-driven solver of Maxwell's Equations; the E_y, H_z, and H_x components are shown in green, purple, and dark orange colour, respectively. **a**, Discretised TE Maxwell equations along with **b**, a corresponding example 2D TE grid representation of the electromagnetic field component arrangement in space. **c**, The grid implicitly assumes an equivalent graph topology, and consequently, **d**, the electromagnetic field update equations can be expressed in a graph-driven manner via mathematical operations between the graph nodes.

can be omitted. Consequently, building on the burgeoning software (in terms of existing libraries) and hardware advances on the implementation of GNNs, enabling their seamless deployment into GPUs, one can attain a numerical solution substantially faster than with conventional methods.

Overall, the discretization of Maxwell's equations in space and time creates a grid of nodes that intrinsically assumes a graph structure, and a straightforward way to numerically solve Maxwell's equations entails a simple inference pass of a two MPNN layer GNN. Based on this fundamental insight, and without loss of generality (in the sense that via duality one can derive an equivalent scheme for the transverse magnetic case, or easily extend it to 3D), the following original propositions are formally articulated for the TE case:

Proposition 1 *The discrete electromagnetic field nodes of an $N_x \times N_z$ FDTD grid, providing a numerical solution to Maxwell's equations, can be perceived as a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with $3 \times N_x \times N_z$ nodes. For the set of nodes $\mathcal{V} = v_0, v_1, ...v_{3\times N_x \times N_z - 1}$, a node index equal*

to $t_{i,k} = 2i + 1 + N_x(3k + 1)$ *represents electric field components, whereas indices equal to* $u_{i,k} = i + 3kN_x$ *and* $q_{i,k} = 2i + N_x(3k + 1)$ *correspond to the* H_x *and* H_z *components, respectively (* $0 \leq i \leq Nx - 1$ *and* $0 \leq k \leq Nz - 1$*).*

The set of edges, $\mathcal{E}$*, indicates the connections and interactions between different electromagnetic field components. Specifically, in GEM,* $\mathcal{E}$ *comprises the neighbourhoods of the electric and magnetic field components, defined as follows: (i) for the electric field nodes,* $N(v_{t_{i,k}}) = \{v_{t_{i,k}}, v_{u_{i,k}}, v_{u_{i,k-1}}, v_{q_{i,k}}, v_{q_{i-1,k}}\}$*, (ii) for the* H_x *nodes,* $N(u_{t_{i,k}}) = \{v_{u_{i,k}}, v_{t_{i,k}}, v_{t_{i,k-1}}\}$*, and (iii) for the* H_z *nodes* $N(v_{q_{i,k}}) = \{v_{q_{i,k}}, v_{t_{i,k}}, v_{t_{i-1,k}}\}$*. The non-trainable edge weights are then straightforwardly derived from (2.7)–(2.9), e.g., for an edge* $e_{t_{i,k},t_{i,k}} = (v_{t_{i,k}}, v_{t_{i,k}})$ *the corresponding weight is equal to* $w_{t_{i,k},t_{i,k}} = A_-/A_+$*.*

Proposition 2 *The nodes of* $\mathcal{G}(\mathcal{V}, \mathcal{E})$ *form a network of neurons, whose hidden states,* h_l*, depict the electromagnetic field evolution over space and time. Consequently, updating the discretised Maxwell's equations is equivalent to a two-step message-passing process between the network neurons. During the first step, messages flow towards* $h_{u_{i,k}}$ *and* $h_{q_{i,k}}$ *from neighbouring nodes, and the new hidden state values are computed as* $h_l = \sum_{l_n \in N(v_l)} w_{l,l_n} h_{l_n}, v_l \in \{U, Q\}$*, with* U, Q *being the set of* H_x *and* H_z *nodes, respectively. In the second step, the neurons updated earlier remain constant and messages are conveyed towards the electric field components,* $h_{t_{i,k}}$*, using the same update formula, but with* $v_l \in T$*, where* T *is the set of* E_y *nodes.*

Propositions 1 and 2 compose GEM, which is conceived as a specialised case of the general MPNN module, where the edge weights are tuned in advance without training according to the physical problem. Conspicuously, GEM benefits from the native GPU implementation of MPNNs. Furthermore, while the message-passing and the readout functions are typically implemented via computationally expensive fully connected layers, GEM permits a reduction in the complexity of the MPNN via the use of domain knowledge that allows a proper pre-determined message-passing function to be selected and the readout function to be eliminated. Ultimately, the solution of Maxwell equations is elegantly derived by a fast matrix multiplication between edge weights and the graph node hidden states, followed by an aggregation operation.

3.2.2 Graph-Driven Solution with Absorbing Boundaries

In the presence of the absorbing boundaries, the principles of GEM remain the same with the only difference being that the graph node number is increased. For the 2D TE case, the node number is equal to $4 \times N_x \times N_z$ and the edges are adapted to take into account the additional field components introduced due to the split field formulation. Specifically, node indices equal to $t_{x_{i,k}} = 3i + 1 + N_x(4k + 1)$ and $t_{z_{i,k}} = 3i + 2 + N_x(4k + 1)$ represents the E_{yx} and E_{yz} components, while the indices $u_{i,k} = i + 4kN_x$ and $q_{i,k} = 3i + N_x(4k + 1)$ are linked to the H_x and H_z components, respectively ($0 \leq i \leq Nx - 1$ and $0 \leq i \leq Nz - 1$). Additionally, the

neighbourhoods of the electric and magnetic field components are now changed as follows: (i) for the E_{yx} component, $N(v_{t_{x_{i,k}}}) = \{v_{t_{x_{i,k}}}, v_{q_{i,k}}, v_{q_{i-1,k}}\}$, (ii) for the E_{yz} component, $N(v_{t_{z_{i,k}}}) = \{v_{t_{z_{i,k}}}, v_{u_{i,k}}, v_{u_{i-1,k}}\}$, (iii) for the H_x nodes, $N(v_{u_{i,k}}) = \{v_{u_{i,k}}, v_{t_{x_{i,k}}}, v_{t_{x_{i,k-1}}}, v_{t_{z_{i,k}}}, v_{t_{z_{i,k-1}}}\}$, and (iv) for the H_z nodes $N(v_{q_{i,k}}) = \{v_{q_{i,k}}, v_{t_{x_{i,k}}}, v_{t_{x_{i-1,k}}}, v_{t_{z_{i,k}}}, v_{t_{z_{i-1,k}}}\}$. Finally, the corresponding edge weights can be derived in a similar way as before by observing the update equations that arise by discretising (2.10) – (2.13).

3.3 Numerical Results

The validity of GEM is verified by conducting 2,000 simulations with an in-house FDTD package using CPU parallelization and GEM and juxtaposing the electromagnetic field evolution over space and time for the two methods. For each simulation, one or more objects are laid in the simulation grid (cylinders, triangles, rectangles, walls) with randomly sampled electromagnetic parameters, and with a sinusoidal, Gaussian, or modulated Gaussian source placed at a random position within the grid. Rectangular grids with different side sizes of 200, 400, 600, 800, and 1,000 cells are considered, while the number of simulation time steps is selected so that by the end of the simulation the fields have decayed or, in the case of a sinusoidal source, there is a steady-state response. To avoid prolonged simulation times when using the FDTD technique, the number of simulations, N_{sim}, is reduced as the grid size increases. Specifically, we conduct 800, 600, 300, 200, and 100 simulations for the grid sizes mentioned previously.

GEM is implemented in PyTorch Geometric [95] using an NVIDIA A100 with 80GB of memory. Both GEM and the FDTD implementation use a PML to avoid artificial reflections at the boundaries of the simulated grid. It is paramount to remark that GEM does not require training, as its structure is already defined to perfectly mimic the numerical solution to Maxwell equations. In this sense, GEM is innately superior to all data-driven approaches proposed for CEM [82, 90, 91, 92, 93, 94, 96, 100], since its predictions do not introduce any error, and it employs very simple operations to infer the electromagnetic fields; specifically, two matrix multiplications and two aggregations. At the same time, GEM does not exclude the integration of learnable components in future work, *e.g.*, for the inference of multiple time steps simultaneously, where the messages could be a function of the pre-determined edge weights, the node values, and other physics-based information. Consequently, in addition to a guaranteed converged initial solution, GEM yields higher versatility and potential for further improvement.

Indicatively, Figs. 3.2a and 3.2b present several snapshots of the electric field component, E_y, computed via GEM and FDTD, respectively, for the simulated scenarios whose geometry is depicted in Fig. 3.2c. These qualitative results visually convey how GEM reproduces the propagation of electromagnetic waves in space and time while also capturing their interaction

with the environment in a very precise manner. This is further corroborated by Fig. 3.2d which portrays the waveform recorded at a random point of the simulated domain with the two models, demonstrating once more a perfect correspondence between the results. From a quantitative standpoint, the coefficient of determination [101], R^2, between the spatiotemporal output of GEM and FDTD is computed across the whole set of experiments conducted. The estimated R^2 is equal to 0.9999, clearly indicating the equivalence of the two methods in terms of output quality.

To showcase the wide applicability of GEM, two distinct examples are considered: one from the field of computational biology and the other from photonics. The first is to measure the electric field intensity and specific absorption rate (SAR), , distribution of microwaves interacting with a biological tissue [102]. A four-layer structure is simulated, representing an air-skin-fat-muscle interface. The skin layer has a thickness of 2 mm, that of the fat layer is 33 mm, whilst the muscle layer occupies the remaining space below the fat layer until the end of the grid. The electromagnetic properties of each layer constituting the biological tissue are set according to [103]. A Gaussian point source excitation is placed 10 mm above the skin, with the frequency of the propagation pulse set to 6.8 GHz. The spatial grid is discretised using 20 points per the shortest wavelength of the structure, *i.e.*, that occurring

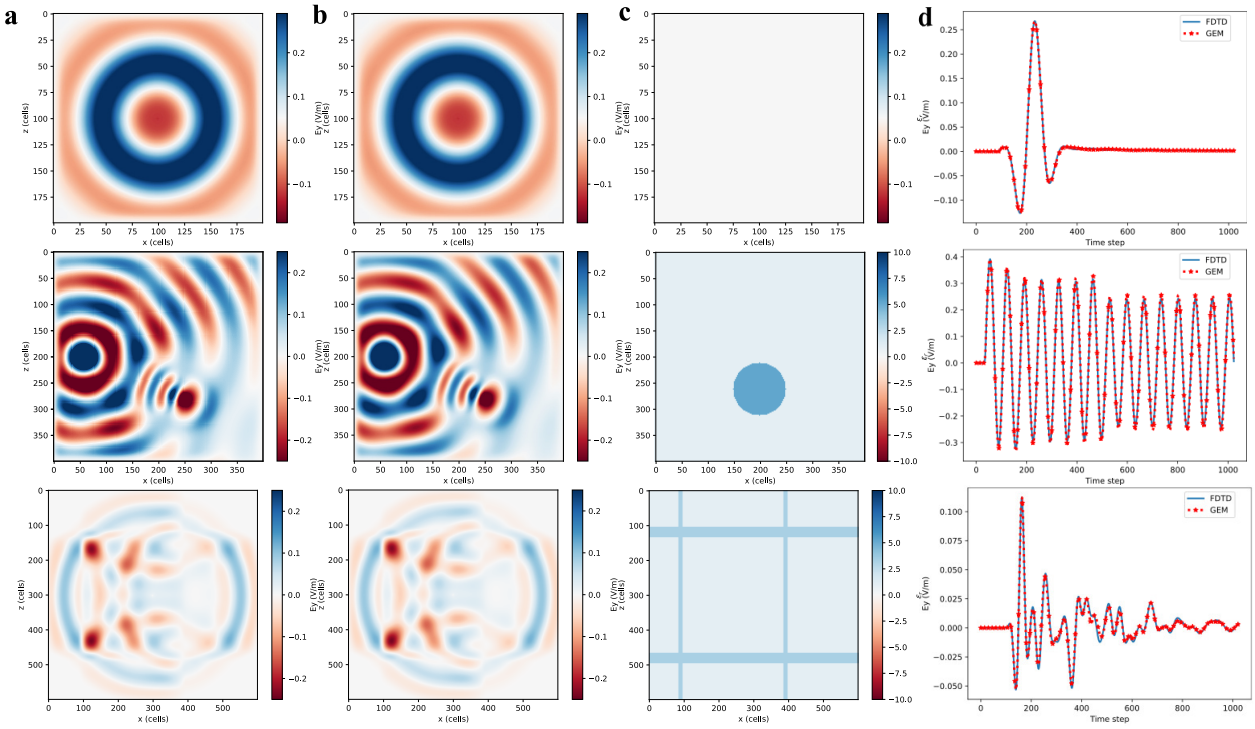


Figure 3.2: Computational emulation of electromagnetic wave propagation for three use cases (top, centre and bottom rows); **a** shows a snapshot of the electric field E_y evolution simulated via GEM, whilst **b** depicts the snapshot at the same time instance as computed with FDTD. The simulated domain is shown in **c**, whereas **d** illustrates the waveform of the E_y component sampled at a random point of the simulated grid.

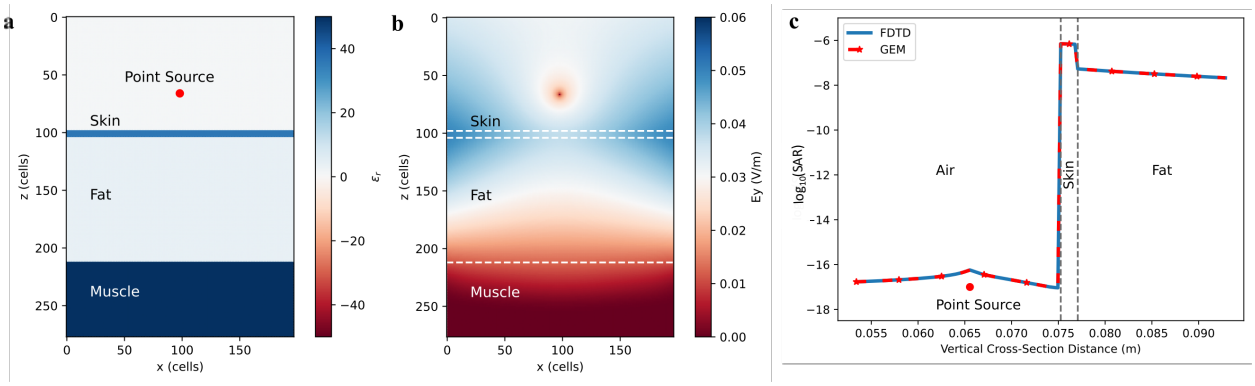


Figure 3.3: Estimation of SAR with GEM for a biological tissue. **a**, Electric permittivity for the air, skin, fat, and muscle layers of the biological tissue. **b**, Snapshot of the electric field intensity dissipated throughout the tissue layers. **c**, Computed SAR with GEM and FDTD along a vertical cross-section passing through the point source.

in the muscle, and the Courant limit is set to 0.95. A snapshot of the, simulated via GEM, electric field intensity diffused throughout the biological tissue is shown in Fig. 3.3b, while Fig. 3.3c presents the logarithmic SAR distribution on a vertical cross-section passing through the source estimated via GEM and conventional FDTD. Evidently, the results of the two methods are identical and yield the same SAR, which exhibits a sharp increase when the electromagnetic waves penetrate the skin and subsequent attenuation while propagating through the fat layer. The simulation time with GEM is 9.5 seconds while the conventional FDTD method takes approximately 470 seconds, *i.e.*, almost 8 minutes which is 50 times more than that required by GEM.

The second application emulates wave propagation in a 2D photonic crystal, implemented by simulating a THz Gaussian pulse traversing a periodic dielectric waveguide which assumes a 1D periodicity [72]. The simulated structure is shown in Fig. 3.4a; the waveguide material has a dielectric constant $\epsilon = 12$, and exhibits a periodicity imposed by two pairs of round cavities with radius $r_c = 0.36$ μm spaced 1 μm apart from each other. The two cavity pairs are separated by 1.4 μm and each one of them comprises 3 circular holes. A line source is placed at the beginning of the simulated grid by exciting every grid cell along the vertical direction of the waveguide with a 75 THz Gaussian pulse. The snapshot of the electric field component, emulated via GEM, shown in Fig. 3.4b illustrates how light is trapped within the waveguide due to the existence of the periodic structure. The sampled waveform shown in Fig. 3.4c evinces again a full equivalency between the graph-driven and the conventional approach.

To highlight the expediency of the graph-driven approach the impact of the simulated domain size on the computational time is probed. To this end, in a free space scenario, the size of the simulated domain is gradually increased considering rectangular grids with a side equal to 250, 1,000, 2,000, 3,000, 4,000, 5,000, 6,400 cells, and to quantify the uncertainty in the

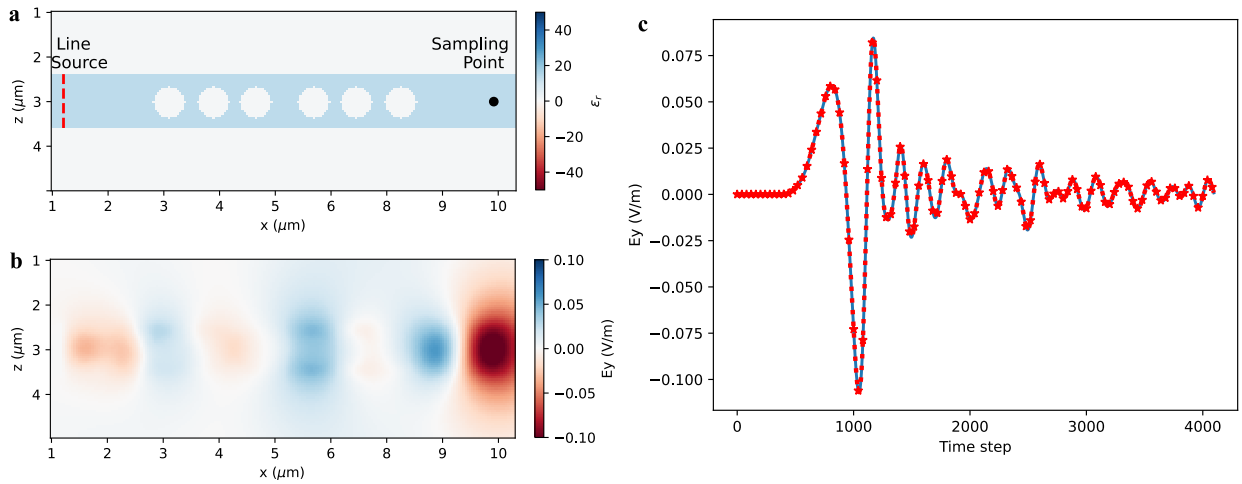


Figure 3.4: Wave propagation in a cavity waveguide. **a** shows the geometry of the waveguide, while **b** depicts a snapshot of the electric field intensity across the waveguide. **c** Simulated waveform via GEM and FDTD sampled at the end of the waveguide.

execution time 5 simulations are run for each cell size, lasting 1,024 time steps. In addition to comparing the efficiency of GEM to that of an in-house FDTD code using CPU parallelization, GEM is benchmarked against a publicly available FDTD suite, namely Meep developed at MIT [104]. It is noted that the in-house FDTD is the standard implementation of the method without leveraging any laborious optimisation techniques. On the other hand, Meep constitutes a more sophisticated approach, differing from typical FDTD implementations. Specifically, Meep enables parallelisation by exploiting symmetries, by efficiently storing auxiliary fields only in certain regions (*e.g.*, boundary absorbing layers), and by splitting the computational domain into chunks which are combined into an arbitrary topology via boundary conditions, as discussed in [104]. Furthermore, Meep also supports distributed-memory parallelism over multi-node clusters, which is referred to as Parallel Meep, and in the discussion below the simulations are distributed over 10 different processes in a server with 64 CPUs, and 315 GB of RAM (note that this is the same server hosting the NVIDIA A100).

Fig. 3.5a presents the time required to conduct a single FDTD simulation along with the 95% confidence intervals. As can be observed, the computational time scales exponentially as the grid size increases. For the largest grid size considered, a single FDTD simulation with the in-house CPU-based code takes approximately half an hour, as shown in the inset plot. This time is reduced to approximately 3 and 2 minutes by Meep with and without parallelisation, respectively. However, GEM favoured by its native GPU implementation can complete the simulation in less than a minute, *i.e.*, almost 40 faster than the standard FDTD implementation and 2 to 4 times faster than an existing highly elaborate state-of-the-art open-source software. Note that currently, GEM does not make use of any FDTD optimisation techniques, *e.g.*, grid chunking or exploiting symmetries, employed in Meep. However, in the future similar functionalities can be embedded in GEM as well, further enhancing its

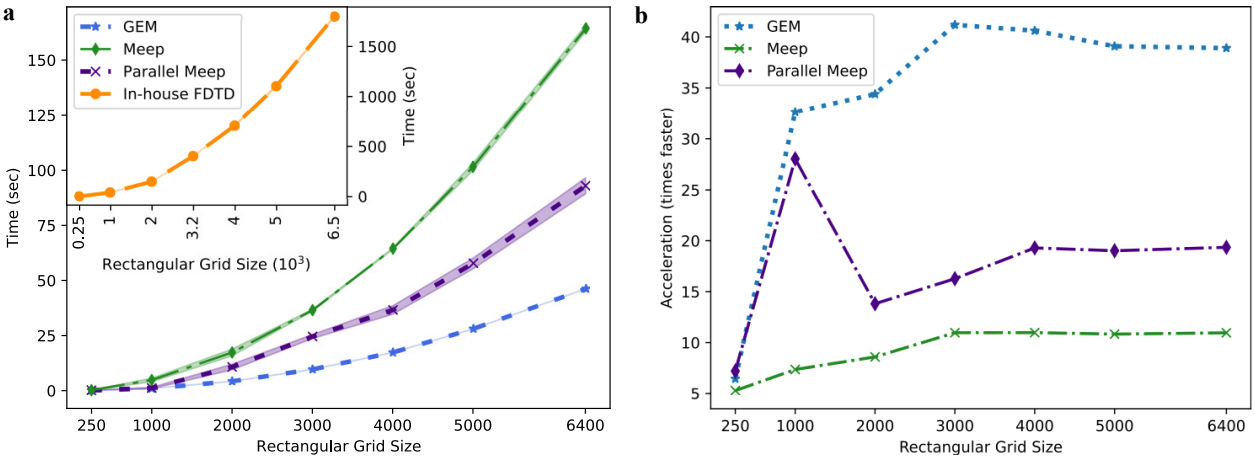


Figure 3.5: Performance comparison. **a**, Simulation time for GEM, our in-house FDTD code and Meep. **b**, Performance gain achieved by GEM and Meep compared to the in-house FDTD code using CPU parallelization.

efficiency. Yet, even without leveraging them, GEM can outperform a state-of-the-art CEM solver in terms of speed. Additional insight regarding the acceleration achieved by the different benchmarks over the in-house FDTD code can be provided by inspecting Fig. 3.5b. It can be seen, that as the grid size increases the gain over the conventional FDTD technique plateaus, with GEM, Meep, and Parallel Meep being approximately 40, 20, and 10 times faster than it, respectively.

3.4 Summary

GEM tackles an open problem: specifically, how to train artificial intelligence models to comprehend the laws of electromagnetism and emulate electromagnetic field evolution over space and time. The simple yet counter-intuitive insight is that there is no need to do so, since there exists a network of neurons, with specific connections and weights, that yields a solution that is equivalent to that of the discretised Maxwell equations. Consequently, the exchange of messages between the network neurons resembles the spatiotemporal evolution of the electromagnetic fields. Unlike, data-driven CEM approaches, which are ill-fated owing to the fast error accumulation in their predictions, GEM yields no error due to the proper setting of the connections between the graph nodes and the corresponding edge weights. The minimal computational complexity entailed by the properly tuned MPNNs along with their native GPU implementation leads to a 40× acceleration over a direct implementation of the standard FDTD technique, while it also outperforms in terms of efficiency previous elaborate state-of-the-art transient response solvers.

The benefits stemming from GEM are not confined to the field of physics, but they can have a profound impact on many other sciences in which it is necessary to capture the interaction of electromagnetic fields with matter, such as biomedical engineering and

photonics. More importantly, it is anticipated that GEM opens up a new field that can potentially lead to a breakthrough in computational sciences heralding a new class of graph-driven solvers. Indeed, differential equations describe a plethora of physical phenomena, *e.g.*, the Navier–Stokes equations which depict the motion of viscous fluid substances, or the Pennes Bioheat equation which captures the heat transfer between tissue and blood. Consequently, similar GNNs structures exist for these problems as well, and we thus advocate moving from the long-discussed data-driven solutions [100] to graph-driven approaches where differential equation systems are solved via neuron message passing. Graph-driven solutions enable simulating large domain physical problems, which is impossible to accomplish with conventional techniques, and it is anticipated that when it is applied to other fields it will outperform existing legacy or data-driven computational approaches.

Chapter 4

Radio Propagation Modelling in the Era of Artificial Intelligence

To bridge the gap between accuracy and efficiency, demonstrated by the conventional empirical and deterministic propagation models, the idea of machine ML-based propagation modelling techniques has recently become popular [11]. These models aim at distilling knowledge from propagation solvers or real-world measured data and eventually become capable of predicting characteristics related to radio wave propagation. As shown in Figure 4.1, data-driven propagation models leverage a set of descriptive input features that can be processed by an ML model, agnostic to the laws of electromagnetism, in order to develop an understanding of how such features affect radio wave propagation. Eventually, the goal is to retrieve a mapping that transforms the input data features to a target physical QoI related to the wireless channel. Thereby, the input features ought to convey information that is intrinsically correlated to the target QoI in order to facilitate the training procedure and enhance the generalisability of the model. This information can be related to the physical environment itself or to the details and configuration of the wireless communication system. For instance, for outdoor environments, physics-based input features can be related to the city layout, the building height, or the terrain type, *e.g.*, sea, forest, or urban. Likewise, useful information about indoor environments can be the wall construction materials, as well as, the respective electromagnetic parameters, the wall height and thickness, and the number of walls between the Tx and the receiver (Rx). Parameters related to the wireless system configuration can be the operating frequency, the antenna radiation pattern and polarisation, and the height of the Txs and Rxs.

The physical QoI at the output of the ML model can be related to large-scale fading, such as the PL and shadowing, or to small-scale fading, *e.g.*, mean angle of arrival or delay and angular spread. Furthermore, the estimates of the target QoI can be real-world measured data and/or, due to the scarcity of such data, synthetic data generated by a high-performance propagation solver. In cases where only synthetic data are used, the ML model can effectively

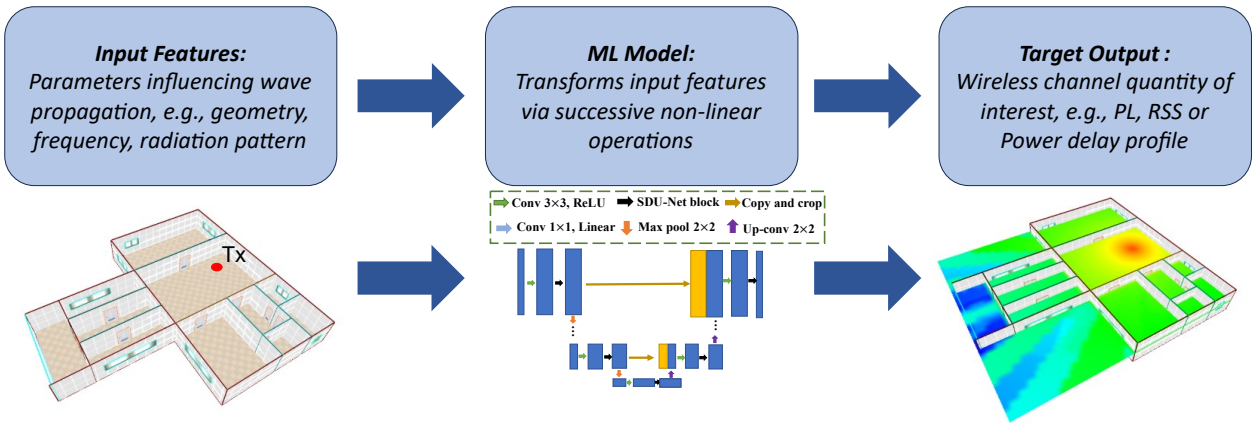


Figure 4.1: Pipeline of a data-driven propagation model.

learn to reproduce the results of the high-performance propagation solver, but in substantially reduced computational time, while simultaneous training with real and synthetic data can be used to enhance or calibrate existing propagation models. Finally, since the target outputs are continuous physical quantities, radio propagation modelling is commonly treated as a regression problem; however, if one considers discrete values of the output QoI, *e.g.*, assume that PL can take integer values such as 40, 41, ..., 120 dB, disregarding the intermediate real number values, then it is also possible to be treated as a classification problem [105].

4.1 EM DeepRay: An Expedient, Generalisable, and Realistic Data-Driven Indoor Propagation Model

This section discusses how an expedient, generalisable, and credible data-driven indoor radio propagation modelling tool is developed by training a DL convolutional encoder-decoder with simulated data generated via a high-performance radio propagation simulator with [33]. Specifically, through consecutive non-linear transformations, the DL module encodes physics-based information of an indoor environment, such as the electromagnetic properties of the walls, and decodes it as the PL heatmap, *i.e.*, the spatial distribution to the signal attenuation, for the environment of interest. In what follows, the format of the input and output data, the general framework pipeline, and the structure of the convolutional encoder-decoder used to implement EM DeepRay are discussed. Furthermore, numerical results are provided, demonstrating that indeed AI can be used to effectively replicate the results of electromagnetic solvers. The implementation presented in this Section is limited to inferring the PL distribution in a 2D plane, while Section 4.2 will outline a more general problem formulation enabling inferring PL in the 3D space.

4.1.1 Input features and target output

The input features exploited by EM DeepRay are: (i) the permittivity and (ii) the conductivity of the building materials, (iii) the distance between the Tx and every point within the simulated grid, (iv) and the free space path loss (FSPL) at every point of the indoor geometry, assuming the absence of the building. The predicted parameter is the PL at every spatial point within an indoor geometry, given a specific transmitter position and building layout. The reasoning behind the input feature selection and the generation of the input tensor is explained in the following paragraphs.

Wave propagation is affected by four basic mechanisms: reflection, transmission, diffraction, and diffuse scattering [50]. It is well-known that the impact of these mechanisms depends on the wavelength λ of the propagating wave. In particular, reflections and transmissions require that the objects found within the propagation environment are electrically large, *i.e.*, their dimensions are much larger than λ. When λ is much larger than the object (sharp edges, small openings), the propagating waves will bend around the object and diffraction will occur. Finally, diffuse scattering requires that abrupt variations of the surface height are an order larger than the wavelength. The impact of these mechanisms also depends on the material proprieties, which in turn are also related to the radio signal frequency [106].

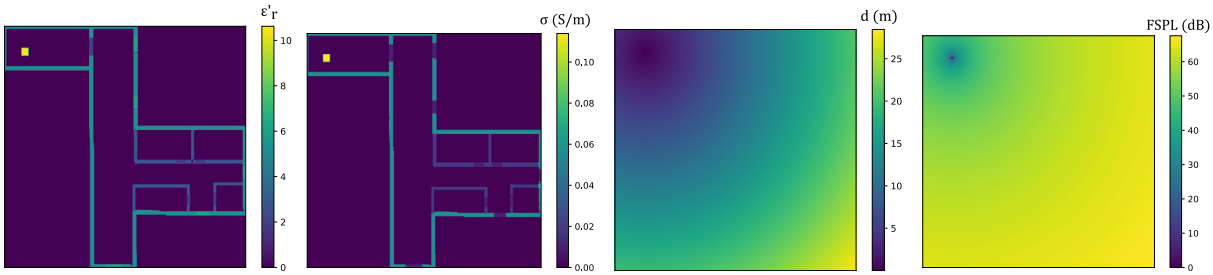
The electromagnetic properties of a material can be quantified by its permittivity and its permeability. In this work, it is assumed that all the construction materials found within the simulated indoor geometries are non-magnetic, *i.e.*, they have a constant permeability equal to μ_0. The permittivity is formulated as: $\epsilon = \epsilon_r \epsilon_0 = (\epsilon'_r - j\frac{\sigma}{\epsilon_0 \omega})\epsilon_0$, where ϵ_r is the relative permittivity. Thus, to consider how different materials affect wave propagation two input channels are used, which convey information regarding the relative permittivity. Specifically, the first channel depicts ϵ'_r at every point of the simulated grid, and the second includes the value of the conductivity. An example of these channels is shown in Figure. 4.2a and Figure. 4.2b, respectively. The conductivity is modelled as $\sigma = c_\sigma f_{Tx}^{d_\sigma}$, where f_{Tx} is the frequency of the propagating wave in GHz, and the values of ϵ'_r, c_σ and d_σ, are derived from the ITU-R P.2040-1 Recommendation [106]. The first channel allows EM DeepRay to understand the presence of an object and infer the strength of the reflected and the transmitted components of an electromagnetic wave impinging onto it. The second channel accounts for the attenuation that an electromagnetic wave undergoes while it propagates through an absorbing medium.

Another important parameter that affects wave propagation in indoor environments is wall thickness. In outdoor ray tracing simulations, building walls are modelled as semi-infinite spaces due to the high losses, thus the penetration into buildings or the multiple reflections within the building facets can be omitted. In indoor environments, this assumption does not hold, since the walls are thinner than the building facets, and for materials such as plasterboard or wood, the losses are considerably smaller. Hence, objects such as walls, doors,

and windows are modelled as slabs, and consequently, the reflection and the transmission coefficients depend on the thickness of the slab. To account for that, a third channel is used, which depicts the physical distance between the transmitter and every point in the simulated grid. An example of the distance channel is shown in Figure. 4.2c. Thus, when a convolutional kernel is applied to a sub-area of the input tensor, apart from detecting the wall type through the first two channels, it can also consider the wall length via the difference between the values of the third channel.

Furthermore, the third channel can also provide a good estimate of the deterioration of the signal over distance. To further enhance the awareness of the data-driven model regarding the impact of distance on wave propagation, a fourth channel is added, which includes the FSPL for each point in the simulated grid. It is noted that if EM DeepRay was trained only over one frequency, the fourth channel would have been a simple transformation of the third channel, and thus this information would have been redundant. However, EM DeepRay is trained over multiple frequency bands, hence the fourth channel helps the data-driven model to unveil correlations over the frequency and the space domain.

The target output of EM DeepRay is a tensor including the PL at every point of the simulated grid; however, a similar treatment can be applied for another QoI such as the received signal strength (RSS) or the delay spread. The output tensor assumes a single channel, depicting the simulated QoI for a receiving horizontal plane found at a certain height. In particular, for the results shown later in this section, the output channel represents the PL distribution at the horizontal plane at a height equal to 1.5 m above the ground, and the target PL values are computed using a ray tracing simulator [107]. Ultimately, the goal is to train a convolutional encoder-decoder that encodes the input tensor and then decodes it as a PL heatmap, *i.e.*, to find a mapping between the material properties, physical distance, and frequency space and the PL distribution space, *i.e.*, $f : V_{\epsilon,\sigma,d,FSPL} \rightarrow V_{PL}$.



(a) Permittivity channel. (b) Conductivity channel. (c) Distance channel. (d) FSPL channel.

Figure 4.2: Example of the input tensor channels for a random sample at an operating frequency of 2 GHz.

4.1.2 Workflow

The block diagram of EM DeepRay is shown in Figure 4.3a. For each ray tracing simulation two pieces of information are obtained: (i) a blueprint, X_i, representing the geometry of an indoor environment, and (ii) a tensor, y_o, comprising the simulated PL values at each point of the indoor geometry. The target PL tensor is converted to a grayscale image, $y_{o,gs}$, and consequently, it is bilinearly resized to a standard-size image, y; *i.e.*, the values of y at each point of the new grid are computed through a linear interpolation first in the horizontal and then in the vertical direction, using the values of the four closest grid points of $y_{o,gs}$ [108]. Similarly, the input geometry blueprint is also resized to a standard-size image. Although convolutional encoder-decoders do not necessitate that all the input and output samples have the same size, typically when this is not the case, there is a decrease in the model's performance [109]. Additionally, having input and output samples of the same size facilitates the training procedure, since during training the data are processed in batches by the GPU, and all the samples within one batch must have the same size. Hence, resizing all the samples to a certain size makes the training procedure smoother; otherwise, it would have been necessary to plan in advance and create the batches so that the same-size samples are found within each batch.

Then, the resized blueprint of the input geometry is used to extract physics-based information and make end-to-end, pixel-to-pixel predictions, translating the physics-based information of the input geometry to a PL heatmap. Specifically, as discussed in Section 4.1.1, for each indoor geometry image, X_i, a four-channel input tensor, X, is derived, representing: (i) ϵ'_r and (ii) σ at each point of the grid, (iii) the physical distance between the transmitter and each point, and (iv) the FSPL for each point. The first two channels are derived by matching the colours of the blueprint, *i.e.*, different RGB pixel values, to the permittivity and conductivity values of the ITU standard [106]. The 2D physical distance, $d_{i,j}$, for the (i, j)-th pixel is computed as:

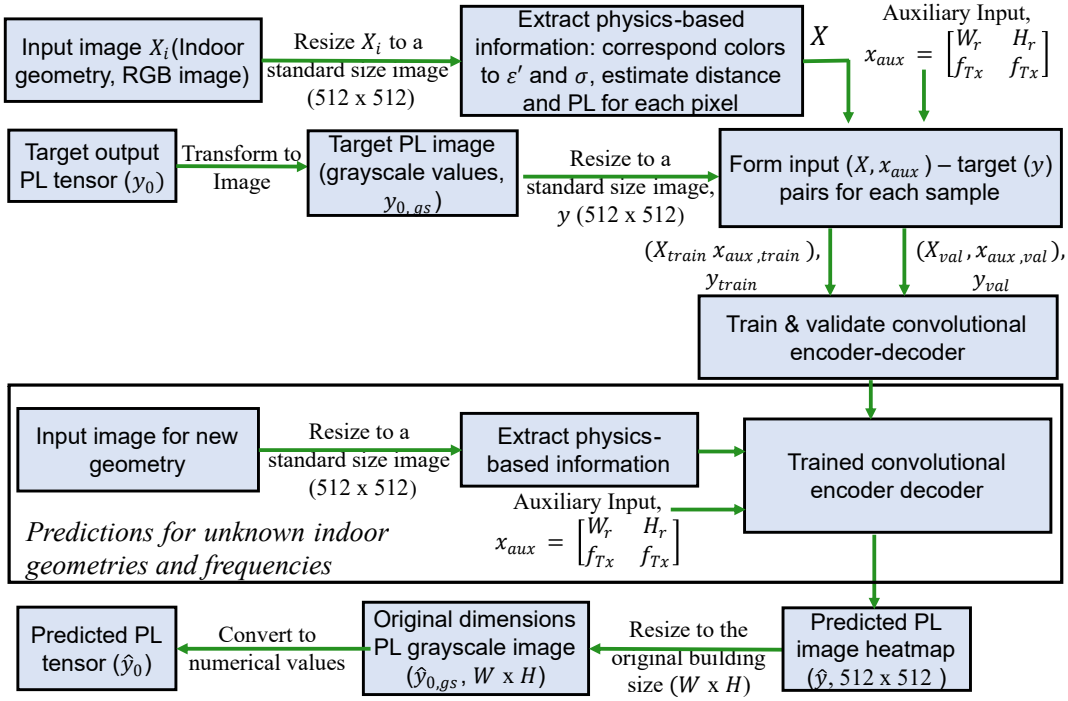
$$d_{i,j} = R\sqrt{W_r(x_{Tx} - i)^2 + H_r(y_{Tx} - j)^2}, \tag{4.1}$$

where R is the spatial resolution of the ray tracing simulation, (x_{Tx}, y_{Tx}) corresponds to the Tx position in the simulated grid, and $W_r, H_r = W, H/512$ are the width and height resizing ratios, respectively. Consequently, the values of the fourth channel are estimated as:

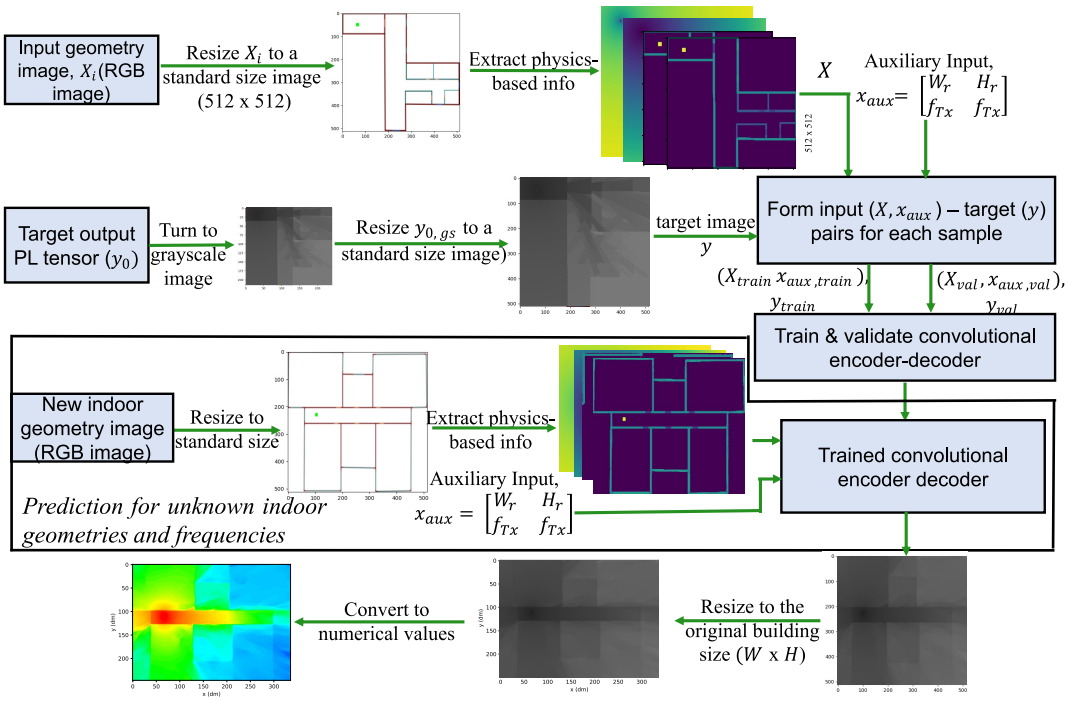
$$FSPL_{i,j} = 20log_{10}\left(\frac{4\pi d_{i,j} f_{Tx}}{c_o}\right), \tag{4.2}$$

where f_{Tx} is the operating frequency in Hz, and c_o is the speed of light.

A schematic representation of this process is shown in Figure 4.3b. As can be observed, the initial input blueprint is resized to a 512×512 image, and it is used to extract the four-channel tensor X, which is input to the convolutional encoder-decoder. Apart from the



(a) Block diagram of EM DeepRay architecture.



(b) Schematic representation of EM DeepRay functionality.

Figure 4.3: EM DeepRay architecture.

input tensor X, the convolutional encoder-decoder receives an auxiliary input, x_{aux}, which conveys information about the frequency, f_{Tx}, and the resizing ratios, W_r and H_r, of each sample. The details of the incorporation of the auxiliary input with the input tensor are outlined in Section 4.1.2.1. The target PL tensor is turned to a grayscale image, assuming values between 0 and 255, and the grayscale image is resized to be the same size as X, *i.e.*, 512×512. The physics-based information tensor along with the auxiliary input tensor form a pair of input values, $(X,\ x_{aux})$, while y is the target value. Each input-output pair accounts for a sample used to train the convolutional encoder-decoder.

Once trained, the convolutional encoder-decoder can be used to directly predict the PL for new geometries or frequency bands that are not included in the training data set (solid black line box in Figures 4.3a and 4.3b). Predictions for new geometries require only an image of the indoor environment, which is used to extract the physics-based input tensor, as shown in Figure 4.3b. The request for an image depicting the input geometry is not undue, since most existing commercial ray tracing software use such a representation [12, 107]. In addition, the proposed approach can be further facilitated by recent research which allows computer-aided design (CAD) floor plans to be readily turned into blueprints [110]. Alternatively, the physics-based input tensor can be defined manually by a user, specifying ϵ_r' and σ at each point of the grid, and then given the Tx position, estimate the distance and FSPL via (4.1) and (4.2), respectively. Furthermore, the extraction of the physics-based input tensor does not entail any complex operations, only matrix multiplications whose computation requires a few milliseconds.

Given the physics-based input tensor, the convolutional encoder-decoder predicts a PL heatmap image, $\hat{y}$, for the desired input geometry. The predictions of the convolutional encoder-decoder are pixel-wise and the size of the output tensor is 512×512. Thus, after a prediction, a bilinear interpolation is utilised to resize the output image back to the initial dimensions W and H of each geometry. As a final step, the grayscale PL images are converted to numerical values. The latter simply implies perceiving the elements of the resized input image in dB instead of pixel intensity values.

4.1.2.1 Convolutional encoder-decoder

A convolutional encoder-decoder is an fully convolutional network that consists of two basic components: (i) an encoder, used to downsample the input tensor and compress its context, and (ii) a decoder which is used to recapture the input tensor details and reconstruct it in the form of the target output tensor. Convolutional encoder-decoders may also include skip connections between the encoding and the decoding path, as shown in Figure 4.4 (dark yellow arrows), in order to retain information that might be lost during the encoding procedure. As a proof of concept, in this Section only one convolutional encoder-decoder is considered as a backbone for EM DeepRay, namely an SDU-Net [29] depicted in Figure 4.4. Section 4.2

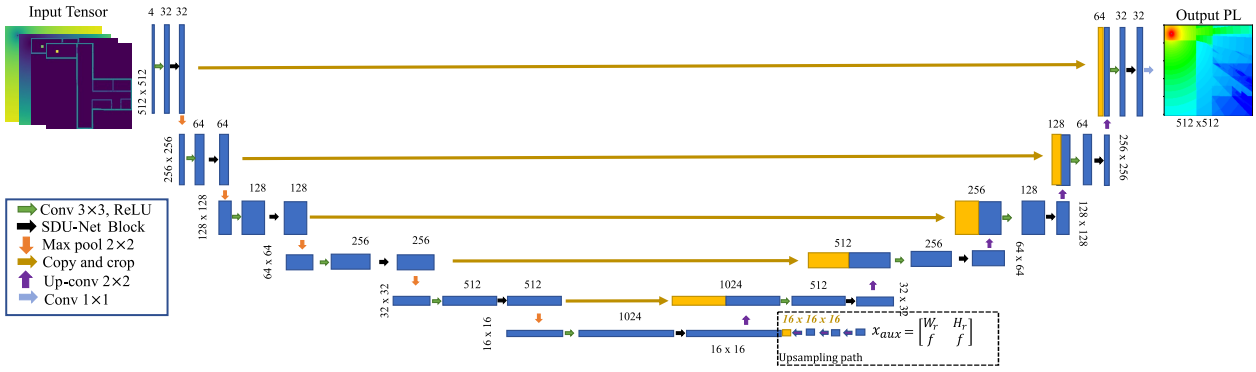
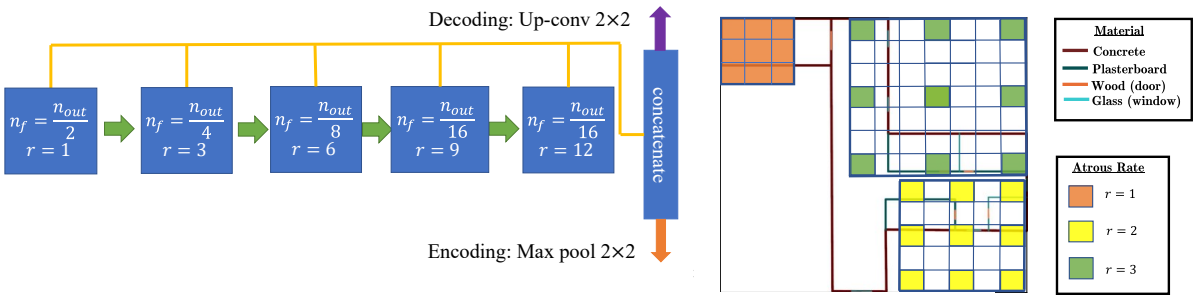


Figure 4.4: Convolutional encoder-decoder used in EM DeepRay to encode a physics-based input tensor and decode it as a PL heatmap.

will provide a comparative performance analysis considering another convolutional encoder-decoder [28] and a Trasformer-based model [111]. Finally, in [32] it was shown that simplistic approaches using MLPs, severely underperform, and thus they are not considered here.

The SDU-Net employed has five encoding and decoding layers, *i.e.*, five downsampling and upsampling operations are applied, denoted by orange and purple arrows at the encoding and decoding branches, respectively. As can be seen in Figure 4.4, the left path of the SDU-Net is used to encode and downsample the input tensor containing information about the indoor geometry, while the right path upsamples the encoded tensor and transforms it to a PL heatmap. At the end of the encoding path, an upsampling path is added aiming at upsampling the auxiliary input x_{aux}. Consecutive upsampling operations are applied to x_{aux} to increase its size and concatenate it with the encoded representation of the input tensor. This will allow the SDU-Net to understand the operating frequency and be aware of the resizing ratio for each sample. This information is not included in the input tensor for computational efficiency, *i.e.*, there is no need to reserve an entire channel for scalar variables. Instead, it is considered during the decoding by fusing the upsampled x_{aux} with the encoded input tensor just before the decoding procedure begins.

At the encoding path, each layer includes a 3×3 standard convolution, followed by the SDU-Net block, shown in Figure 4.5a, and a 2×2 max pooling with stride equal to 2. At the decoding branch, the feature map of the previous layer is first upsampled, and then it is concatenated with the feature map of the respective layer of the encoding branch (yellow boxes shown at each layer of the decoding branch). Consequently, a 3×3 standard convolution is applied to the concatenated tensor, and the resulting feature map is forwarded to the SDU-Net block, and moved upwards to the next layer. All the convolutional layers use a ReLu activation function, except for the last 1×1 convolution which uses a linear activation function. The number of convolutional filters is increased or decreased by a factor of two at each encoding or decoding layer, respectively. At the top layer of the decoding branch, a 1×1 convolution is used after the SDU-Net block to derive the PL heatmap image.



(a) Structure of the SDU-Net block [29].

(b) Example of convolutional kernels with a different atrous rate.

Figure 4.5: Details of the cascaded atrous convolutions used in the SDU-Net block and an example of convolutional kernels with different atrous rates, r, applied to the image of an indoor environment.

The number of the filters for the final 1×1 convolution is equal to the number of the desired output channels, *i.e.*, one.

The SDU-Net block, shown in Figure 4.5a, comprises five cascaded atrous convolutions, each one using a different number of filters and a different atrous rate, r. The feature maps of the different atrous convolutions are concatenated and forwarded to the next layer after being down- or upsampled. Atrous convolution, also known as dilated convolution, is a generalised case of the standard convolution operation. Its popularity has recently increased since it can be used to efficiently augment the receptive field of each neuron without increasing the number of model parameters [112]. In atrous convolution, the kernel is widened, and its sparsity is increased by introducing blank spaces between the kernel elements. The distance between neighbouring non-blank elements of the kernel is $r - 1$, where r is the atrous rate. The parameters for each atrous convolution are shown in Figure 4.5a, where r is the atrous rate, and n_{out} denotes the number of filters at the next layer (which is twice or half the number of filters at the current encoding or decoding layer, respectively).

An example of 3×3 atrous convolutions with different values of r applied to the image of an indoor environment is shown in Figure 4.5b. It can be observed that as r increases, the receptive field is augmented since the convolutional kernel has an enhanced field of view over the indoor geometry. Indeed, the area covered by the convolutional kernel with $r = 1$ (standard, non-dilated, convolution, orange box) is substantially smaller than that of the kernel with $r = 3$. That comes at no increase in the overall computational cost since the number of trainable parameters remains the same. Furthermore, the use of various atrous convolutions with different atrous rates enables the aggregation of information originating from different spatial scales and the identification of correlations between distant points. Moreover, due to the multiple layers of the SDU-Net architecture, it is also possible to capture multi-scale correlations at different resolutions (since the size of the input tensor is decreased by a factor of two at each encoding layer). The augmented receptive field, and the potential

of capturing multi-scale correlations (using various r values) at multiple resolutions (due to the multiple encoding and decoding layers), render the SDU-Net an exemplary convolutional encoder-decoder for radio propagation modelling tasks.

4.1.2.2 Evaluation metrics

To evaluate the performance of EM DeepRay four error metrics are used: the mean absolute error (MAE), the mean absolute percentage error (MAPE), the root mean square error (RMSE), and the Pearson correlation coefficient, r_p, defined as follows:

$$MAE = \frac{1}{NWH} \sum_{n=1}^{N} \sum_{i=1}^{W} \sum_{j=1}^{H} \left| y_{o,(n)}(\mathrm{i,j}) - \hat{y}_{o,(n)}(\mathrm{i,j}) \right|, \tag{4.3}$$

$$MAPE = \frac{1}{NWH} \sum_{n=1}^{N} \sum_{i=1}^{W} \sum_{j=1}^{H} \frac{\left| y_{o,(n)}(\mathrm{i,j}) - \hat{y}_{o,(n)}(\mathrm{i,j}) \right|}{\left| y_{o,(n)}(\mathrm{i,j}) \right|}, \tag{4.4}$$

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N} \sum_{i=1}^{W} \sum_{j=1}^{H} \left(y_{o,(n)}(\mathrm{i,j}) - \hat{y}_{o,(n)}(\mathrm{i,j}) \right)^2}{NWH}}, \tag{4.5}$$

$$r_p = \frac{\sum_{n,i,j}^{N,W,H} \left(y_{o,(n)}(\mathrm{i,j}) - \overline{y}_{o,(n)} \right) \left(\hat{y}_{o,(n)}(\mathrm{i,j}) - \overline{\hat{y}}_{o,(n)} \right)}{\sqrt{\sum_{n,i,j}^{N,W,H} \left(y_{o,(n)}(\mathrm{i,j}) - \overline{y}_{o,(n)} \right)^2 \sum_{n,i,j}^{N,W,H} \left(\hat{y}_{o,(n)}(\mathrm{i,j}) - \overline{\hat{y}}_{o,(n)} \right)^2}}, \tag{4.6}$$

where $y_{o,(n)}(\mathrm{i,j})$ and $\hat{y}_{o,(n)}(\mathrm{i,j})$ are the ray tracing and the predicted PL values for the n-th sample at the spatial point (i, j), respectively, $\overline{y}_{o,(n)}$ and $\overline{\hat{y}}_{o,(n)}$ are the corresponding mean values for that sample, and N represents the number of samples. Note that the error is measured at the end of the proposed framework, *i.e.*, it is not pixel-wise but is estimated with respect to the actual PL values.

4.1.3 Comparison with simulated data

This section presents the implementation of the EM DeepRay framework discussed in 4.1.2, evincing that is possible to transform physics-based information to the PL distribution, and providing a comparison between the PL inferred by EM DeepRay and that simulated via a ray-tracer. To create the synthetic data to train EM DeepRay the Ranplan Professional ray tracing simulator is used, which is developed by Ranplan Wireless [107], and it is a robust radio propagation engine, supporting radio propagation simulations for both indoor and outdoor environments. Ranplan Professional enables the easy and efficient design of entire floors or even buildings and it supports a large number of different communication technologies (WiFi, 5G New Radio, IoT). The software has been widely used for actual network planning,

which provides assurance regarding its reliability. To accelerate the calculation of the signal strength, the ray-tracer filters out the incoherent addition of the electric fields carried by the rays. Hence, the fast fading effect is mitigated, which results in the absence of abrupt variations of the signal level for close-range spatial points, as can be seen from the PL heatmaps in 4.6a-4.9a.

To train and validate EM DeepRay ray tracing simulations are conducted in 20 different indoor environments. The buildings considered include simple and more complicated indoor environments, with more than 20 sub-rooms in the same building floor, and the walls are made of various construction materials (concrete, brick, plasterboard, wood, and glass) and have various thicknesses. For each building, Txs operating at the following 3 different frequencies are considered: (i) 0.433, (ii) 2, and (iii) 3.7 GHz, which correspond to frequency bands utilised in IoT and 5G systems. To create the data set, each time it is assumed that a sole Tx operating at a given frequency is placed within a building. Then, a ray-tracing simulation is conducted only for this Tx. Once the ray-tracing simulation is finished, the Tx position and/or the frequency are changed, and a new ray-tracing simulation is run in order to create a new sample. This corresponds to consecutively sampling the PL response in an environment of interest for a given Tx position and frequency.

For all the simulations the transmitter antenna is $z_{Tx} = 1.5$ m above the floor and has an isotropic beam pattern; a more general problem formulation to remove this constraint will be presented in Section 4.2. The resolution of the ray tracing simulations is set to 0.1, *i.e.*, the PL is sampled once per 0.1 m. For each frequency band, approximately 35 different ray tracing simulations are conducted at each building, *i.e.*, 35×3 samples per building, assuming a different Tx position within the indoor geometry for each sample. From the entire volume of data, 80% of the samples is used to train EM DeepRay, whereas the remaining 20% is utilised to validate its performance (this is a commonly used data splitting ratio in ML problems [61]). To test EM DeepRay three cases are considered, aiming to explore how well the data-driven model: (1) can generalise to new geometries, not included in the training data set, (2) can infer the PL for frequency bands other than those of the training set, and (3) behaves in a combination of (1) and (2).

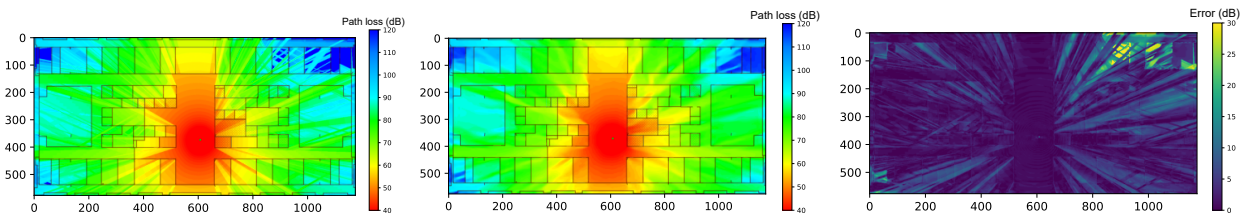
In what follows, these cases are referred to as Test 1, 2, and 3, respectively, and are summarised in Table 4.1, where the terms "known" and "unknown" indicate either included or not included in the training data set. To implement Test 1, five new indoor environments are considered, generating about 35 samples at each building, for each frequency band of the training data set (0.433, 2, and 3.7 GHz). EM DeepRay has no prior information regarding the PL distribution within these indoor environments, and it will infer the PL heatmaps based on the weights estimated during the training phase. For Test 2, approximately 30 Txs operating at 866 MHz and 2.6 GHz are randomly placed at each of the 20 buildings used to train and validate EM DeepRay. Finally, for Test 3, Txs operating at 866 MHz and 2.6

GHz are deployed at the five buildings used in Test 1, *i.e.*, for buildings and a frequency band not included in the training data set, taking approximately 40 samples per building and frequency.

Note that the way the test sets are created slightly differentiates from the common approach where the training, test, and validation dataset should come from the same distribution. For instance, some material parameters, *e.g.*, conductivity, encountered in 866 MHz and 2.6 GHz (testing frequencies), might be different than those seen in the training phase. Likewise, the building geometries differ from those seen during training. In a sense, the test sets also comprise *out-of-distribution* data representing propagation environments that deviate from those encountered during training. The reasoning underpinning this strategic choice is that this way one can assess the potential of the data-driven model to replace a legacy solver without having to train the model with all the possible material parameters and wireless network configuration combinations.

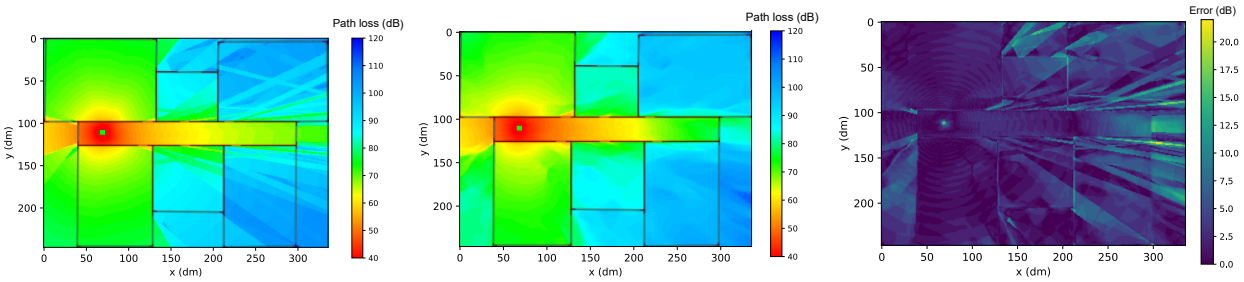
EM DeepRay is trained over two NVIDIA TITAN RTX GPUs in Tensorflow, using the Adam optimisation algorithm for 250 epochs, with the learning rate set to 0.0005, and a batch size of 4. Before passing the input data to it, a min-max normalisation is applied separately at every input channel [61]. The loss function to be optimised is the RMSE. During training, the loss function is minimised with respect to y and $\hat{y}$, *i.e.*, based on the pixel intensity values of the ray tracing and the predicted PL images. However, the error metrics are estimated with respect to the ray tracing and the predicted actual PL values (*i.e.*, between y_o and $\hat{y_o}$, in dB). The training takes almost 2 hours, while a prediction for a single sample requires about 100 milliseconds, including the extraction of the physics-based input tensor.

The overall error metrics between the simulated and the predicted PL values for all the case studies are presented in Table 4.1. The error metrics are negligible for the training set, with the RMSE and the MAE being equal to 1.2 and 0.99 dB, respectively. This implies that the proposed framework can indeed learn to translate a map of physics-based information to a PL heatmap. The validation set error metrics are approximately 3 dB larger than these of the training set, and they still assume low values. Test 1 and 2 yield approximately an



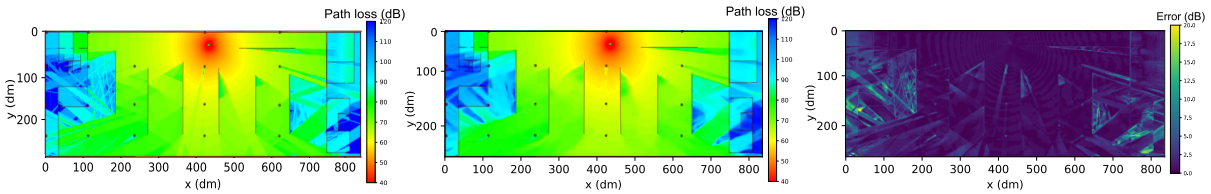
(a) Ray tracing. (b) EM DeepRay. (c) Error Map.

Figure 4.6: Comparison between the PL heatmap simulated through (a) ray tracing and predicted by (b) EM DeepRay for a validation set sample at 433 MHz (known building and frequency, but unknown Tx position). The (c) error map depicts the absolute error between (a) and (b).



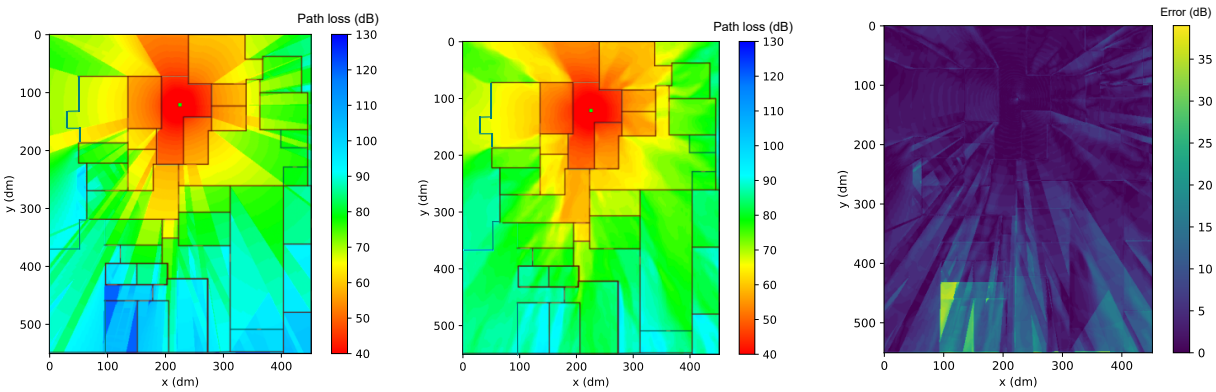
(a) Ray tracing. (b) EM DeepRay. (c) Error Map.

Figure 4.7: Comparison between the PL heatmap simulated through (a) ray tracing and predicted by (b) EM DeepRay for a Test set 1 sample at 2 GHz (known frequency band, but unknown indoor geometry). The error map (c) depicts the absolute error between (a) and (b).



(a) Ray tracing. (b) EM DeepRay. (c) Error Map.

Figure 4.8: Comparison between the PL heatmap simulated through (a) ray tracing and predicted by (b) EM DeepRay for a Test set 2 sample at 2.6 GHz (known indoor geometry, but unknown frequency band). The error map (c) depicts the absolute error between (a) and (b).



(a) Ray tracing. (b) EM DeepRay. (c) Error Map.

Figure 4.9: Comparison between the PL heatmap simulated through (a) ray tracing and predicted by (b) EM DeepRay for a Test set 3 sample at 868 MHz (unknown indoor geometry and frequency band). The error map (c) depicts the absolute error between (a) and (b).

RMSE of 5 dB and an MAE of 3.8 dB, while for Test 3 the respective values are 5.6 and 4.5 dB. More importantly, the good resemblance between the simulated and the predicted data is also preserved for the three different test cases, as is indicated by the values of r_p that are very close to 1, and the MAPE which assumes values below 0.1 for every scenario. Thus, in addition to learning to infer the signal attenuation within a given geometry and frequency

Table 4.1: EM DeepRay performance for each case study; known and unknown indicates included or not included in the training data set.

	Tx Position	Geometry	Frequency	RMSE (dB)	MAE (dB)	MAPE	r_p	Samples
Training	Known	Known	Known	1.2	0.99	0.01	0.97	1680
Validation	**Unknown**	Known	Known	4.21	3.26	0.03	0.937	420
Test 1	**Unknown**	**Unknown**	Known	5.05	3.86	0.04	0.94	525
Test 2	**Unknown**	Known	**Unknown**	4.77	3.63	0.04	0.95	600
Test 3	**Unknown**	**Unknown**	**Unknown**	5.85	4.63	0.05	0.94	400

band, the proposed data-driven model is generalisable, and it can provide accurate estimates of the PL for new indoor geometries and frequencies. By exploring the generalisability of EM DeepRay to buildings and frequencies separately, it is easier to understand its potential limitations, and to where these could be attributed. Also, in Test 3, by studying the same buildings and frequency bands as in Test 1 & 2, one can be confident that any error is not owing to the use of new unknown buildings or frequencies but can be attributed to the simultaneous generalisation over new buildings and frequencies.

Figures 4.6, 4.7, 4.8, and 4.9 visualise the results for a random sample from the validation set and from the test sets 1, 2, and 3, respectively. Figures 4.6a-4.9a show the simulated ray tracing "ground-truth" for each sample, while the corresponding PL heatmap predicted by EM DeepRay is illustrated in Figures 4.6b-4.9b. Note that DeepRay outputs solely the PL heatmap, and in order to illustrate the input geometry its blueprint is consequently imposed on top of the PL heatmap. As can be observed, the similarity between the simulated and the predicted PL heatmaps is very high for all the cases. The absolute error maps in Figures 4.6c-4.9c depict the absolute error, $|y_o(\text{i,j}) - \hat{y_o}(\text{i,j})|$, at each spatial point (i, j) within the simulated indoor environment. A common trend is that the predicted PL exhibits the largest absolute errors at positions far away from the transmitting device. For instance, as can be seen in Figure 4.9c, the largest errors are found in the lower left area of the floor. That is more than 40 and 10 m away from the transmitter, in the horizontal and vertical directions, respectively. This is encouraging, since when a user is located far away from a transmitter it is likely that they are served by another device, and thus these areas do not have a significant impact on network design decisions.

It should be mentioned, that EM DeepRay enables the inference of the PL distribution for a single Tx based on an input tensor tailored to that Tx, *i.e.*, the distance and FSPL are computed according to the position of this Tx only. Thus, similarly to ray tracing, in the case of multiple Txs one should conduct a distinct simulation for each Tx. Then, if the transmitting devices operate at the same frequency, the received signal at each spatial point can be evaluated by selecting the strongest signal from all the Txs and considering the rest as interference.

4.2 Generalisation to 3D and Antenna Radiation Pattern

The EM DeepRay framework presented in Section 4.1 is subject to two important limitations: (i) it assumes Txs with only isotropic antennas, *i.e.*, radiating with the same intensity in all directions, neglecting the impact of the antenna directivity on wave propagation, and (ii) it assumes that the Tx and the Rx are found at the same receiving plane. In fact, these limitations are closely related, as signal propagation over the various receiving planes in the 3D space is substantially affected by the antenna radiation pattern. Apparently, the structure of the input tensor described in Section 4.1.1 does not carry any information related to the antenna radiation pattern, and thus cannot be exploited to predict the PL distribution for directive antennas; however, such cases are of high relevance in the simulation and design of indoor wireless communication networks.

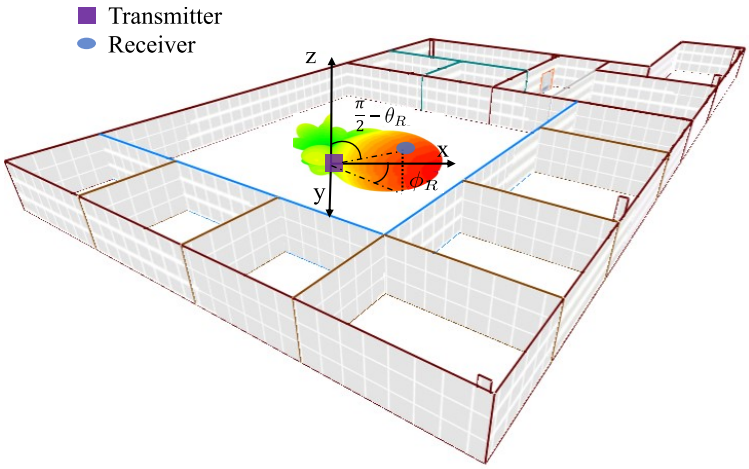


Figure 4.10: Example of the angles between the Tx and a Rx.

To overcome these limitations and allow EM DeepRay to generalise over different radiation patterns, the form of the fourth channel is modified in the following way; let $G(\phi, \theta)$ be the function that indicates the $3D$ antenna gain for the azimuth and elevation angles ϕ and θ, respectively. Then, as shown in Fig 4.10, assuming that the Tx is located at the point $(x_{T_x}, y_{T_x}, z_{T_x})$, one can compute the azimuth and the elevation angles between the Tx and a point $(x_{R_x}, y_{R_x}, z_{R_x})$ at the target receiving plane, as follows:

$$\phi_R = Arctan(\frac{y_{R_x} - y_{T_x}}{x_{R_x} - x_{T_x}}), \tag{4.7}$$

$$\theta_R = Arcsin(\frac{z_{R_x} - z_{T_x}}{d}), \tag{4.8}$$

where $d = \sqrt{(x_{T_x} - x_{R_x})^2 + (y_{T_x} - y_{R_x})^2 + (z_{T_x} - z_{R_x})^2}$ is now the 3D distance between the Tx and the Rx (note that in order to make the visualization more comprehensible the Tx in Figure 4.10 is placed at the origin of the coordinate system, but this is not necessarily

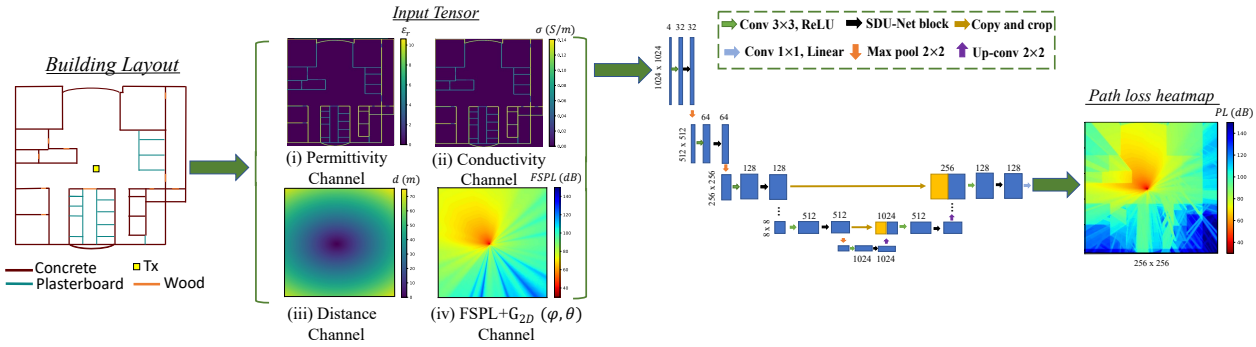


Figure 4.11: Schematic of EM DeepRay, using an asymmetric SDU-Net with more encoding than decoding layers, where the fourth channel is modified to consider the impact of the antenna radiation pattern.

the case, *i.e.*, x_{T_x}, y_{T_x}, and z_{T_x} are non-zero). Consequently, this process can be generalised to all the points (i, j) appertaining to the receiving plane z_{R_x}, and the matrices $\mathbf{\Phi_{2D}}$ and $\mathbf{\Theta_{2D}}$ can be computed, which give the azimuth and elevation angles, respectively, for all the points of the receiving plane. Then, a cut of the antenna gain over the 2D dimensional receiving plane can be straightforwardly computed as $\mathbf{G_{2D}} = G(\mathbf{\Phi_{2D}}, \mathbf{\Theta_{2D}})$, which provides an approximation for the impact of the antenna directivity over the receiving plane of interest. With $\mathbf{G_{2D}}$ estimated, in order to make EM DeepRay directive-aware, the fourth channel is modified and set equal to:

$$FSPL + \mathbf{G_{2D}} = 20log_{10}\left(\frac{4\pi \mathbf{D} f_{Tx}}{c_o}\right) + G(\mathbf{\Phi_{2D}}, \mathbf{\Theta_{2D}}), \tag{4.9}$$

where $\mathbf{D}$ is a matrix including the distance between the Tx and each point at the receiving plane, and as mentioned earlier f_{Tx} is the operating frequency in Hz, and c_o is the speed of light. An example of the new fourth channel form can be seen in the example input tensor (iv), shown in Figure 4.11. This modification will allow the data-driven model to comprehend the effect of the antenna radiation pattern on radio wave propagation and will help find a proper mapping between the material properties, physical distance, frequency, antenna directivity, and the PL distribution space. This formulation does not necessitate including implicit information related to the Rx height and the angle difference explicitly in the input tensor, rather, this information is implicitly contained in the fourth channel. Finally, it should be noted that if the building structure is different at different receiving planes then the first and the second channels of the input tensor are modified accordingly. In particular, if a wall does not reach the ceiling or an object's height is less than that of the receiving plane then the values of the permittivity and channels will be 1 and 0 (air), respectively, while when the object's height is larger than the receiving plane height, these channels will comprise the permittivity and conductivity value of the material. It should be mentioned that this modelling approach of the 3D propagation environment and the antenna radiation pattern entails some simplifications and inevitably cannot capture all relevant details. For instance,

the model does not have insight into the 3D indoor environment structure, and likewise, it observes only a cut of the 3D antenna radiation pattern. Consequently, the model may not be able to perceive some more complicated phenomena, *e.g.*, diffractions from objects found at a lower height than the receiving plane z_{Rx}. To account for such detail it would be necessary to provide distinct 3D tensors for both the indoor environment and antenna pattern, but that would increase substantially the computational cost. Hence, the data-driven framework of this Section enables ad interim the generalisation in different receiving planes through some simplified assumptions that can be removed in future work.

4.2.1 Modification in EM DeepRay Architecture

To exploit a more fine-grained representation of the indoor environment while evaluating the PL distribution at a suitable resolution, two modifications are introduced in the DL module used in EM DeepRay. First, unlike Section 4.1, where the output tensor and the input tensor had the same size, specifically, 512×512, in what follows the size of the input tensor and output tensor is 1024×1024 and 256×256, respectively. Second, the SDU-Net is made asymmetric (ASDU-Net), as shown in Figure 4.11, having more encoding than decoding layers. Specifically, the number of encoding and decoding layers used is 7 and 5, respectively. As mentioned, this allows the use of a larger size input tensor, thus having a more detailed representation of the indoor geometry, while capturing the PL distribution with sufficient granularity. For instance, for a building with dimensions 100×100 m, that is equivalent to probing the input features almost every 0.1 m and sampling PL approximately every 0.5 m.

4.2.2 Augmented Data Set and Assessing Generalisation

To allow EM DeepRay to generalise over 3D and forge a high-fidelity full-fledged data-driven propagation model that can effectively replace legacy simulators, a substantially larger volume of data is generated compared to that of Section 4.1.3. In particular, this time 110 diverse, in terms of size, complexity, and construction material type (concrete, plasterboard, wood, glass, metal), indoor environments are generated out of which 80 are used to train EM DeepRay and 30 to test its performance. For each geometry, approximately 40 samples are generated by randomly changing the position and the height of the transmitting device within the geometry, and evaluating the PL distribution at 5 different receiving planes, *i.e.*, $z_{R_x} = 0.5, 1, 1.5, 2.0$ and 2.5 m above the floor. The maximum permitted PL value was set at a threshold of 150 dB, with any values exceeding this limit set to 150 dB as well. For each position 16 different radiation patterns are considered (including the case of an isotropic antenna), randomly probing their steering direction, utilising 11 of them during training and reserving 5 to explore how well the data-driven model can generalise to unknown ones. This process is repeated for 5 different frequencies, namely 2, 2.4 3.7, 4.9, 26, and 28 GHz, which

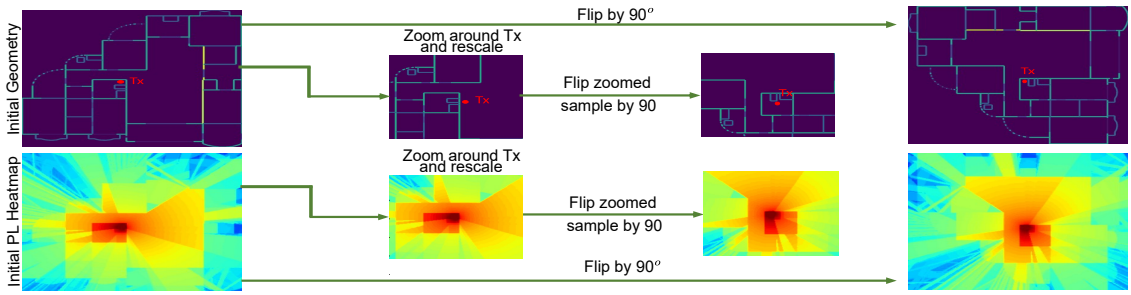


Figure 4.12: The Data augmentation process followed to increase the number of training samples and thus enhance the model's performance. The initial input and target tensors are flipped by $90^o, 180^o$, and 270°, while an MST strategy is followed focusing on a box area of 200, 400, 600, and 800 pixels around the Tx, and consequently flipping the zoomed samples to create additional samples.

correspond to sub-6GHz and millimetre wave frequency bands used in 5G systems. For all the simulations up to 8 ray reflections, 8 ray transmissions, and 2 diffractions are taken into account.

Two data augmentation techniques are employed to further enrich the field of observation at which EM DeepRay is exposed, and so improve its performance. The first data augmentation approach is a multiscale training (MST) scheme that aims at providing the model insight related to the PL distribution in scenarios with higher granularity. To this end, for each one of the samples used to train the model, additional samples are extracted by zooming in on the area around the TX with multiple zooming ratios. Consequently, for the input and the target tensors of each sample from the data set, a rectangular area centred around the Tx is cropped, assuming rectangles of different sizes. The rectangle sizes are set to 1/5, 2/5, 3/4, and 4/5 of the tensor size, effectively capturing the effect of zooming in with different zoom ratios (note that if the rectangle side exceeds the simulation grid the "zoomed" sample is extended until the grid boundary). Then, these new samples are upsampled to the input and output tensor sizes mentioned in Section 4.2.1 in order to be used in the training process. This procedure not only synthetically increases the number of training samples, but more importantly, transfuses to the data-driven model the potential to make accurate predictions for sites with different granularity and rich indoor geometry representation. An example of a "zoomed" sample can be seen in Fig. 4.12, presenting the geometry and PL cut obtained when considering a rectangular box with a size equal to 400 pixels around the Tx (note that the initial and the "zoomed" samples actually have the same size, *i.e.*, 1024×1024 and 256×256 for the input and output tensor, respectively, but for space efficiency, the latter is shown smaller). The second data augmentation approach enables the prediction of EM DeepRay to remain unchanged under rotations, which is a fundamental property of Maxwell's equations [113]. By flipping the input and target tensors of the initial and "zoomed" samples by $90^o, 180^o$, and 270°, as shown in Figure 4.12, the model can deduce that as long the propagation environment remains unchanged, rotations will not impact radio

Table 4.2: Test set error metrics for different DL backbones.

Models	MAE (dB)	RMSE (dB)	MAPE	r	Params (millions)
ASDU-Net w/o MST	3.19	4.70	0.02	0.92	115
AU-Net w/o MST	3.66	5.11	0.03	0.92	351
T5 SegFormer w/o MST	4.05	5.43	0.04	0.89	85

wave propagation.

Through an automated pipeline, the ground-truth data generation process lasted approximately two weeks in a workstation equipped with an Intel Core i9-9960X X-series Processor with 16 central processing units and two 24 GB Nvidia Titan RTX graphics processing units. By the end of the data generation process, there are slightly in excess of 500,000 training samples, 25,000 validation samples (utilized to select the best-trained model), and approximately 125,000 additional samples used to test the generalisability of the model to new geometries, radiation patterns, and frequency bands, not seen during the training phase. The number of training data is synthetically increased via the discussed data augmentation approaches, ultimately exceeding 10 million.

The training setup and the evaluation metrics remain the same as those discussed in Section 4.1.3. For a fair evaluation, the performance of other DL backbones is explored in the implementation of EM DeepRay and an ablation study is conducted using different data augmentation techniques. Specifically, an asymmetric U-Net (AU-Net) and a T5 SegFormer [111] are considered as alternatives backbones. In AU-Net, each encoding and decoding layer comprises two standard convolutions followed by a downsampling and upsampling operation, respectively, and similarly to the ASDU-Net, 7 encoding and 5 decoding layers are used. Furthermore, to compensate for the limited receptive field of the 3×3 convolutions used in the vanilla U-Net, the size of the convolutional kernels used in the AU-Net is 5×5 (a smaller

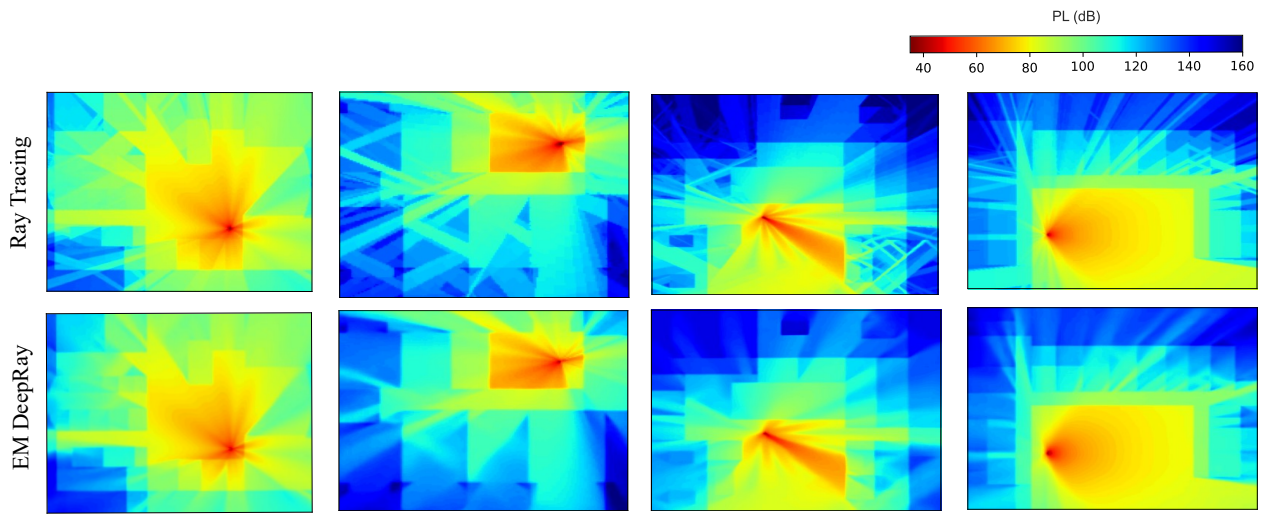


Figure 4.13: Simulated via ray tracing and predicted by EM DeepRay PL heatmaps for samples with directive antennas.

Table 4.3: Test set error metrics with different data augmentation techniques using ASDU-Net as a backbone.

Models	MAE (dB)	RMSE (dB)	MAPE	r	Params (millions)
ASDU-Net with MST-F	2.85	4.29	0.01	0.94	115
ASDU-Net w/o MST	3.19	4.70	0.02	0.92	115
ASDU-Net w/o MST-F	3.96	5.61	0.03	0.91	115

kernel size would result in a substantially degraded accuracy). The configuration details of the T5 SegFormer, *e.g.*, encoding blocks, layers per block, or attention heads at each layer, can be found in [111]. The comparative evaluation for the 3 models without using MST is summarized in Table 4.2. It can be seen that ASDU-Net attains the best performance for all metrics. The number of trainable parameters is slightly higher compared to that of the T5 SegFormer and is approximately three times smaller than that of AU-Net. It should be noted that the error metrics were lower for the 26 GHz band, *e.g.*, MAE was 4.3 and 2.2 for 2.4 and 26 GHz, respectively, due to the larger number of points reaching the threshold value (150 dB) that was easier for the model to predict .

To avoid prolonged training times, in the previous experiments the three models were trained without MST. Then, the impact of different data augmentation techniques is probed only for the best model, *i.e.*, ASDU-Net. Three cases are investigated in these experiments. Specifically, first, the case where both MST and flipping are employed is considered, denoted as MST-F. Then, the case where MST is not used, denoted as w/o MST. In the last situation explored, neither MST nor flipping is used, denoted as w/o MST-F. The first case, *i.e.*, ASDU-Net with MST-F, constitutes the final implementation of EM DeepRay. Each training epoch over the full data set takes almost 2 days, and the model converges approximately at $8 - 10$ iterations, whereas for the non-MST data set each epoch requires 5 hours. Notably, the training time is a one-time cost that occurs only once during the training phase of the data-driven model and does not impact the inference stage. From the results presented in Table 4.3, it is evident that data augmentation can improve the accuracy of the model. Indeed, the values of all error metrics deteriorate without MST, and further worsen if neither MST nor flipping is employed. The model trained both with MST and flipping exhibits the smallest error, and some indicative PL heatmaps can be seen in Fig. 4.13, demonstrating a very good alignment between the predicted and the simulated ray tracing spatial PL distribution for directional antennas.

Ultimately, it can be seen that EM DeepRay can learn to correlate the PL distribution with the input information, and more importantly, the antenna radiation pattern which constitutes the new feature. It is crucial to remark that once EM DeepRay is trained, it can be used as a standalone propagation solver, *i.e.*, to make predictions for new scenarios with unknown parameters, and there is no need to train from scratch each time. The inference time

for a single simulation is about 500 milliseconds, including the extraction of the physics-based input tensor. Only approximately 10 ms are required for the transformation of the input tensor, whilst the remaining time is related to calculations entailed for the input tensor construction, *e.g.*, estimation of angles.

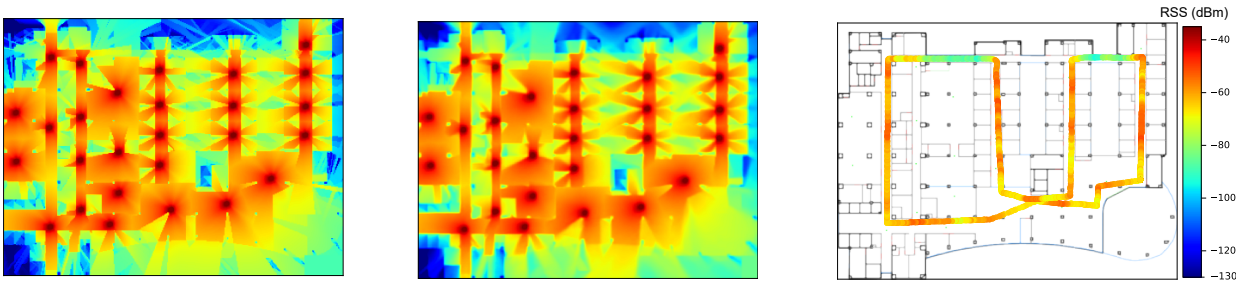
4.2.3 Evaluation with Measurements in Real World-Deployments

The results of the previous section verify that EM DeepRay can successfully generalise to previously unseen use cases, in terms of building layout, the antenna radiation pattern, or the frequency band. Therefore, the performance of EM DeepRay is now evaluated with respect to real-world measured data, depicting the RSS in operating indoor wireless networks, and benchmarked with the ray tracing results. In particular, two scenarios are considered: a hospital and a large workspace environment, both having a complicated structure and including a large number of rooms. The dimensions of the hospital are 86.5×65.5 m, whilst these of the workspace are 112.0×112.0 m. The two geometries and the RSS measurements taken along them can be seen in Figures 4.14c, 4.15c, respectively.

Table 4.4: Comparison with Measured Data.

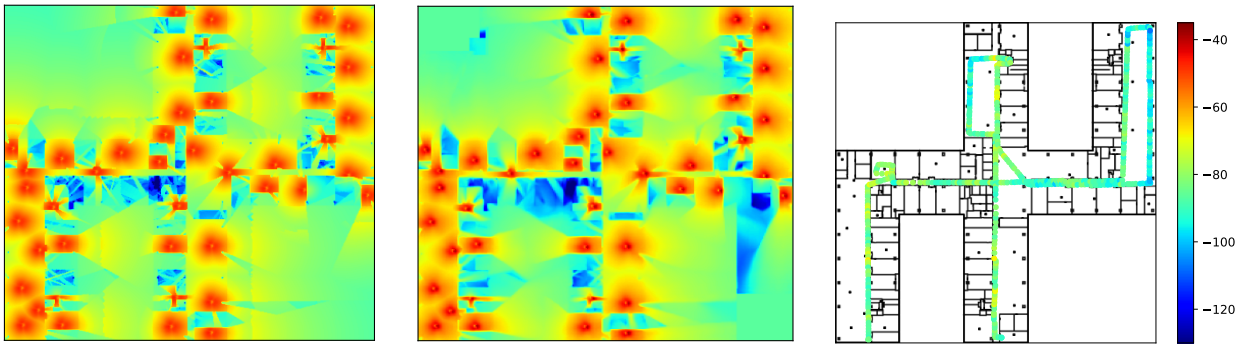
	Hospital		**Workspace**	
	Ray Tracing	EM DeepRay	Ray Tracing	EM DeepRay
RMSE	5.99	5.94	9.19	9.12
MAE	5.05	4.90	7.43	7.36
Computational time (sec)	2334.78	19.67	5509.07	27.75

In the hospital, there are 27 Txs placed at different locations at 2.7 m above the ground and operating at 2.6 GHz, whereas, for the workspace 44 Txs are considered at 2.4 m above the ground transmitting at 2.4 GHz. A distributed antenna system (DAS) is used hence there is no interference between the Txs. Neither the geometries nor the frequency bands have been seen by EM DeepRay during the training phase. The error metrics for the ray-tracer and EM DeepRay for the two scenarios are shown in Table 4.4. As can be observed both the ray-tracer



(a) Ray Tracing. (b) EM DeepRay. (c) Measured RSS.

Figure 4.14: Hospital heatmaps for ray tracing and EM DeepRay and measured RSS.



(a) Ray Tracing. (b) EM DeepRay. (c) Measured RSS.

Figure 4.15: Workspace heatmaps for ray tracing and EM DeepRay and measured RSS.

and the data-driven model attain the same error levels with respect to the measured RSS, *i.e.*, the student (EM DeepRay) has successfully distilled the knowledge of the teacher (ray-tracer), and can now perform at the same level. That can be also corroborated by the heatmaps of Figures 4.14a and 4.14b for the hospital, and 4.15a to 4.15b for the workspace, which exhibit high similarity. Indeed, as can be seen, EM DeepRay can capture precisely the rays emitting out of doors or windows, as well as the shadowing due to obstructions. For the hospital, ray tracing and EM DeepRay attain an MAE between 5 and 6 dBm with respect to the measured RSS, while for the workspace the respective errors are 2-3 dBm larger. The distinct advantage of the emulation via data-driven propagation models though stems from the sharp decrease in the time required to conduct the simulations. When tracing all the rays up to the predefined number of intersections, the ray-tracer requires approximately 38, and 91 minutes for each scenario, respectively. On the other hand, EM DeepRay can provide similar quality results in 19, and 27 seconds, respectively, which illustrates the dramatic decrease in the simulation time.

4.3 Calibration via Transfer Learning

In the previous sections, it was shown that EM DeepRay can be trained with simulated data from various indoor wireless communication system simulations and eventually learn to predict the PL distribution for them. Once trained, EM DeepRay can be employed as a full-fledged propagation model to furnish estimates of the PL for an arbitrary indoor geometry, radiation pattern, and frequency. The credibility of the data-driven model highly depends on the data used to develop it. These data are only an approximation of the actual signal attenuation and are themselves subject to errors. However, the purpose of an ML-based propagation model is to deliver results that closely resemble actual rather than simulated data. Indeed, as evinced in Section 4.2, despite constituting a very fast surrogate, the accuracy of EM DeepRay is innately capped by the performance of the ray-tracer, *i.e.*, the data-driven model mimics the PL distribution of actual propagation solver but it cannot compensate for

its inaccuracies.

In this section, this issue is addressed by outlining a simple, yet efficient, approach that allows the calibration of EM DeepRay through transfer learning. The aim of transfer learning is to leverage knowledge accrued to tackle a certain problem, and employ it to solve a different problem, which is related back to the initial problem [61]. In this case, the first problem is the design of an ML-based propagation model that can predict the PL in indoor environments, whilst the second problem is to make the ML-based propagation model realistic. The motivation behind the use of transfer learning, rather than training EM DeepRay simultaneously with synthetic and measured data, is that measured data are scarce and sparse. Indeed, measurement campaigns are time-consuming and expensive, and usually, the results from large-scale campaigns are not publicly available. In addition, typically only a small number of measured RSS values is available for each indoor environment since these campaigns are realised by moving the receiving device around within the area of interest and recording the RSS at some sparse points.

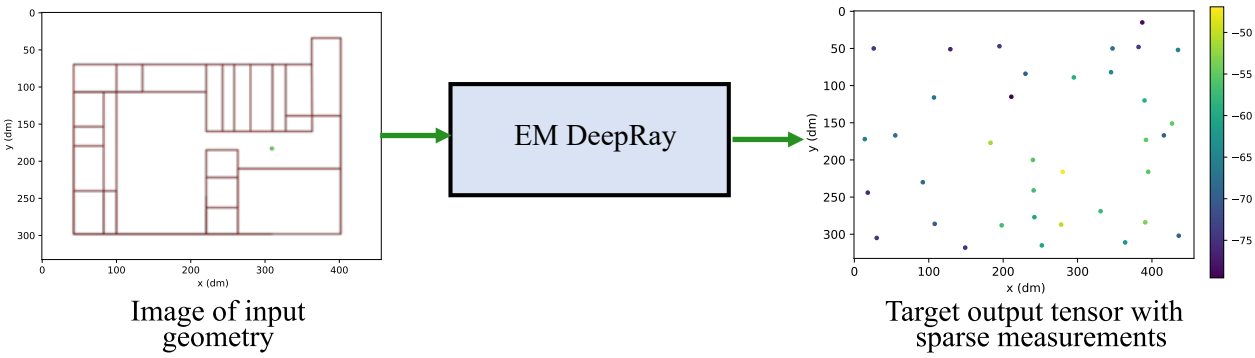


Figure 4.16: Calibration framework; the dots in the target tensor correspond to points in the indoor geometry at which the RSS is measured, and their colour represents the RSS intensity.

Hence, if a data set with both simulated and measured data is used to train an ML propagation model, it would comprise a limited number of measurement scenarios (compared to the simulated ones) with only a few recordings per scenario. That would render the training challenging, as it would be necessary to find a balance between the importance of the measured over the simulated data. Instead, treating the calibration as a distinct problem enables handling a limited quantity of sparse data in a more efficient manner. In particular, to calibrate EM DeepRay the same framework shown in Fig. 4.3a is used, however, the target output tensor is now different. As shown in Fig. 4.16, for the calibration the target output tensor comprises the recorded measurements at some points within the simulated grid, instead of the simulated ray tracing PL values. To distinguish the target output tensors of the calibration framework the subscript c is used, referring to the initial and the resized target output as $y_{c,0}$ and y_c, respectively. Thus, the calibration process constitutes a retraining of the pre-trained model obtained in Section 4.1.3 to adapt its PL predictions to match the measured PL values of y_c.

A significant difference between the target output tensors of the frameworks, shown in Figs. 4.3a and 4.16, is that for most elements of $y_{c,0}$ and y_c there are no recorded measurements (for convenience these empty grid point values are filled with zeroes.) For that reason, it is not possible to use the same loss function as the one used in Section 4.1.3. More specifically, the operation of EM DeepRay is equivalent to applying a mapping function f, parameterised through a set of learnable weighs Θ, to the initial input tensor X in order to derive the output PL image heatmap, *i.e.*, $\hat{y} = f(X|\Theta)$. The values of Θ are computed iteratively, such as to minimise the difference between the pixel intensity of the ray tracing "ground truth" and the predicted PL image heatmap:

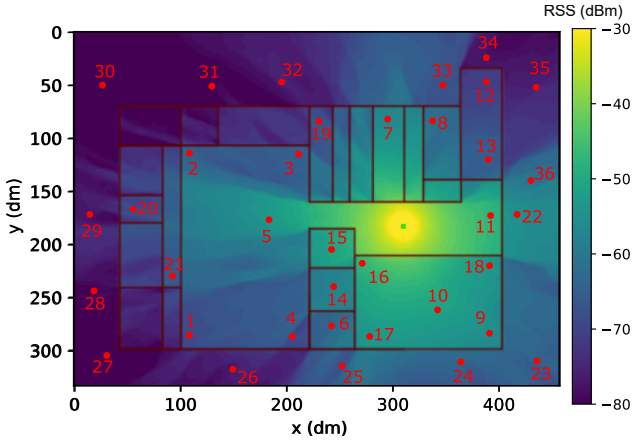
$$\min_{\Theta} \quad \left(\frac{1}{NWH} \sum_{n,i,j=1}^{N,W,H} \left(y_{(n)}(\mathrm{i,j}) - \hat{y}_{(n)}(\mathrm{i,j})\right)^2 \right)^{1/2}, \tag{4.10}$$

where $y_{(n)}(\mathrm{i,j})$ and $\hat{y}_{(n)}(\mathrm{i,j})$, are the ray tracing and the predicted pixel intensity values, respectively, for the n-th sample at pixel (i, j). Evidently, if y is zero at most points, for (4.10) to be minimized, the values of Θ should be selected such as $\hat{y}$ is also zero at these points. Thus, using the same loss function for y_c, which has a few non-zero points, will force the pre-trained EM DeepRay to breakdown. To overcome this limitation, during the calibration when (4.10) is computed for y_c and $\hat{y}_c$, their difference is multiplied with a term Q_c defined as:

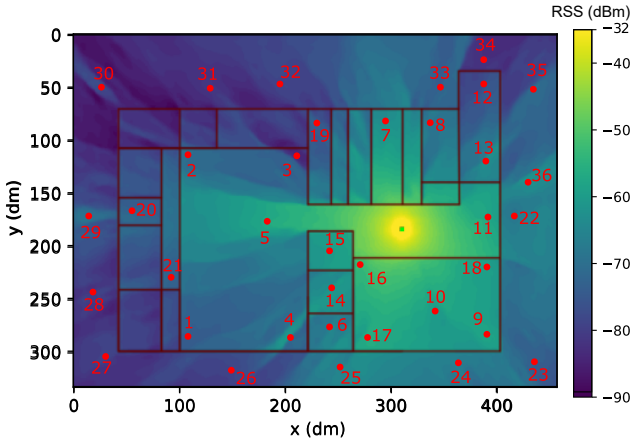
$$Q_c(i, j) = \begin{cases} 1 & \text{if } y_c(\mathrm{i,j}) \neq 0 \\ 0 & \text{if } y_c(\mathrm{i,j}) = 0 \end{cases}, \tag{4.11}$$

and hence the only elements that contribute to the loss function are those for which measured data exist. This allows the pre-estimated weights Θ to be slightly modified during the calibration (retraining) of EM DeepRay, and compute some new weights Θ', in a way that $\hat{y}_c = f(X|\Theta')$ matches only the non-zero values of y_c. It is noted that the role of Q_c is to consider only the non-zero points of y_c during the computation of Θ', however, that does not imply that the predictions for other points of the simulation grid will remain intact. That is to say, that Θ' is estimated based only on a few points, but the change of the weights affects the predictions for the entire grid. Finally, the calibration framework aims at inducing only small changes at EM DeepRay's weights, since the data-driven model already produces a good solution and substantial weight changes using only a small volume of data would drastically alter the PL heatmap and lead to overfitting.

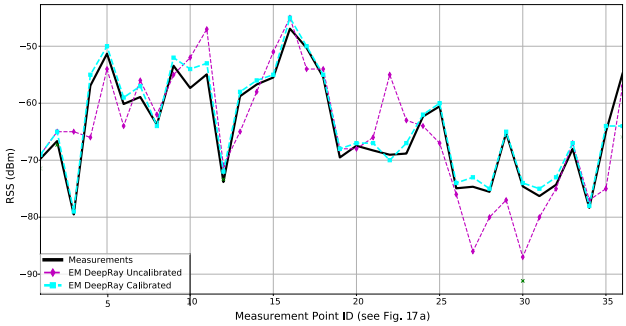
To demonstrate the effectiveness of this approach, the indoor geometry of [114] is generated and provided as an input to DeepRay. In the absence of detailed information regarding the material types, all the walls found within the geometry are assumed to be made of concrete and are 10 cm thick. A single Tx operating at 868 MHz with a transmitting power of 14 dBm is placed in the building, utilising a half-wavelength dipole antenna. The received signal strength is measured at 36 different locations within the building, *i.e.*, the target



(a) EM DeepRay RSS prediction before calibration for the indoor geometry of [114].



(b) EM DeepRay RSS prediction after calibration for the indoor geometry of [114].



(c) Comparison between simulated and measured RSS.

Figure 4.17: Predicted RSS, and comparison with measured data.

Table 4.5: Error of EM DeepRay compared to measurements before and after calibration.

	EM DeepRay	Calibrated EM DeepRay
RMSE (dB)	6.10	1.69
MAE (dB)	4.65	1.22
Computational time (sec)	0.09	15.1

output tensor, y_c has only 36 non-zero points, which are depicted as red dots in Figs. 4.17a and 4.17b, numbered from 1 to 36. At each position, 200 RSS measurements are taken and they are averaged to evaluate the mean signal strength. More detailed information about the measurement setup can be found in [114].

Then, input and target tensors are passed to the SDU-Net which performs a pixel-to-pixel regression, updating its weights considering only the non-zero points of the target output image tensor, y_c, as discussed earlier. The SDU-Net is retrained using the Adam optimisation algorithm. To avoid abrupt changes of Θ, the SDU-Net is trained over fewer epochs and at a smaller learning rate than that of Section 4.1.3. Expressly, the learning rate is equal to 10^{-5} and the training lasts for 50 epochs. Finally, to prevent overfitting an L_2 regularisation is added at each convolutional layer, setting the regularisation parameter equal to 0.01 [61]. The calibration procedure takes 15 seconds since the SDU-Net is retrained over a single sample. Once the SDU-Net is retrained, the predicted PL tensor $\hat{y}_{c,o}$ is obtained and the transmitting power is subtracted from it to derive the predicted RSS values.

The RSS predicted by DeepRay before calibration is shown in Fig. 4.17a, with the blueprint of the input geometry printed on top of it. The predicted RSS after calibration is shown in Fig 4.17b. In both cases the mode received the same physics-based input tensor, yet due to the SDU-Net weight recalculation, the produced PL heatmap is different. The measured, simulated, and predicted RSS for each point is shown in Fig. 4.17c, where the x-axis signifies the corresponding red dot position shown in Fig. 4.17a. The largest errors are observed for points 27 to 31, which are outside the building, and the RSS is underestimated with the model inferring higher losses.

Overall, before calibration, the calculated RMSE and MAE values for EM DeepRay are approximately 6 and 4.5 dBm, respectively. After the calibration, the predicted RSS values for EM DeepRay show a close correspondence with the measured data, depicted by the alignment of the black and teal line in Fig. 4.17c, exhibiting an RMSE and an MAE equal to 1.69 and 1.22 dBm, respectively. The computational time for the calibrated EM DeepRay is shown in Table 4.5. Note that even when including the training time (15 seconds), the overall time remains low. More importantly, as pointed out earlier and can be seen in Fig. 4.17b, due to the new weights Θ' the predicted RSS values are different for the entire grid, and not just for the few measurement points.

Table 4.6: Calibration results using a smaller number of random training points, and estimating the error with respect to the remaining measurement points.

Number of Training Points	4	8	16	27
RMSE (dB)	4.46	3.78	3.32	2.57
MAE (dB)	3.62	3.24	2.45	2.01

For instance, as mentioned earlier, the RSS was underestimated for the entire outdoor area on the left side of the building, but after the computation of Θ' the signal for the entire area (and not only for points 26, 27, 28, 29, and 30) assumes higher values. To further assess the assumption that due to the calibration the predicted RSS values for the entire grid are improved, EM DeepRay is calibrated using only a fraction of the total 36 measurement points, and measuring the error with respect to the remaining points. The results are shown in Table 4.6, where EM DeepRay is calibrated using randomly only 4, 8, 16, and 27 out of the 36 measurements, and the error metrics are estimated with respect to the remaining 32, 28, 20 and, 9 measurement points, respectively. As can be seen, the predictions for the rest of the grid are improved, and the accuracy of our model is increased even using a small number of points during the calibration, thus corroborating the assumption that the model's performance can be improved even through a small number of measurements.

4.4 Summary

Wireless communication system design requires robust and expedient propagation modelling tools to ensure optimal network performance. This chapter introduced a generalisable and realistic ML-based radio propagation modelling framework for indoor environments. EM DeepRay exploits physics-based information of an indoor environment to predict the PL within an indoor area of interest, and unlike previous work in the field, the predictions of our model are not restricted to indoor geometries, frequencies and antenna radiation patterns included in the training set, but it can be employed to readily predict the PL for an arbitrarily complex indoor wireless communication system set up. The numerical results indicate that EM DeepRay can successfully replicate the PL simulated by a ray-tracer, and benchmarking its performance with that of a ray-tracer with real-world measured data yields similar results that have a distinct advantage in terms of computational time. Additionally, a calibration method was presented, allowing the data-driven model to adjust its weights in order to make predictions that are closer to measured rather than synthetic data. After calibration, which only takes a few seconds, our model can provide estimates of the signal level that resemble actual measured data with outstanding fidelity. Ultimately, this chapter tackles two fundamental problems of ML-based propagation modelling, specifically generalisability, and efficiency.

Chapter 5

Applications of Data-Driven Propagation Models in Indoor Wireless Networks

Cell densification through the installation of small and femto base stations (fBS) in indoor environments is an emerging solution to enhance the operation of wireless networks. Due to resource and budget constraints, MNOs will deploy 5G indoor networks in a phased and orderly manner. The appropriate building and network topology selection requires robust and reliable tools and metrics that will be able to assess the performance of an indoor wireless network, and consequently guide its optimal design. This chapter presents metrics portraying the aggregated signal quality and throughput of indoor network deployments and enabling the IWNP quantification. It is then illustrated how these metrics can be leveraged for the expedient and automated planning of fBS in a UDN, as well as how one can consider the intrinsic uncertainty related to the material properties of the indoor environment into IWNP calculations, and express the IWNP metrics as stochastic quantities. In both applications, it is shown that leveraging a data-driven propagation model to deliver estimates of the signal attenuation yields results similar to that of a high-performance ray-tracer [107] but in a substantially reduced computational time.

5.1 Indoor Wireless Network Performance Metrics

An important effort towards the characterisation of the performance of in-building wireless networks was made in [115, 116], where the authors introduced the idea of building wireless performance (BWP), which emerges as a fundamental utility of a building, similarly to its eco-friendliness and its electricity, water, and gas consumption. Although this work advanced significantly the state-of-the-art, two main limitations can be identified. First, the authors considered this as an intrinsic property of the building, with BWP providing an upper bound

to the network's performance, assuming an extremely dense small-cell network with antennas deployed throughout the entire building. That is commonly not the case, as MNOs choose to install only a limited number of fBs per building. Second, to derive an analytical solution for their metrics, the authors used simplistic empirical models which as discussed earlier in this thesis exhibit low fidelity.

Aiming to overcome these limitations and to provide compound measures that can be leveraged by wireless network engineers to deliver optimal wireless network design, a set of metrics that can be used for the evaluation of IWNP is introduced in the section. Specifically, to quantify IWNP, three metrics are considered in this work, aiming to characterise the aggregated network coverage and capacity. The first two are the power and the interference effectiveness, P_{eff} and I_{eff}, respectively. The power effectiveness quantifies the coverage and how the building layout facilitates the propagation of the electromagnetic waves, and consequently how strong is the received signal from the serving device. Unlike [116, 115], where the BWP metrics were defined with respect to the open-space scenario, the IWNP metrics used in this work consider only the propagation characteristics within the building. This is primarily because the goal is to quantify the performance of an indoor wireless network itself, rather than comparing it to the open-space propagation scenario.

Let $P_t^{(i)}$ and $PL^{(i)}(\omega)$ be the transmitting power and the PL distribution for the i-th fBs, respectively, defined over all locations ω in the area Ω covered by the indoor environment of interest. The power effectiveness over Ω is then:

$$P_{eff} = \frac{1}{\Omega} \int_{\Omega} P_t^{(a(\omega))} / PL^{(a(\omega))}(\omega) w(\omega) d\omega, \tag{5.1}$$

where $a(\omega) = \arg\max_{i \in A \cup A'} P_t^{(i)} / PL^{(i)}(\omega)$ represents the strongest available signal at each point of the grid, selected from the set of all fBS, A, and the set of all outdoor macro BSs, A', and $w(\omega)$ is a weighting function assuming values between 0 and 1. The intuition behind the use of $w(\omega)$ is that certain areas within the building might not be of interest, *e.g.*, storage rooms or bathrooms, thus through $w(\omega)$ we can limit their contribution in the compound metric. Also, it should be noted that the definition implies that in some areas of the building, a user could be served by an outdoor BS.

The second metric is the interference effectiveness that captures how the signals emitted from different fBSs interact with each other, quantifying the received SINR:

$$I_{eff} = \frac{1}{\Omega} \int_{\Omega} \frac{P_t^{(a(\omega))} / PL^{(a(\omega))}(\omega)}{\sum_{i \in \bar{a}(\omega)} I_i(\omega) + \sum_{j \in \bar{a'}(\omega)} I'_j(\omega) + N_{ch}} w(\omega) d\omega, \tag{5.2}$$

where $\bar{a}(\omega)$ and $\bar{a'}(\omega)$ indicate the sets that include all femto and macro BSs apart from $a(\omega)$, *i.e.*, $\bar{a}(\omega) = \{A - a(\omega)\}$ and $\bar{a'}(\omega) = \{A' - a(\omega)\}$, $I_i(\omega) = P_t^{(i)} / PL^{(i)}(\omega)$ corresponds to the co-channel interfering received power from the i-th fBS, $I'_j(\omega) = P_t^{(j)} / PL^{(j)}(\omega)$ is the

co-channel interfering received power from j-th macro BS, and N_{ch} is the channel noise. Note that I_{eff} is a qualitative proxy for the throughput and it is independent of the actual 5G system configuration.

The third metric is the throughput effectiveness, T_{eff}, which is estimated based on the SINR and provides a quantitative insight regarding the end-user QoS:

$$T_{eff} = \frac{1}{\Omega} \int_{\Omega} \Xi \left(\frac{P_t^{(a(\omega))} / PL^{(a(\omega))}}{\sum_{i \in \bar{a}(\omega)} I_i(\omega) + \sum_{j \in \bar{a'}(\omega)} I'_j(\omega) + N_{ch}} \right) w(\omega) d\omega, \tag{5.3}$$

where Ξ is a mapping function utilised to map the SINR onto the channel quality indicator, which designates the achievable modulation order and coding rate. Specifically, to compute the throughput the 3GPP Technical Specification (TS) 38.306 is used [117], which covers the user equipment radio access capabilities for 5G new radio (NR) and includes the formula for the maximum supported data rate (DR) in megabits per second (Mbps):

$$DR = n_{str} \cdot f_{SC} \cdot \eta \cdot \frac{N_{RB/B} \cdot N_{SC/RB}}{T_S} \cdot (1 - OH), \tag{5.4}$$

where n_{str} gives the number of streams between the BS and the end user, f_{SC} captures the scaling factor related to MIMO layers and modulation order, η is the channel spectral efficiency coefficient related to the modulation order and coding rate which depend on the SINR achieved, $N_{SC/RB}$ signifies the number of resource blocks in the available bandwidth, $N_{SC/RB}$ is the number of subcarriers per resource block, T_S corresponds to the average OFDM symbol duration, and OH, stands for the signalling overhead. To achieve the maximum data rate the TS assumes that the highest possible coding rate and modulation order are employed, *i.e.*, η takes its highest value. To adapt the formula to a practical case, not achieving the best channel conditions, the measured or simulated SINR is matched to a channel quality indicator (CQI), as shown in Table 5.1. Then, based on the CQI, the corresponding spectral efficiency is determined for a block error rate not exceeding 0.1, as per 3GPP TS 38.214. Ultimately, Ξ maps the SINR at every point of the grid to a spectral efficiency value via Table 5.1, and then computes the throughput at each point via (5.4).

The calculation of any of the three metrics requires knowledge of the signal distribution over space, which can be determined using a propagation model. Then, assuming a dense grid discretisation, the integrals in (5.1) – (5.3) can be evaluated as a Riemann sum. The use of a data-driven propagation model for the evaluation of the signal distribution, such as EM DeepRay, can bridge the efficiency-accuracy gap that conventional propagation models exhibit, providing reliable estimates of the PL in a computationally affordable manner. Specifically, with empirical models one could conduct fast radio map simulations resulting in IWNP estimates of questionable reliability, while employing deterministic site-specific models would fetch more realistic results though entailing long simulation times. Instead, due to its native GPU implementation, EM DeepRay can produce very fast results alike to those of a

Table 5.1: Mapping estimated SINR to CQI and spectral efficiency values.

SINR (dB)	CQI	Modulation	Code Rate × 1024	η
-6.9360	1	QPSK	78	0.1523
-5.1470	2	QPSK	193	0.3770
-3.1800	3	QPSK	449	0.8770
-1.2530	4	16 QAM	378	1.4766
0.7610	5	16 QAM	490	1.9141
2.6990	6	16 QAM	616	2.4063
4.6940	7	64 QAM	466	2.7305
6.5250	8	64 QAM	567	3.3223
8.5730	9	64 QAM	666	3.9023
10.3660	10	64 QAM	772	4.5234
12.2890	11	64 QAM	873	5.1152
14.1730	12	256 QAM	711	5.5547
15.8880	13	256 QAM	797	6.2266
17.8140	14	256 QAM	885	6.9141
19.8290	15	256 QAM	948	7.4063

ray-tracer.

5.2 UDN Planning with Data-driven Propagation Models

From a planning perspective, UDN environments are characterised by very large degrees of freedom in the corresponding locations of numerous fBSs, and a diversity of possible target performance indicators to be optimised. This makes existing strategies for UDN planning based on ray tracing techniques unfit for the task, as they can only explore a limited number of network topologies, leading to deployments that are largely sub-optimal with respect to what could be achieved in practice. For instance, in [13, 14], capacity, spectrum, and energy efficiency metrics were used to evaluate the performance of a UDN, employing ray tracing software [50] to model radio wave propagation under a small set of predetermined deployments, assuming simplified indoor geometries.

This calls for optimisation frameworks that can efficiently and automatically explore the vast space of feasible deployments. Formally, near-optimal UDN planning schemes require enhancing one or more of three components: (*i*) a radio propagation modelling tool; (*ii*) a suitable target KPI metric that quantitatively represents the performance of the deployment; (*iii*) an optimisation problem formulation that can return a high-performing deployment of fBSs [13, 14, 15]. However, the convergence towards effective deployments requires a large number of iterations within the optimisation process. Therefore, computationally expensive approaches based on ray tracing that take minutes or even hours for a single simulation,

such as those in [13, 14], cannot support thorough explorations of the deployment space. For instance, UDN planning with commercial ray tracing software becomes computationally unfeasible under advanced optimisation techniques such as genetic, particle swarm, or evolution algorithms, even in relatively small-scale scenarios with a limited number of fBSs.

On these grounds, this section discusses an original implementation of the UDN planning optimisation process, which builds on the expediency and accuracy of EM DeepRay. This use case provides a first insight into the key role that data-driven propagation models may assume in network planning, allowing the severe computational time impediments imposed by legacy ray-tracers to be overcome. Note that while the optimisation framework adheres to the standard network planning techniques followed in industry and academia, unlike previous works it employs a full-fledged data-driven model to evaluate signal propagation, and thus this implementation provides solid evidence that such model can indeed replace legacy solvers without performance degradation. Consequently, this Section effectively exemplifies the capacity of data-driven models to streamline essential network operations, inaugurating a new era of automation in the wireless ecosystem, and paving the way for the adoption of these models in new applications. Specifically, it is shown how a data-driven propagation can replace a ray-tracer in the default UDN planning scheme mentioned earlier, enabling on-the-fly identification of the network topology that optimises a target KPI. The optimisation process based on EM DeepRay yields deployments with similar performance to those given by a time-consuming ray-tracer. More importantly, it also allows deployment options to be explored much more rapidly, hence supporting advanced optimisers with higher performance and scaling to larger use cases. The proposed approach eliminates human intervention or domain knowledge and, building on the robustness of data-driven propagation models, paves the way for automated AI-powered network planning solutions. By speeding up the planning process, and providing higher flexibility in terms of deployment choices, one can identify low-cost deployments that can increase network spectral efficiency, ensuring higher-performance network operation.

5.2.1 Optimisation Framework Overview

The overall proposed framework for AI-assisted indoor UDN planning is illustrated in Fig. 5.1, and comprises three subcomponents: (A) a data-driven propagation model used to swiftly evaluate signal distribution throughout the building (green sub-box), (B) an appropriate KPI that quantifies the performance of the network topology (purple sub-box), and (C) an AI-based optimiser used to explore potential network topologies (orange box). It is important to note that the proposed strategy is general in nature and can accommodate any implementation of subcomponents (A)-(C). Yet, for the purpose of comparison against legacy approaches that use conventional propagation solvers [13, 14, 118, 119, 120], one specific instance of this general concept is studied in this thesis, exploiting EM DeepRay and the

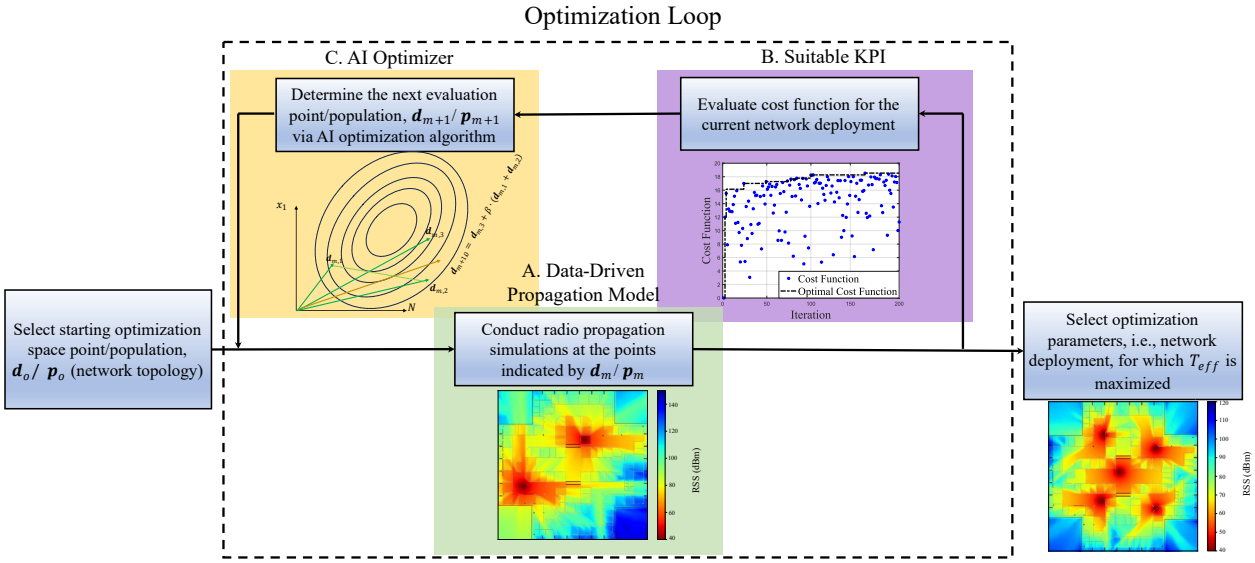


Figure 5.1: Complete framework for AI-assisted indoor wireless network planning with a data-driven propagation model, comprising (A) an expedient data-driven propagation model leveraged to readily emulate wireless signal propagation (green box), (B) evaluation of the metric describing the performance of the network (purple box), and (C) a $black-box$ computational intelligence optimiser that guides the optimisation space point selection by observing the harvested cost function for each point of the optimisation space, *i.e.*, possible deployment (orange box). Starting from a randomly selected deployment (left), the network performance is measured iteratively for deployments indicated by the computational intelligence algorithm, ultimately leading to a nearly-optimal network design (right)

throughput effectiveness metric defined in Section 5.1.

5.2.2 Computational Intelligence Optimisers

To determine the UDN deployment that optimises the KPI based on ML-based radio signal estimates, it is necessary to explore the solution space, *i.e.*, the number of fBSs and their placement, in a smart way. Therefore, the UDN planning framework of Figure 5.1 leverages AI-based optimisers, which are employed in cases where the optimisation objective function, f_{opt}, does not have a known special structure, but it is an unconstrained black-box. In particular, the following state-of-the-art optimisation algorithms are considered: (i) Bayesian optimisation (BO), (ii) genetic algorithms (GA), (ii) differential evolution (DE) and (iv) particle swarm optimisation (PSO) [121, 122]. BO is a non-population-based optimiser, *i.e.*, at each iteration, the cost function is evaluated at a single point, $\boldsymbol{d}_{opt} = [x_{opt,1}, x_{opt,2}, ..., x_{opt,N_{opt}}]$, of the multi-dimension optimisation space with N_{opt} optimisation variables, $x_{opt,1}, x_{opt,2}, ..., x_{opt,N_{opt}}$. The other algorithms are population-based, retrieving at each iteration K different solutions by evaluating the cost function at a set of K distinct optimisation space points referred to as the population, $\boldsymbol{p}_{opt}$. Note that each one of the K distinct optimisation points comprises N_{opt}

optimisation variables.

In the superscript (m, k) let m indicate the iteration of the optimisation algorithm, whilst for population-based methods k designates a specific optimisation point from the population. Initially, a starting point, $\boldsymbol{d}_{opt}^{o}$, comprising the optimisation parameters, or a population of K starting points, $\boldsymbol{p}_{opt}^{o} = [\boldsymbol{d}_{opt}^{(o,1)}, \boldsymbol{d}_{opt}^{(o,2)}, ..., \boldsymbol{d}_{opt}^{(o,K)}]$, is randomly selected and the cost function f_{opt} is evaluated at it. Then, based on the computed values of f_{opt}, the optimisation algorithm is employed to determine the point or set of points from the optimisation space, $\boldsymbol{d}_{opt}^{m+1}$ or $\boldsymbol{p}_{opt}^{m+1}$, at which f_{opt} will be evaluated in the next iteration. This process is repeated iteratively until a convergence criterion is met or a certain maximum number of iterations is reached.

The difference between the various optimisation algorithms lies in the way $\boldsymbol{d}_{opt}^{m+1}$ or $\boldsymbol{p}_{opt}^{m+1}$ are chosen [121, 122]. For instance, BO selects a single candidate point, $\boldsymbol{d}_{opt}^{m}$, per iteration and utilises a Gaussian Process (GP) regressor, and an acquisition function to determine $\boldsymbol{d}_{opt}^{m+1}$ [122]. The GP regression assumes that the prior distribution of f_{opt} follows a multivariate normal distribution, whose mean vector and covariance matrix values are calculated based on the computed values of f_{opt} at selected points of the optimisation space. Then, one can derive the posterior probability of f_{opt}, and use the acquisition function to identify a good candidate point $\boldsymbol{d}_{opt}^{m+1}$ for the next iteration.

In GAs the N_{opt} optimisation variables are referred to as genes. A population $\boldsymbol{p}_{opt}^{m}$ consists of K_{GA} points (also referred to as chromosomes) of the optimisation space each one comprising N_{opt} genes. The cost function is evaluated for each point $\boldsymbol{d}_{opt}^{(m,k)}$ of the population and the $n_{m,GA}$ points with the best cost function values are selected as parents to generate a new offspring (population for the next iteration). To achieve that, crossover and mutation operations are employed. The first entails merging the parents' genes according to a specific scheme in order to create new chromosomes. The scattered crossover scheme, used in the next subsection, generates a binary number for each chromosome gene, and then the offspring gene assumes the value of the respective parent gene. Note that for each chromosome gene, there is $1 - p_c$ probability that no cross-over will occur and the gene value will remain the same for the next iteration. Finally, there is a probability p_{mu} to mutate each offspring gene by randomly changing its value.

DE works similarly, but the mutation and crossover operations exploit the difference between two or more optimisation points of the population to create a new optimisation point; usually by randomly scaling this difference and adding it to one of the existing vectors along with some randomly sampled noise. A common evolution strategy is "Best1bin", where for each optimization point, $\boldsymbol{d}_{opt}^{(m,k)}$, a mutation point, $\boldsymbol{d}_{opt}^{'(m,k)}$, is generated point by mutating the best solution so far, $\boldsymbol{d}_{opt}^{*}$. Specifically, the mutation is accomplished by adding to $\boldsymbol{d}_{opt}^{*}$ a scaled difference of two randomly sampled optimisation points, $\boldsymbol{d}_{opt}^{r_1}$ and $\boldsymbol{d}_{opt}^{r_2}$, respectively, *i.e.*, $\boldsymbol{d}_{opt}^{'(m,k)} = \boldsymbol{d}_{opt}^{*} + \beta \cdot (\boldsymbol{d}_{opt}^{r_1} - \boldsymbol{d}_{opt}^{r_2})$ where β is the differential weight parameter. Then a trial vector is constructed by applying a crossover between $\boldsymbol{d}_{opt}^{(m,k)}$ and $\boldsymbol{d}_{opt}^{'(m,k)}$; for each optmisation

Table 5.2: Optimisation Algorithm Parameters.

Method	Parameters [122, 121]
BO	Acquisition function = EI
GA	$K_{GA} = 8$; number of chromosomes, *i.e.*, distinct solutions of the optimisation space explored per iteration $n_{m,GA} = 4$; number of parents, *i.e.*, top-performing chromosomes selected to generate new solutions Scattered crossover; randomly assigns to each offspring gene the value of one of the respective parent genes $p_c = 0.8$; probability of changing a chromosome gene via crossover Random mutation, $p_{mu} = 0.3$; probability of randomly changing the value of a gene
DE	$K_{GA} = 6$; population size, *i.e.*, distinct solutions of the optimisation space explored per iteration Best1bin Evolution; $\beta \sim U(0, 1)$; differential weight used to scale the difference of two random population points $p_r = 0.7$; probability of selecting a trial vector parameter from the original or the mutated optimisation point
PSO	Global-best particle swarm optimisation; particles consider the best solution from the entire swarm $K_{PSO} = 10$; number of particles, *i.e.*, distinct solutions of the optimisation space explored per iteration $c_1 = 0.3$; acceleration parameter indicating the tendency of a particle to stick to its own best solution $c_2 = 0.5$; acceleration parameter indicating the tendency of a particle to stick to the global best solution $w = 0.7$; inertia parameter indicating the tendency of a particle to stick to its current state;

parameter a random value between 0 and 1 is generated and if this value is larger than the recombination constant, p_r, then the trial vector parameter is set equal to that of $\boldsymbol{d}_{opt}^{\prime(m,k)}$, otherwise it takes the parameter value of the initial optimisation point. Note that the last parameter is always taken from $\boldsymbol{d}_{opt}^{\prime(m,k)}$. Consequently, the cost function is evaluated for the trial vector, and if its value is better than that of $\boldsymbol{d}_{opt}^{(m,k)}$ the trial vector is selected as an optimisation point for the next iteration, otherwise, $\boldsymbol{d}_{opt}^{(m,k)}$ is retained. Note that DE was initially introduced for continuous-valued optimisation spaces, however, it can be easily extended to discrete-valued spaces by rounding the float-number optimisation parameters to the nearest integer [123].

Finally, in PSO the position of K randomly selected points of the optimisation space (also referred to as particles), constituting the population, is iteratively adjusted based on (i) their own knowledge of the optimisation space, *i.e.*, the values of the cost function encountered by the same particle at past iterations, and (ii) that of the neighbouring particles. Consequently, the hyperparameters that control the behaviour of the particles are the number of neighbours whose knowledge is considered when updating the particle position, and the acceleration parameters c_1 and c_2 which quantify the contribution of the particle's personal knowledge and that of its neighbours in the position update process [121]. Additionaly, the inertia parameter, w, depicts the tendency of a particle to remain in the current optimisation solution.

Ultimately, rather than conducting brute-force simulations, and scanning the entire optimisation space, the computational intelligence optimisation algorithms allow the points at which f_{opt} will be evaluated to be chosen in a smart manner. The selection of these new points aims at either mitigating the lack of knowledge regarding the values of f_{opt} or potentially identifying a new best value in an area of the optimisation space that appears to be favourable.

5.2.3 Evaluation in Practical Use Cases

To demonstrate how data-driven propagation models can be leveraged for the rigorous optimisation of IWNP, the urban environment shown in Fig. 5.2 is created, and three indoor environments in which fBSs will be installed are considered. The buildings are represented as black dots and annotated as B1, B2, and B3, in Figure 5.2, whilst their geometry is shown in Figures 5.3a-5.6a, respectively. As can be seen, the buildings are constructed using various materials that are commonly found in indoor environments, such as concrete, brick, glass, and plasterboard. Some of the areas within the building are considered to be of lower wireless communication importance, *e.g.*, toilets or storage rooms, hence, for these locations, the weighting function, $w(\omega)$, assumes a lower value in order to limit their contribution in the optimisation procedure. In particular, for the areas covered with diagonal and horizontal blue lines, the weighting function is equal to 0.7 and 0.2, respectively. Finally, it is noted that none of the buildings has been used during the training phase of EM DeepRay.

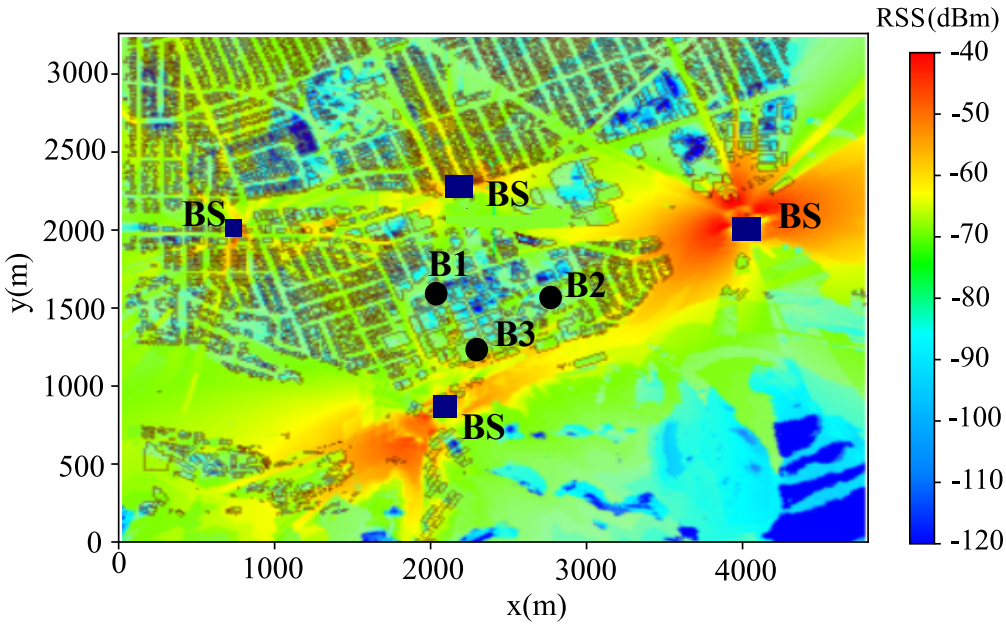


Figure 5.2: Digital twin of an urban environment used to conduct an outdoor-to-indoor ray tracing simulation, providing the signal heatmaps of four macro BSs (black squares) within the digital replicas of three buildings, B1, B2 and B3 (black dots), where fBSs will be deployed.

With the building layouts defined, an outdoor-to-indoor ray tracing simulation is conducted to derive the signal heatmaps owing to the 4 outdoor macro BSs, shown as black squares in Figure 5.2, within the buildings at which the fBS will be deployed. To simulate the outdoor-to-indoor RSS, the Ranplan Wireless ray tracing simulator is used [107]; this reference is not affected by the UDN deployment, hence it is only calculated once, and computing it via an expensive ray tracing approach is affordable. The operating frequency of the macro BS and the femtocells is set to 2 GHz, the channel bandwidth is 5 MHz, the sub-carrier spacing is 15 kHz, and the number of resource blocks is 25. It is assumed that there is a single stream

between the BSs and the user and that the scaling factor f_{SC} is equal to one. Furthermore, the macro BSs use directional antennas with 18 dBi gain, and a half power beam width equal to 57° and 15° in the horizontal and the vertical planes, respectively. The femtocells are equipped with antennas having an omnidirectional beam pattern and a 3 dBi antenna gain. The transmission power for the macro BSs and the femtocells is constant and equal to 49.4 and 20 dBm, respectively.

For each point within the simulated domain, the fBS with the strongest signal is selected as the serving one. It is assumed that there is no interference between the femtocells due to the use of interference mitigation techniques, such as coordinated multipoint, and that the only interference is introduced by the outdoor macro BS to the serving femtocell. It is further assumed a constant noise power spectral density equal to -174 dBm/Hz and a noise figure at the receiving signal chain equal to 5 dB.

The optimisation parameters are the number, N_f, of the fBSs to be placed along with their position within the building, *i.e.*, $\boldsymbol{d}_{opt} = [N_f, x_1, y_1, ..., x_{N_f}, y_{N_f}]$, where $x^{(n)}$ and $y^{(n)}$ are the Cartesian coordinates of the n–th fBS. For the illustrative experiments carried out in this example, the maximum number of fBSs is bounded to 5, as this is already sufficient to demonstrate the practical computational advantage of the proposed approach in settings where the fBSs can be placed anywhere within the building. As shown in Fig. 5.1, at the m-th iteration, the optimiser designates the number, N_f^m, of the fBSs and their positions. Then, N_f^m radio prorogation simulations are conducted to evaluate the received signal throughout the building, one for each one of the N_f^m fBSs. For population-based methods, for each optimisation point $\boldsymbol{d}_{opt}^{m,k}$ in the population $\boldsymbol{p}_{opt}^m$, $N_f^{m,k}$ simulations are run, where $N_f^{m,k}$ indicates the number of fBSs for the k-th optimisation point of the population at the m-th iteration. Consequently, the value of T_{eff} is computed for the current network topology, or the K different network topologies in population-based methods, and it is employed to determine the optimisation point/population for the next iteration according to the optimisation scheme used.

The optimisation setup for each algorithm is as follows; for BO the expected improvement acquisition function is used, allowing up to 200 iterations [122]. For GA 42 iterations are allowed, whilst at each iteration the population comprises $K_{GA} = 8$ distinct points of the optimisation space, out of which $n_{m,GA} = 4$ are chosen as parents to determine the points of the next population, using a steady-state selection scheme [121]. A scattered crossover operation, with a crossover probability $p_c = 0.8$, is applied to the parents, and a random mutation scheme with a mutation probability equal to $p_{mu} = 0.3$ is used [121]. For DE, 25 iterations are allowed, employing the best1bin evolution strategy, with a population size equal to $K_{DE} = 6$, a differential weight β drawn at each iteration from $U(0.5, 1)$, and a recombination probability equal to $p_r = 0.7$ [121]. In the PSO case, the global-best particle swarm optimisation algorithm is used, where each particle is influenced by the best solution

Table 5.3: Optimisation Performance Statistics.

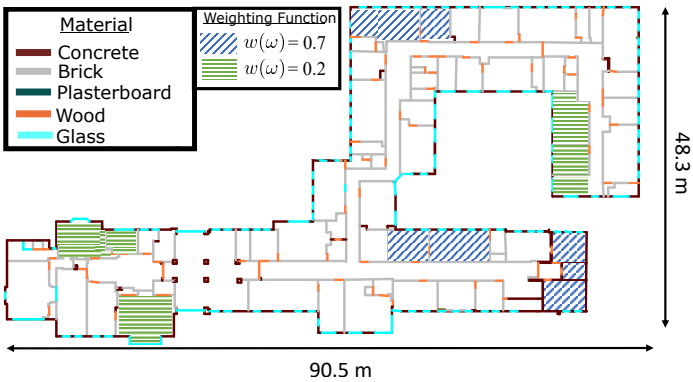
		B1			**B2**			**B3**		
		Ray Tracing	EM DeepRay	*Hybrid*	Ray Tracing	EM DeepRay	*Hybrid*	Ray Tracing	EM DeepRay	*Hybrid*
BO	T_{eff}	15.04	16.16 ± 0.74	16.02 ± 0.73	17.59	19.95 ± 0.55	17.04 ± 0.75	23.53	25.43 ± 0.22	22.86 ± 0.69
	CE	149	108.63 ± 31.73	-	87	98.81 ± 33.50	-	141	77.27 ± 24.97	-
	CT	23,040 (*6.4 h*)	134.78 ± 21.61	-	62,280 (*17.2 h*)	143.27 ± 67.64	-	44,352 (*12.32 h*)	102.30 ± 78.14	-
GA	T_{eff}	-	15.18 ± 1.92	14.30 ± 2.43	-	20.70 ± 0.34	17.02 ± 0.63	-	25.86 ± 0.11	23.56 ± 0.38
	CE	-	980.78 ± 139.61	-	-	977.45 ± 20.18	-	-	796.72 ± 79.07	-
	CT	-	352.62 ± 99.75	-	-	678.44 ± 107.59	-	-	367.86 ± 61.01	-
DE	T_{eff}	-	17.49 ± 0.81	16.25 ± 1.03	-	20.78 ± 0.32	17.96 ± 0.75	-	25.69 ± 0.13	23.37 ± 0.48
	CE	-	236.27 ± 75.01	-	-	183.54 ± 60.98	-	-	147.72 ± 58.06	-
	CT	-	182.47 ± 63.70	-	-	110.33 ± 40.18	-	-	94.52 ± 40.67	-
PSO	T_{eff}	-	16.61 ± 0.90	15.01 ± 1.21	-	20.67 ± 0.48	17.62 ± 0.65	-	25.79 ± 0.15	23.40 ± 0.59
	CE	-	475.09 ± 134.60	-	-	553.81 ± 47.14	-	-	552.72 ± 48.53	-
	CT	-	257.03 ± 74.68	-	-	353.96 ± 49.13	-	-	322.40 ± 35.57	-

found within the entire swarm, with the cognitive parameters c_1 and c_2 set to 0.3 and 0.5, respectively, whilst the inertia parameter, w, is set to 0.7 [121]. The swarms consist of $K_{PSO} = 10$ particles and maximum 60 iterations are allowed. More details regarding the optimisation hyperparameters can be found in [121, 122].

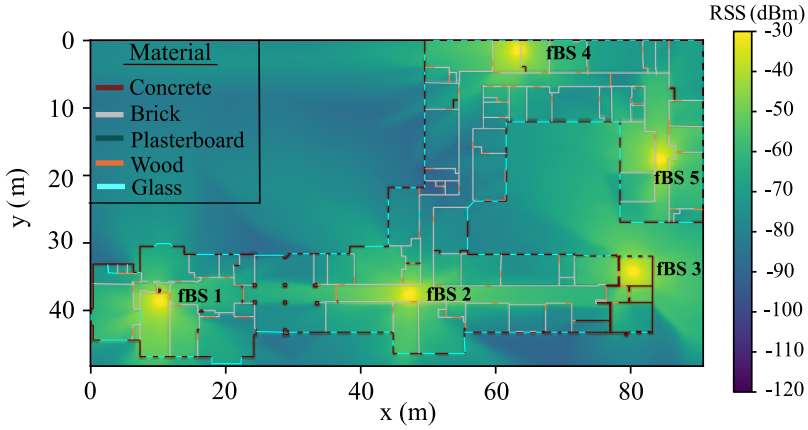
The overall optimisation results for the three buildings are illustrated in Table 5.3, presenting the values of the cost function, the convergence epoch (CE) and the convergence time (CT), measured from the start of the optimisation process until the epoch at which the optimal solution occurs. The optimisation results of the ML-based propagation model are benchmarked against those of ray tracing. Additionally, *Hybrid* optimisation approach is considered, conducting ray tracing simulations for the solution space optimal point determined during the optimisation process with the ML-based model. This approach allows corroborating whether the topologies indicated by the data-driven propagation model optimisation also yield a nearly-optimal point when signal propagation is approximated via a high-performance propagation solver. It should be pointed out that the proposed framework leverages metaheuristic[1] algorithms to retrieve a nearly optimal solution. Thus, to quantify the uncertainty entwined with the intrinsic stochasticity of these algorithms, the optimisation process with each algorithm is repeated 10 times assuming different seeds, and in Table 5.3 the statistics of this process are reported. As will be discussed later, such an approach is not feasible when using the ray-tracer due to the unbearable computational cost.

The first conclusion to be drawn from Table 5.3 is that for each building all optimisation algorithms converge approximately to the same optimal T_{eff} value. Indeed, the mean values attained by each algorithm are very close, while the standard deviation over the 10 optimisation runs per algorithm remains small, yielding tight confidence intervals around the indicated nearly-optimal solution. It should be noted that since network planning is a combinatorial optimisation problem without a closed-form solution, it's not possible to compare the metaheuristic-based topology to some ground truth solution. Additionally, it is

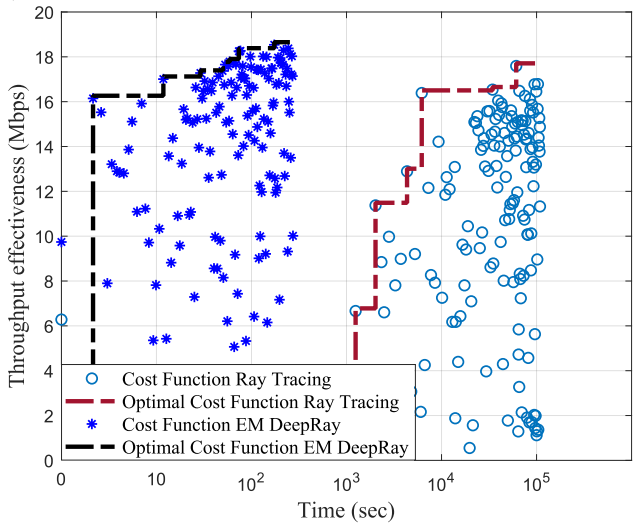
[1]the term metaheuristics is used to describe a set of higher-level strategies that are of general applicability, *i.e.*, they are not problem-dependant, heuristic, and are designed to explore efficiently the optimisation space for complex problems [124].



(a) B1 layout.

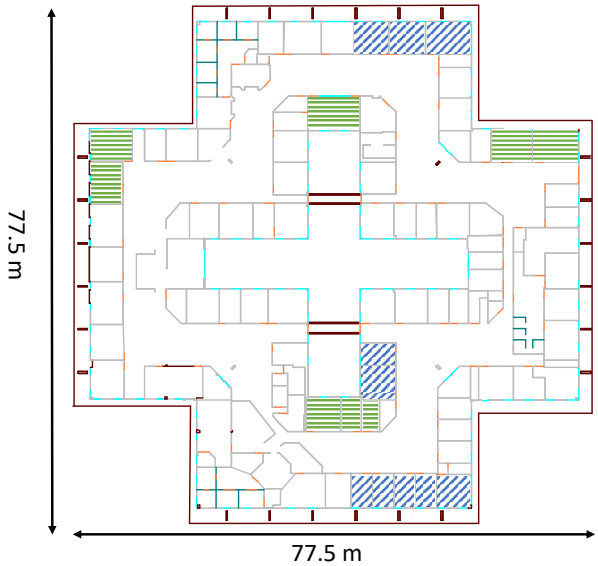


(b) RSS for the indicated optimisation deployment.

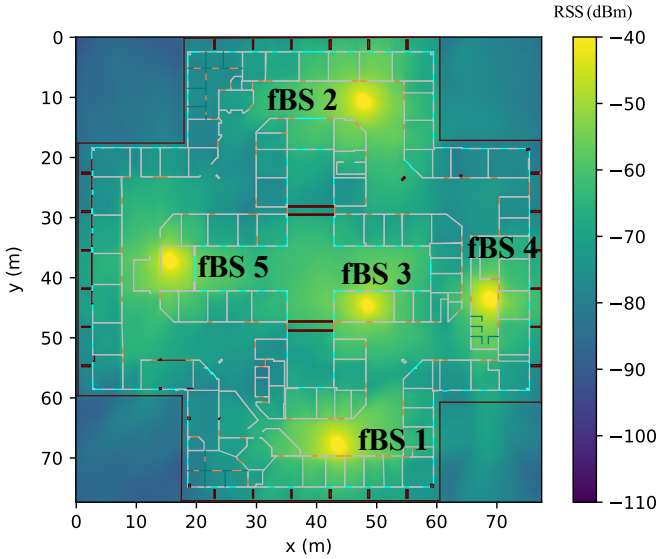


(c) Throughput effectiveness.

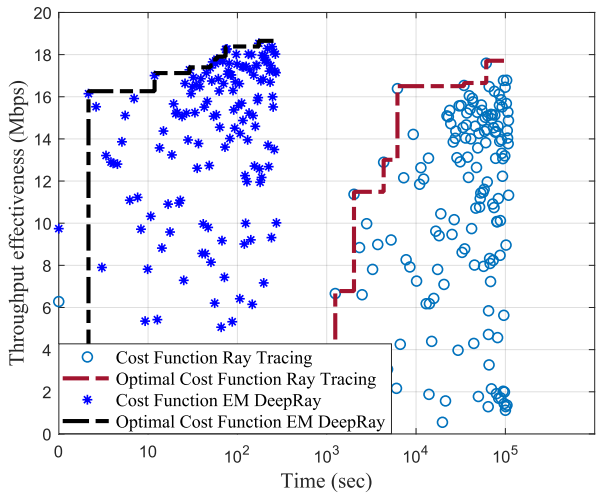
Figure 5.3: Building layout and example of identified nearly-optimal network deployment for B1, along with the evolution of T_{eff} during the optimization with BO, indicating the cost function value achieved the at each iteration, as well as its best value up until that iteration, while using ray tracing and EM DeepRay to emulate signal propagation



(a) B2 layout.

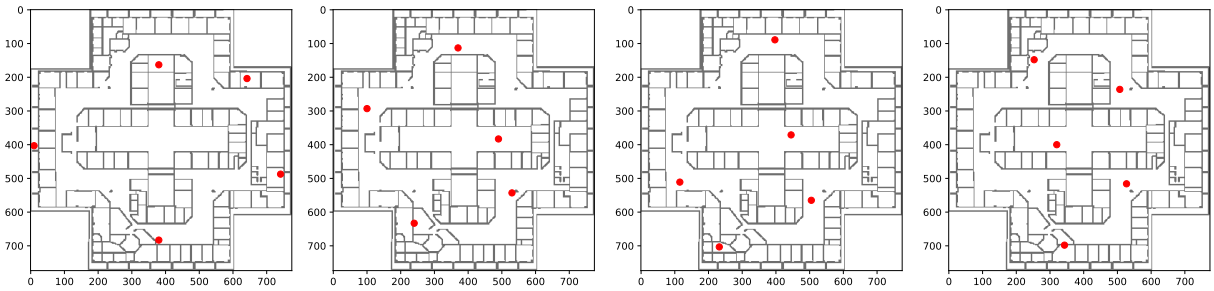


(b) RSS for the indicated optimisation deployment.



(c) Throughput effectiveness.

Figure 5.4: Building layout and example of identified nearly-optimal network deployment for B2, along with the evolution of T_{eff} during the optimisation with BO.



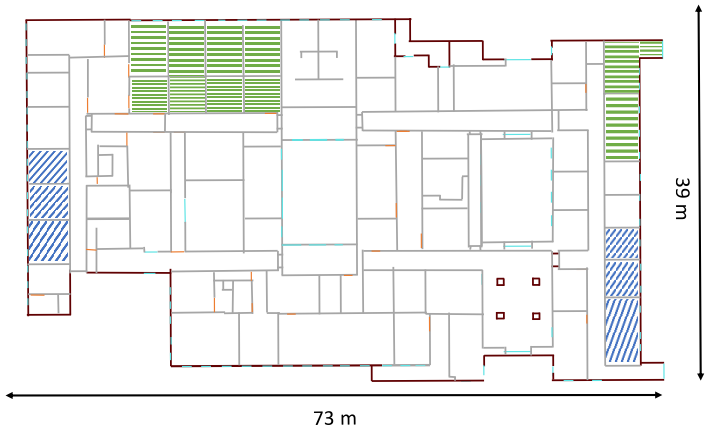
(a) BO. (b) GA. (c) DE. (d) PSO.

Figure 5.5: Optimal site selection for B2 at the end of the optimization process, determined via (a) BO, (b) GA, (c) DE, and (d) PSO

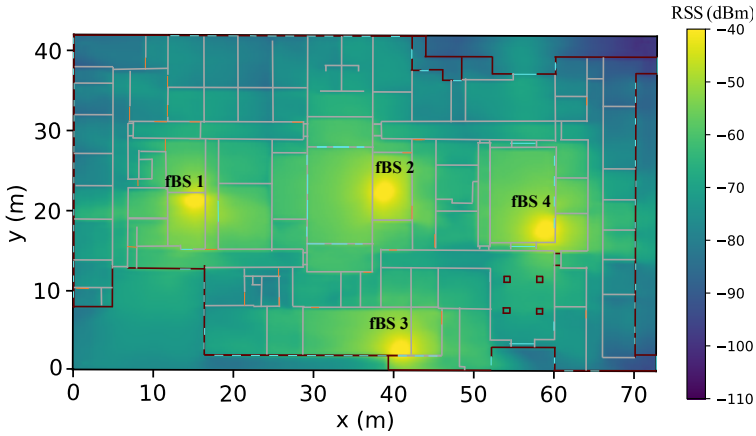
factually infeasible to conduct measurement campaigns for every possible network deployment to decide which is the optimal one and, likewise, to validate with measurements whether the deployment arising from the proposed optimisation process is actually the best possible. Hence, the most objective comparison can rely solely on the numerically evaluated metrics and the overall optimisation time. In this regard, it can be pronounced that DE attains the best sheer performance, achieving the best trade-off between T_{eff} and CT.

Apparently, and unsurprisingly, there can be multiple network deployments that attain nearly-optimal throughput effectiveness, and the potential of the data-driven model to conduct on-the-fly optimisation of network performance enables the identification of several of these deployments; in turn, this translates into higher flexibility for MNOs, which can select the deployment that, *e.g.*, implies the lowest realisation cost in terms of cabling. Furthermore, by juxtaposing some of the optimisation results with the four algorithms, it can be observed that although there are some similarities, there is not a straightforward way to compare the designated deployments. For instance, there are certain areas at which all algorithms tend to place an fBS, *e.g.*, upper right area of B2, but the exact placement points differ. This further corroborates the intuition that there can be multiple network deployments that attain nearly-optimal throughput effectiveness.

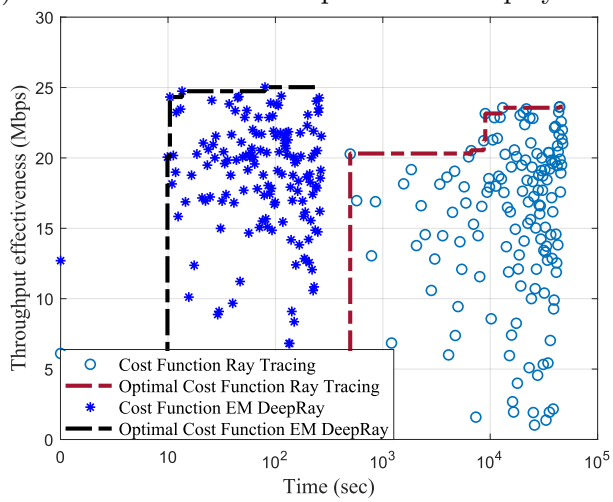
Another key takeout from Table 5.3 is the equivalency between the network topologies indicated by the optimisation with the data-driven propagation model and these determined via the ray-tracer-based optimisation. That is clearly highlighted when comparing the cost function values attained via the ray tracing optimisation and these of the *Hybrid* optimisation scheme (*i.e.*, when T_{eff} is computed via ray tracing but at the optimal points indicated via the data-driven propagation model optimisation process). Indeed, the *Hybrid* optimisation yields similar or even higher throughput effectiveness compared to the ray-tracer-based optimisation for all the buildings and optimisation algorithms. That is to say, if one conducts the optimisation using a ray-tracer or the data-driven propagation model they should expect to find a topology that has approximately the same performance and is equally good. Moreover, this verifies the potential of EM DeepRay to generalise to unknown propagation scenarios and



(a) B3 layout.



(b) RSS for the indicated optimisation deployment.



(c) Throughput effectiveness.

Figure 5.6: Building layout and example of identified nearly-optimal network deployment for B3, along with the evolution of T_{eff} during the optimisation with BO.

effectively replace the legacy ray tracing simulator. The RSS heatmaps for the nearly-optimal network deployment for B1, B2, and B3 at the end of the optimisation procedure with BO are depicted in Fig. 5.3b-5.6b. In all cases, it can observed that the optimisation places the fBSs in diverse areas of the building, to maximise the coverage and capacity of the indoor network.

The distinct advantage of conducting the optimisation process with the data-driven propagation model is the sharp decrease in the time required to identify the nearly-optimal network topology. A ray tracing simulation for a single fBS would take a few minutes. On the other hand, via the data-driven model, one can compute T_{eff} in less than a second. When plugged into the UDN planning framework, this yields a dramatic drop in the optimisation time, as the optimisation with a data-driven model requires a few minutes, whereas with a ray-tracer it takes hours. It is remarked that this is the reason why ray tracing-based optimisations are conducted only with BO, since it is not population-based and it requires the lowest number of epochs to converge; otherwise, the optimisation process could require close to a week. Moreover, the result indicates that a ray tracing-based optimisation for larger scenarios considering a higher number of Txs is prohibitive, as the time will grow exponentially. Finally, it is worth noting that the result obtained with the ray-tracer does not carry standard deviation information, as executing multiple optimisation runs is too expensive; this means that the flexibility granted by a data-driven approach (with multiple proposed deployments of similar quality to choose among, as discussed earlier) is lost with a solution relying on ray tracing.

To better illustrate the computational advantage of the proposed data-driven approach, Figures 5.3c–5.6c present the evolution of the cost function over time on a *log*-scale. The values of T_{eff} associated with each iteration of the optimisation process are depicted as asterisks (for the data-driven solution) and circles (for a legacy ray tracing-based approach). Over time, *i.e.*, moving along the x-axis, the optimisers identify increasingly better deployments with higher T_{eff}, indicated by the dashed black and dark red lines representing the optimal value of T_{eff} attained with time for the data-driven model and ray tracing, respectively. For the majority of the iterations $\boldsymbol{d}_{opt}$ is selected in favourable areas of the optimisation space, while occasionally new areas of the optimisation space are explored resulting in abrupt falls or rises of the optimisation cost function. Evidently, the proposed strategy finds a UDN planning in a time that is more than two orders of magnitude smaller, *i.e.*, less than 3 minutes versus more than 6, 17, and 12 hours for the benchmark for B1, B2 and B3, respectively. Ultimately, using a data-driven propagation model allows the overall planning time to be kept within reasonable bounds while ensuring a robust and effective UDN deployment, with results equivalent to these one would derive if a high-performance ray-tracer had been used.

5.3 Stochastic Wireless Channel Evaluation

An important aspect of radio propagation in indoor environments that is neglected when using empirical models is the uncertainty related to the environment description. This uncertainty can be attributed to the lack of knowledge regarding the geometry itself or regarding the properties, *i.e.*, the permittivity and the conductivity, of the construction materials that are found within the geometry [35]. Indeed, the permittivity and the conductivity can vary significantly according to the material composition and the environmental conditions [125, 126]. Remarkably, in [126] which provides an overview of material parameter measurement campaigns used to develop ITU-R Recommendations [106], the values reported for the relative permittivity of concrete at a single frequency ranged from 3 to 9.

Consequently, the results provided by simulators that use these parameters will be subject to uncertainty due to the randomness in the simulator's inputs. For instance, to estimate the strength and the phase of the waves reflected from and transmitted through building walls in ray tracing simulators, it is necessary to compute the respective slab reflection and transmission coefficients based on the values of ϵ_r' and σ [50]. Analogously, EM DeepRay infers the PL based on the values of the electromagnetic properties of the construction materials depicted in the first two channels of the input tensor shown in Fig. 4.11. However, as mentioned, the values of ϵ_r' and σ are typically derived from look-up tables of ITU standards [106], and they correspond to the median value estimated through large measurement campaigns. Thus, it is likely that the actual material parameter values in different scenarios can vary considerably from those reported in [106].

Therefore, the material parameters should be treated as random variables following a certain probability density function (PDF) [35], and consequently, all the physical quantities related to them, such as the PL distribution, should be considered as stochastic quantities. Consequently, uncertainty will be introduced to any compound features that depend on the results of the propagation solver, *e.g.*, the BWP metrics. This uncertainty can be quantified using conventional uncertainty UQ tools, *e.g.*, Monte Carlo methods, or efficient approaches such as polynomial chaos expansions (PCE) [127, 128, 129].

5.3.1 Uncertainty Quantification

The common method to quantify the uncertainty for these stochastic quantities is the Monte Carlo approach which allows the evaluation of statistical parameters, such as the mean or the standard deviation, by conducting multiple runs of the same problem assuming different values of the input random variables. Despite its simplicity, the Monte Carlo approach exhibits a slow convergence rate, inversely proportional to the square root of the number of the problem runs; instead, a PCE-based representation of the stochastic quantity provides a surrogate solution to the UQ problem, and it has been observed that for problems with a

relatively small number of random variables, *e.g.*, less than 15, the PC statistical quantities converge faster than these of conventional Monte Carlo sampling approaches [130, 131]. Specifically, in PCE the stochastic quantity, $SQ(\boldsymbol{\xi})$, is expanded to an orthogonal basis of polynomial functions [127, 129]:

$$SQ(\boldsymbol{\xi}) = \sum_{p=0}^{P} c_p \Psi_p(\boldsymbol{\xi}), \tag{5.5}$$

where c_p are the coefficients of the basis functions, the input vector $\boldsymbol{\xi} = [\xi_1, \xi_2, ..., \xi_N]$ comprises the N_{RV} input random variables, and Ψ_p are multi-variate polynomial basis functions, selected according to the PDF of the input random variables [128]. The number of the polynomials, P, used to represent the stochastic quantity is $P + 1 = (N + D)!/N!D!$, where D is the highest polynomial order taken into account.

The coefficients of the expansion in (5.5), can be computed through projection-based methods, such as the Smolyak sparse quadrature, or least-squares minimisation methods, such as the orthogonal matching pursuit (OMP) algorithm [132]. The OMP algorithm creates a sparse representation of the PCE in (5.5), by considering only the terms with the highest contribution, and hence reducing the computational cost. Assuming N_{PCE} repetitions of a UQ problem, one can form an output matrix $\mathbf{Y}_{PCE} = [Y^{(1)}, Y^{(2)}, ..., Y^{(N_{PCE})}]^{\intercal}$ including the deterministic responses of the stochastic quantity for each repetition. Similarly, an input matrix can be formed, $\boldsymbol{\Xi} = [\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, ..., \boldsymbol{\xi}^{(N_{PCE})}]^{\intercal}$, where $\boldsymbol{\xi}^{(m_{UQ})}$ denotes the sampled input random variables for the m_{UQ}-th repetition, and a matrix $\boldsymbol{\Psi}(\boldsymbol{\xi}^{(m_{UQ})}) = [\Psi_0(\boldsymbol{\xi}^{(m_{UQ})}), \Psi_1(\boldsymbol{\xi}^{(m_{UQ})}), ..., \Psi_P(\boldsymbol{\xi}^{(m_{UQ})})]$ containing the values of the orthogonal polynomials for each repetition. Then, the coefficients can be computed through:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \mathbb{E}[(\boldsymbol{\Psi}(\boldsymbol{\Xi})\mathbf{c}^{\intercal} - \mathbf{Y}_{\mathbf{PCE}})^2], \tag{5.6}$$

where $\mathbf{c} = [c_0, c_1, ..., c_P]^{\intercal}$ is a vector containing all the PCE coefficients. The OMP algorithm searches and determines iteratively the polynomial basis functions with the highest contribution in the PCE [132], dropping the less influential terms. To achieve that, the algorithm searches iteratively for the polynomial terms that aid the most in reducing the error between the constructed solution and the output matrix $\mathbf{Y}_{PCE}$ until this error is less than a threshold, typically set to 10^{-6} [133]. Hence, assuming an empty set of active polynomials, $\mathcal{A}$, and that all PCE coefficients are equal to zero, at each iteration, the algorithm selects the polynomial basis function that exhibits the highest correlation to the OMP residual, *i.e.*, $p' = \arg\max_{p \notin \mathcal{A}} |\langle \mathbf{R}_{\mathbf{OMP}}, \Psi_p(\boldsymbol{\Xi}) \rangle|$, where $\mathbf{R}_{\mathbf{OMP}}$ is the OMP residual and $\langle\rangle$ indicates the inner product in the Euclidean space. Then, the coefficient p' of the basis function is added to $\mathcal{A}$, and its value is estimated through the least squares problem (5.6). Following the computation of the coefficient, the residual is updated as $\mathbf{R}^{=}\mathbf{Y}_{PCE} - \boldsymbol{\Psi}(\boldsymbol{\Xi})\hat{\mathbf{c}}^{\intercal}$; note that the residual initially is equal to $\mathbf{Y}_{\mathbf{PCE}}$. Finally, at each iteration, the leave-one-out (LOO) cross-validation error is computed as per (1.19) in ([133]), and the algorithm runs until LOO

Table 5.4: Input sampling distributions.

	Concrete	Brick	Glass	Plasterboard
ϵ_r	$\mathcal{N}(5.31, 0.66)$	$\mathcal{N}(3.75, 0.46)$	$\mathcal{N}6.27, 0, 78)$	$\mathcal{N}(2.94, 0.36)$
σ	$\mathcal{N}(0.070, 0.008)$	$\mathcal{N}(0.038, 0.004)$	$\mathcal{N}(0.013, 0.001)$	$\mathcal{N}(0.022, 0.002)$

stabilises or falls below the error threshold mentioned earlier.

Once the coefficients are estimated, the PCE constitutes a solution to the UQ problem, portraying the response of the stochastic quantity to some random inputs. In addition, the mean value and the standard deviation (note that a tilde is used over the common notation of the standard deviation, *i.e.*, $\tilde{\sigma}$, in order to distinguish it from the conductivity, σ) of the stochastic quantity can be then estimated as [127]:

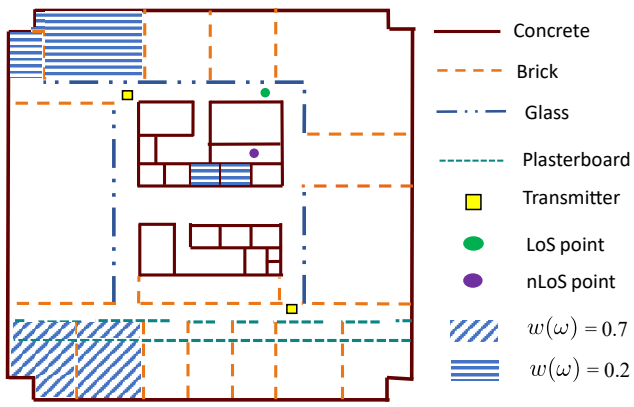
$$\mu[SQ(\boldsymbol{\xi})] = c_0, \tag{5.7}$$

$$\tilde{\sigma}^2[SQ(\boldsymbol{\xi})] = \sum_{k=1}^{P} c_k^2 \langle {\Psi_k}^2(\boldsymbol{\xi}) \rangle. \tag{5.8}$$

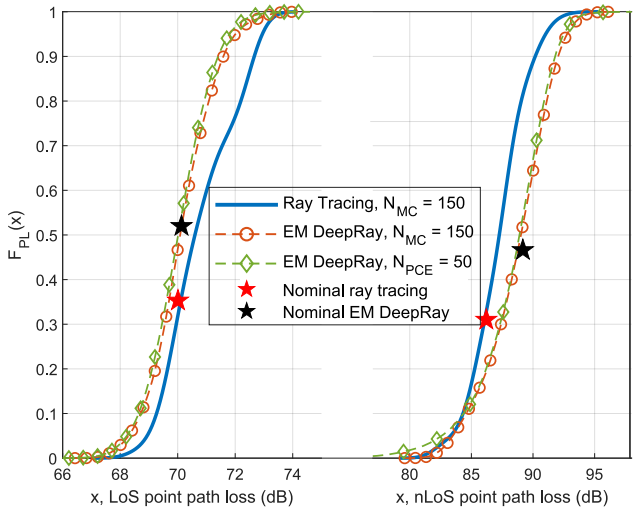
5.3.2 Numerical Results

To quantify the uncertainty in signal attenuation, the geometry shown in Figure 5.7a is generated, assuming that two Txs depicted as yellow square boxes are found within the building. Both Txs use antennas with an omnidirectional beam pattern, and their operating frequency is 2.6 GHz. The channel bandwidth is 5 MHz, the sub-carrier spacing is 15 kHz, and the number of resource blocks is 25. Again, a sole stream exists between the BSs to the user and the scaling factor f_{SC} is set to one. The transmission power for both Txs is set to 0 dBm. Then, for each point within the simulated grid, the Tx with the strongest signal is selected as the serving one, while it is assumed that the two fBSs operate in a different channel and hence they do not interfere. Similarly to Section 5.2.3, a constant noise power spectral density equal to -174 dBm/Hz and a noise figure at the receiving signal chain equal to 5 dB.

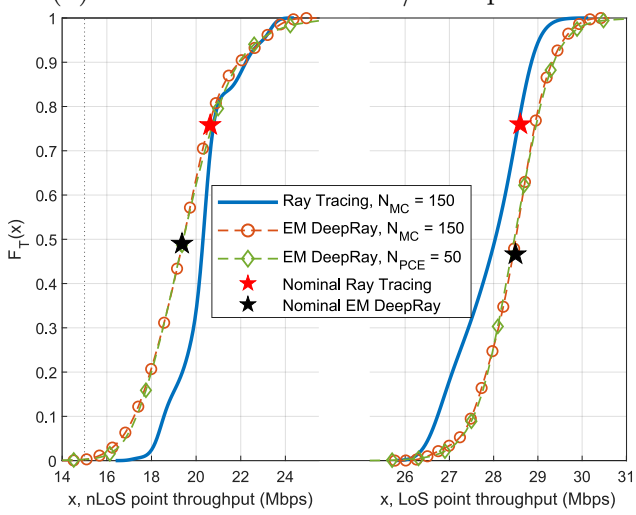
The geometry comprises four different construction materials that are commonly found in indoor environments: concrete, brick, glass, and plasterboard. For each material it is assumed that its electric properties are random variables following the sampling distributions shown in Table 5.4. The mean values of the sampling distributions correspond to the nominal values provided by [106], whilst the standard deviation is set equal to approximately 15% of the nominal value [35]. Hence, there are in total $N_{RV} = 8$ random variables, ξ_i, depicting the uncertainty regarding the two fundamental electromagnetic properties of the four construction materials found within the geometry shown in Figure 5.7a. Thus, the PL distribution, $PL(\omega)$ is a stochastic quantity depending on the material parameters, *i.e.*, $PL(\omega, \boldsymbol{\xi})$, and consequently so are the IWNP metrics.



(a) Geometry under consideration.



(b) Path loss CDFs for LoS/nLos points.



(c) Throughput CDFs for LoS/nLos points.

Figure 5.7: Geometry under consideration, along with the simulated PL and throughput CDFs for a LoS and nLos point.

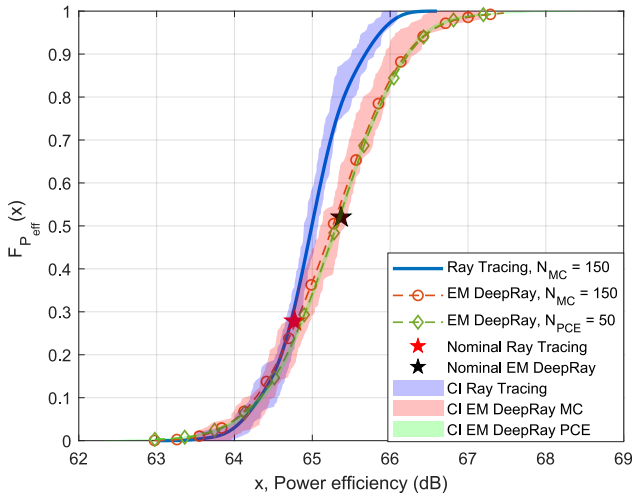
Table 5.5: Statistics of the LoS and nLoS point.

	$\mu \pm \tilde{\sigma}^2$		[min, max]	
	Ray Tracing	EM DeepRay	Ray Tracing	EM DeepRay
PL(LoS) (dB)	70.7 ± 1.4	70.1 ± 1.1	[68.3, 72.8]	[67.5, 73.1]
PL(nLos) (dB)	87.7 ± 2.1	88.8 ± 2.5	[81.8, 93.3]	[82.1, 92.8]
T(LoS) (Mbps)	27.8 ± 0.5	28.4 ± 0.4	[26.7, 29.1]	[26.4, 29.9]
T(nLos) (Mbps)	20.6 ± 1.6	19.4 ± 2.3	[17.1, 23.6]	[15.7, 23.5]

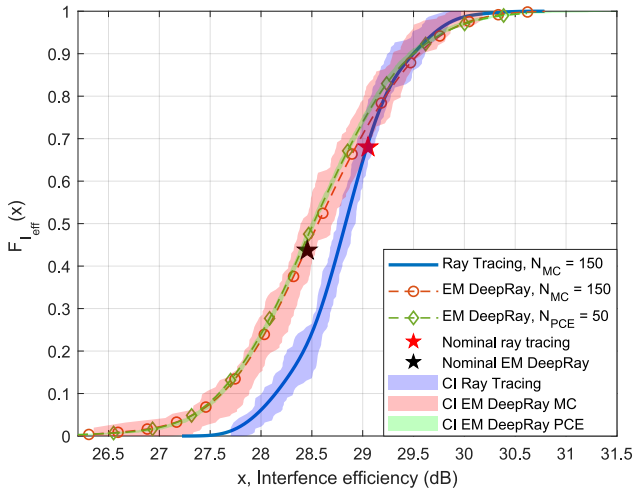
To quantify the uncertainty, $N_{MC} = 150$ ray tracing and EM DeepRay simulations are conducted, assuming different values of ϵ_r and σ, drawn from the input sampling distributions of Table 5.4 using the Latin hypercube sampling method. The pre-trained EM DeepRay yielded an approximately 5 dB RMSE compared to the nominal values ray tracing simulation results. Thus, to have equivalent results in the UQ analysis, we calibrate the pre-trained model, as outlined in [33] to almost match the results of the nominal ray tracing simulation, and then we run the 150 EM DeepRay UQ simulations.

Initially, the impact of the input uncertainty is probed for the simulated PL for a LoS and a non-LoS (nLoS) point, which are depicted in Figure 5.7a as a green and a purple dot, respectively. As can be seen in Figure 5.7b, where the PL for the nominal values of ϵ_r' and σ is shown with an asterisk, the simulated PL can vary due to material parameter uncertainty. The statistics of the UQ analysis are presented in Table 5.5. An interesting observation is that the uncertainty is higher for the nLoS point compared to that of the LoS point, since one can observe that the range of the PL fluctuation is considerably larger. That conforms with intuition, since for nLoS points there are multiple walls interposed between the receiver and the Tx, and thus PL is affected by multiple unknown parameters. Consequently, the throughput is calculated for these two points based on [134], and by mapping the estimated SINR to the modulation scheme and the coding rate according to the values reported in Table 5.1 [135]. As can be observed in Figure 5.7c, the uncertainty in the PL also translates into a variation of the estimated throughput. Hence, the randomness in the material parameters is also important from the perspective of the end-user, as it affects their quality of experience and service.

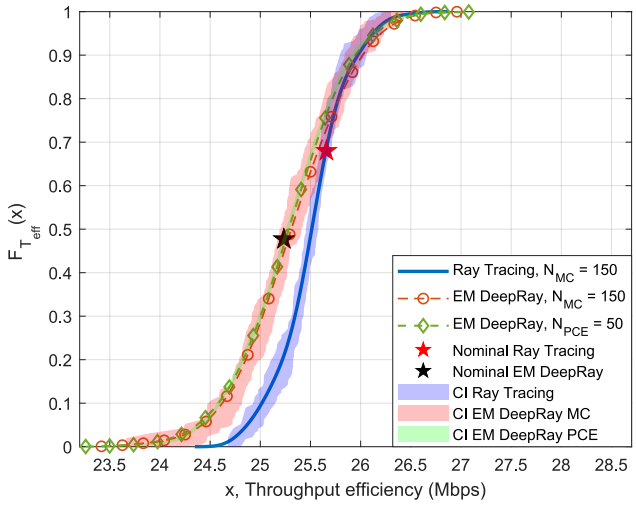
Similarly to the conclusions drawn in Section 5.2.3, the distinct advantage of quantifying this randomness with a data-driven propagation model is the tremendously reduced computational time. A single ray tracing simulation requires 2-3 minutes per Tx, *i.e.*, 4-6 minutes per simulation scenario. On the other hand, EM DeepRay can furnish estimates of the PL distribution for both Txs only within approximately half a second. Thus, conducting a UQ analysis with ray tracing can take up to 10 hours, whereas the same results can be derived with a data-driven model within approximately a minute. The expediency of EM DeepRay does not affect the quality of the results, since as evinced by Figures 5.7b and 5.7c, the



(a) Power effectiveness CDFs and CIs.



(b) Interference effectiveness CDFs and CIs.



(c) Throughput effectiveness CDFs and CIs.

Figure 5.8: CDFs and CIs of the various IWNP metrics for the indoor geometry shown in Fig. 5.7a.

Table 5.6: Statistics of the IWNP metrics.

	$\mu \pm \tilde{\sigma}^2$		[min, max]	
	Ray Tracing	EM DeepRay	Ray Tracing	EM DeepRay
P_{eff} (dB)	64.9 ± 0.5	65.3 ± 0.7	[63.5, 66.1]	[63.4, 67.2]
I_{eff} (dB)	28.8 ± 0.5	28.5 ± 0.7	[27,7, 30,3]	[26.3, 30.3]
T_{eff} (Mbps)	25.5 ± 0.4	25.3 ± 0.5	[24.6, 26.5]	[23.6, 26.5]

cumulative distribution functions (CDFs) for the ray-tracer and EM DeepRay results yield a very good resemblance. It is also observed that the stochastic solution generated from the PCE, with $N_{PCE} = 50$ samples and with highest polynomial order $D = 4$, yields the same CDF as Monte Carlo, however, the number of simulations required to converge is 3 times smaller.

The impact of the uncertainty in ϵ_r' and σ on the signal attenuation will implicitly affect the estimates of the IWNP metrics. The CDFs for the three metrics are presented in Figures 5.8a-5.8c, while Table 5.6 shows their statistics. As can be seen from Figs. 5.8a-5.8c, there is again a close correspondence between the CDFs provided by the ray-tracer and EM DeepRay either using the Monte Carlo method or a PCE-based representation. However, the construction of the stochastic solution through PCE requires only 50 EM DeepRay simulations, *i.e.*, three times fewer than that of the Monte Carlo, whilst the estimation of the sparse PCE coefficients via OMP takes only one second. Additionally, the confidence intervals (CIs) estimated through a PCE representation are tight (visually invisible green area around the dashed green lines) compared to those calculated via the conventional Monte Carlo approach. Furthermore, the mean, the standard deviation, as well as the minimum and the maximum values of the IWNP metrics estimated via ray tracing and EM DeepRay are almost the same. However, as mentioned earlier the UQ analysis of the IWNP metrics with EM DeepRay takes approximately a minute, whilst using ray tracing requires a couple of hours.

5.4 Summary

The results presented in this chapter exemplify the importance of generalisable data-driven propagation models in applications related to wireless network design and operation. The drastic decrease in the time required to infer wireless channel characteristics renders optimisation frameworks leveraging data-driven propagation models a key enabler for the timely installation of indoor cellular and industrial private networks, which are expected to be massively deployed in 5G/B5G systems. In addition to the initial deployment stage, such frameworks can be employed to redesign indoor wireless networks, which are much more agile and likely to be subject to changes compared to legacy outdoor networks. These changes

can be ascribed to modifications in the use of certain building spaces (*e.g.*, deciding to create a conference room or rearranging workplaces), or to additional interference introduced by newly installed outdoor BSs (whose deployment is not that flexible). Furthermore, on top of considering the details of the environment of interest, by probing the variance of electromagnetic properties the use of physics-based data-driven propagation models allows the innate uncertainty that exists within indoor environments to be accounted for, enabling the rigorous quantification of the subsequent uncertainty and the swift computation of wireless channel statistics.

Chapter 6

Characterising Indoor Cellular Network Mobile Service Demands

Understanding mobile network usage is an important task with manifold implications, in networking and beyond. The exploration, characterisation, and modelling of mobile network traffic have a key role in developing and supporting an efficient wireless ecosystem [136]. Indeed, understanding and forecasting traffic demands enables the proactive configuration of the wireless network in order to accommodate the needs of end users and provide them with enhanced quality of service and experience [36, 37]. At the same time, the traffic generated by mobile services has become a primary source of insights on human activities, needs, and habits [137]; this has benefited research in diverse scientific disciplines, ranging from socioeconomic topics like unemployment [138], wealth distribution [139, 140, 141], social inequality and access to digital services and media [142, 143], and to climate change [144, 145]. More recently, suitably processed mobile network traffic has been employed to track and prevent the spread of infectious diseases [146, 147] or to assess their impact [148].

Interestingly, all existing studies on characterising and exploiting mobile traffic invariably focus on measurements collected in outdoor environments. This can be ascribed to the fact that cellular networks have been primarily intended as a technology for mobile, outdoor users. Yet, this is not true anymore, and as discussed in Section 1.1, main actors in the wireless ecosystem forecast that more than 80% of the future cellular data traffic will be generated in indoor environments [16], which has motivated the deployment of native ICNs [16, 17, 18, 19]. To better understand the functionality of these new network components, this chapter sheds light for the first time on the characteristics of the mobile service demands observed in a nationwide ICN deployment, providing novel insights that may support an improved design and operation of ICNs, *e.g.*, via resource allocation, network slicing, caching, or energy adaptation schemes.

6.1 Internet Measurement Data

The measurement traffic data used was collected in the production network of a major MNO providing services throughout a nationwide network in France. To conduct the measurement collection, the operator deploys passive measurement probes to monitor the Gi, SGi, and Gn interfaces that connect Gateway GPRS Support Nodes (GGSNs) and Packet Data Network Gateway (PGWs) to external Public Data Networks (PDNs), gathering data about the traffic generated by mobile subscribers using 4G and 5G connectivity. Indeed, the MNO is presently operating a 5G non-standalone (NSA) deployment, which allows gathering data about both 4G and 5G traffic via measurements limited to the 4G Evolved Packet Core (EPC) network that is shared by 4G eNodeBs and 5G gNodeBs in the Radio Access Network (RAN).

The MNO identifies the mobile service associated with each TCP and UDP session recorded by the probes, by running Deep Packet Inspection (PDI) and analysing the results via proprietary traffic classifiers. Each such IP session is also geo-referenced at the level of Base Transceiver Station (BTS), by exploiting the User Location Information (ULI) field present in the Packet Data Protocol (PDP) Contexts and Evolved Packet System (EPS) Bearers over the GPRS Tunnelling Protocol control plane (GTP-C). Based on this information, it is possible to estimate the traffic demand for each mobile application at an individual BTS, by aggregating the traffic of all IP sessions from/to the same BTS. Such data is aggregated over time within intervals of one hour for the purpose of this study.

Overall, the data used throughout this chapter comprise per-hour downlink and uplink traffic of various mobile services at all indoor cellular antennas managed by the MNO in the target country. This includes $4,762$ ICN antennas installed at more than $1,000$ sites, comprising different types of indoor environments located at an urban, suburban, or rural surrounding. The vast majority of those antennas are 4G, as apparently 5G is very scarcely used for ICN at this stage of the roll-out of the technology in France. The recording period was approximately two months, starting from the 21st of November 2022 and extending until the 24th of January 2023. The mobile services considered span a diverse range of mobile applications used throughout daily life related to activities such as social networking, messaging, audio and video streaming, transportation, professional activities, and well-being.

Regarding the ethical considerations related to data collection and processing, the traffic measurements used to compute the measures introduced in Section 6.2.1 were gathered by the operator for network management and research purposes, and temporarily stored within a secure platform at their own premises. The hour-level aggregation was also carried out in the same platform by personnel of the network operator, in full compliance with Article 89 of the General Data Protection Regulation (GDPR) [149] of the European Commission. The data collection and processing were approved by the Data Protection Officer (DPO) of the operator, and authorised by the French National Commission on Informatics and Liberty (CNIL), within the context of a collaborative research project. Additionally, it should

be noted that the original network measurements contained personal identifiers (*e.g.*, the International Mobile Subscriber Identifier, or IMSI) and sensitive data (*e.g.*, locations of visited cells, or mobile services consumed) about individual users, and were deleted upon aggregation. Therefore, the aggregated traffic demands used (over the two-month period or the hourly patterns presented in Section 6.5) do not contain personal identifiers or sensitive information. The level of spatiotemporal aggregation ensures that no data subject can be re-identified and that the traffic data or the metric used for clustering is not configured as personal data in the GDPR acceptance.

For the purpose of this work, access was provided only to such aggregated and privacy-preserving traffic data for the purpose of carrying out the ICN traffic study. Ultimately, the dataset and research do not involve risks for the mobile subscribers, while they provide new knowledge about the dynamics of indoor mobile traffic demands, which will benefit an improved design and more dependable validation of technical solutions for mobile network operations.

6.2 Classifying ICN Behaviours

This section discusses the methodology followed to study the dynamics of ICN traffic and unveil the distinct patterns that reside in the data set described in Section 6.1.

6.2.1 Measuring relative ICN service usage

To probe the dynamics of the ICN mobile traffic demands across the network, an unsupervised learning approach is pursued. In particular, the aggregated sum of the downlink and uplink traffic recorded for each mobile service over the target two-month period is considered as a distinct feature, and a matrix, $T^{N_{ant} \times M_{app}}$, is formed, comprising the overall traffic in megabytes (MBs) for the N_{ant} indoor antennas and M_{app} applications included in the data set. In particular, for the rest of this study, $N_{ant} = 4,762$ indoor antennas and $M_{app} = 73$ mobile services are considered.

The goal is to pass this matrix to an unsupervised learning algorithm, which will be able to identify patterns and existing hidden structures within the traffic data, without further guidance. However, the overall traffic is not a defining feature *per-se*, and it will inevitably instigate bias in the clustering algorithm for two reasons. First, some applications intrinsically produce a larger volume of traffic than others, *e.g.*, streaming services generate demands that can be orders of magnitude larger compared to those induced by texting applications. Second, due to their location, antennas are also expected to serve highly heterogeneous traffic volumes. Therefore, clustering directly the aggregated traffic would result in overlooking the impact of many services and essentially grouping together antennas according to their

popularity. Hence, it would not be possible to quantify the diversity of application usage across the various antennas.

To overcome these limitations, a more representative and fair measure is considered: the revealed comparative advantage (RCA), which was initially employed in the field of international economics [150]. The RCA is an index of the relative advantage or disadvantage of a specific sample in a certain category. In the context of this research, RCA quantifies the degree of over- or under-utilisation of a certain service at a specific antenna. Given the matrix T, the RCA per application for each antenna can be straightforwardly computed as:

$$RCA_{i,j} = \frac{T_{i,j}/T_i}{T_j/T_{tot}}, \tag{6.1}$$

where $T_{i,j}$ stands for the traffic recorded for the j-th service at the i-th antenna, T_i refers to the total traffic generated at the i-th antenna for all the services, T_j depicts the summed traffic over all antennas for the j-th service, and T_{tot} is the total traffic channelled through the network during the entire data collection period.

Values of RCA below 1 indicate that an antenna is disadvantaged in the use of a certain service with respect to the other antennas, *i.e.*, there is an under-utilisation of the service by that antenna, while values higher than 1 suggest that an antenna is advantaged, *i.e.*, there is an over-utilisation for the same service. Evidently, from the definition of RCA it follows that while under-utilisation has a finite lower limit equal to to 0 (when an antenna hasn't recorded traffic for a service), over-utilisation is unbounded and ranges from 1 to infinity. That is a well-known shortcoming of RCA that can lead to misleading results and curb the performance of the unsupervised learning clustering algorithm. A remedy for that was discussed in [151, 152], by making the index symmetric, thus enabling an equitable comparison between the under- and over-utilisation intervals. In particular, Laursen and Engedal introduced the revealed symmetric comparative advantage (RSCA) which is defined as follows [151]:

$$RSCA_{i,j} = \frac{RCA_{i,j} - 1}{RCA_{i,j} + 1}, \tag{6.2}$$

from which it emerges that the values of RSCA lie within the $[-1, 1]$ interval, with values below 0 yielding under-utilisation and values higher than 0 implying over-utilisation.

Figure 6.1 shows the histogram of the (*i*) normalised traffic (over the maximum application load observed among all selected antennas), (*ii*) RCA, and (*iii*) RSCA of the M mobile services for some antennas of the data set. As it is normalised by the application with the highest load observed at any antenna, the normalised traffic presents a spike-like behaviour with most applications having values squeezed close to 0, due to their inherently lower produced load or infrequent use, as well as owing to the less central location of the antenna. A closer look can be obtained when zooming between the 0 and 0.5 interval (upper right

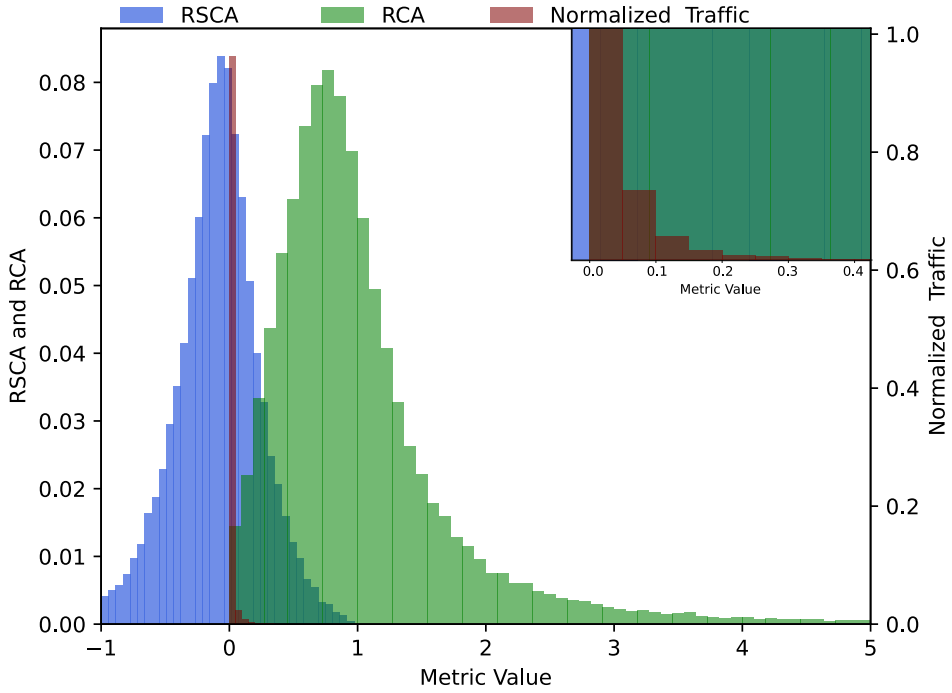


Figure 6.1: Distribution of the Normalized traffic, the RCA, and the RSCA for the M mobile services of a single antenna.

corner of Figure 6.1), where it can be seen that the number of popular mobile services with higher load drops extremely quickly. Ultimately, the presence of highly imbalanced demands across antennas and services makes a straightforward normalisation by the global maximum not viable, as it tends to hide the preponderant low-traffic samples.

On the other hand, the RCA values are clearly better (*i.e.*, more diversely) distributed for the same set of samples; however, one can observe that the distribution is skewed, with the underutilised services wedged between 0 and 1, whereas the over-utilised services RCA values span from 1 to beyond 5 (specifically, the largest RCA in this example is 75.88). Thus, RCA still includes outliers, designated by the histogram tail, which are precisely the points that can jeopardise the reliability of the clustering by drawing the multidimensional cluster barycenters towards cases with higher-than-average usages. Such a behaviour is removed from the data in the RSCA distribution, which yields a properly balanced distribution among samples that show lower- and higher-than-usual application consumption. Hence, in the following clustering analysis, we select the RSCA of each application as a distinct feature, separating the data based on the utilisation profile of the M_{app} mobile services across the N_{ant} distinct antennas.

6.2.2 Clustering ICN antennas

With the proper transformation of the recorded traffic defined, the next step is to feed the processed data to an unsupervised learning algorithm. The aim is to identify clusters of ICN

antennas based on similarities and differences in the considered set of futures, *i.e.*, the RSCA values that denote under- or over-utilisation of the M_{app} applications.

While multiple techniques are available for unsupervised learning clustering, due to its comprehensibility agglomerative clustering is selected, which is a state-of-the-art hierarchical clustering algorithm [153]. The algorithm follows a bottom-up approach, assuming initially that each data point belongs in its own cluster, and thereafter merging the clusters greedily according to a specified criterion. In particular, Ward's criterion is utilised [154] which aims at minimising the total intra-cluster variance, measured as the squared distance between the cluster centres, when merging two clusters. Hence, the clustering algorithm employed starts from N_{ant} distinct clusters and repeatedly merges the clusters, thus reducing their number, such as the new cluster yields a reduced intra-cluster variance.

During this process, an important point is to determine when the merging should stop, *i.e.*, what is the number of the total clusters, k_c, that separates optimally the data. To quantify the performance of the clustering algorithm and determine the most suitable number of clusters, the Silhouette score and the Dunn index are employed [155, 156]. Qualitatively, the first expresses the degree of similarity among objects of the same cluster compared to other clusters, while the latter indicates how compact and well-separated the clusters are. Quantitatively, the Silhouette score, SC, for the n_i-th antenna found in cluster C_i, with size $|C_i|$, is defined as [155]:

$$SC_{n_i} = \frac{ch(n_i) - sp(n_i)}{\max(ch(n_i), sp(n_i))} \tag{6.3}$$

where $ch(n_i)$ depicts the cohesion, *i.e.*, how close/similar is antenna n_i to all the other antennas, n_j, in cluster C_i, $ch(n_i) = \sum_{n_j \in C_i} d_{Eu}(n_i, n_j)/(|C_i - 1|)$, and $sp(n_i)$ represents the separation, *i.e.*, how far/different is antenna n_i from all antennas, n_n, in the nearest cluster C_n, $sp(n_i) = \min_{C_n \neq C_i} \sum_{n_n \in C_n} d_{Eu}(n_i, n_n))$, where d_{Eu} is the euclidean distance (note that this is mobile service utilisation "distance", not physical distance). The aggregated overall Silhouette score for a clustering instance with k_c clusters is then computed by averaging the individual Silhouette scores of all antennas.

The Dunn index, DI, is defined as the ratio between the minimum inter-cluster distance, d_{inter}, and the maximum intra-cluster distance Δ_{intra} [156]:

$$DI = \frac{\min\limits_{0 \leq k_1 < k_2 \leq k_c} d_{inter}(C_{k_1}, C_{k_2})}{\max\limits_{0 \leq k_m \leq k_c} (\Delta_{intra}(C_{k_m}))} \tag{6.4}$$

where the inter- and intra-cluster are computed as the average distance between the antenna features in different clusters and the average distance between all antennas within the same cluster, respectively.

To identify the most suitable number of clusters, one should identify a high value of the

Silhouette score or the Dunn index, followed by an abrupt drop, which suggests a substantial deterioration of the intra- and inter-clustering quality. As can be seen from Figure 6.2, such a behaviour can be observed for $k_c = 6$ and $k_c = 9$. Hence, in what follows $k_c = 9$ is selected, which exhibits the steepest drop for both metrics, while the differences that occur for a reduced number of clusters $k_c = 6$ are discussed qualitatively.

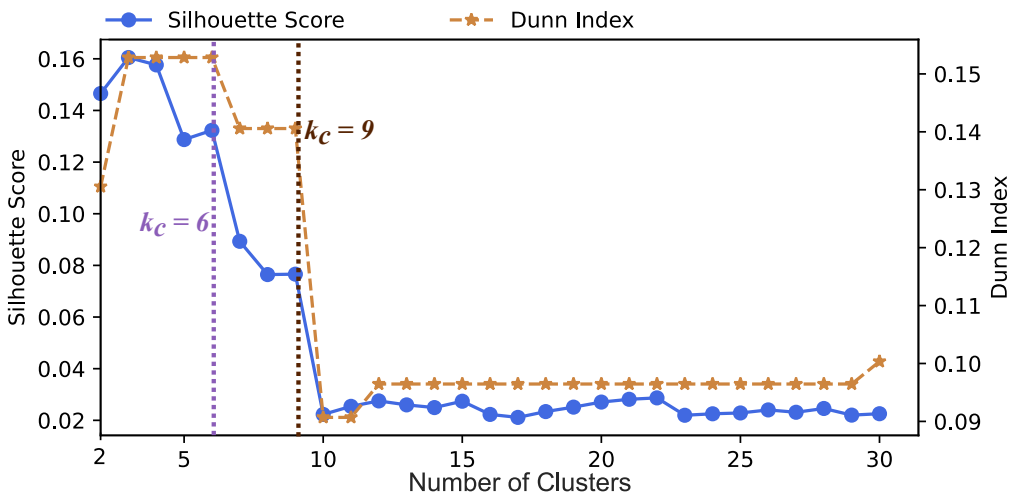


Figure 6.2: Silhouette score and Dunn index versus the number of clusters, serving as a stopping criterion to select the optimal number of clusters.

The clustering results based on the mobile service RSCA are summarised in Figures 6.3 and 6.4. Figure 6.3 depicts the complete hierarchy returned by the agglomerative algorithm on the service RSCA values, starting from the individual antennas at the bottom and reaching a single cluster at the top. The optimal clustering identified by $k = 9$ is also highlighted, with the set and number of antennas in each final cluster reported along the x-axis of the dendrogram. Figure 6.4 presents a complementary RSCA heatmap, clumping the antennas per cluster, and presenting the RSCA of all the indoor antennas for each service. Blue lines in the heatmap suggest over-utilisation of a certain service across the antennas, while the opposite holds for dark red lines. Evidently, it can be observed that RSCA follows a distinct visual pattern for each cluster, *i.e.*, the indoor antennas appertaining to the same cluster manifest the same pattern with respect to the service utilisation. The RSCA pattern can be substantially different compared to that of the other clusters, but there are also some clusters between which the differences are soothed.

To acquire further insight regarding the degree of similarity between the 9 clusters, one can peruse the hierarchical algorithm dendrogram in Figure 6.3, which includes two horizontal lines to indicate the distance threshold that distinguishes the clusters for $k_c = 6$ and 9. From the dendrogram, and considering that dissimilarity in terms of service usage is associated with distances along the y-axis, it emerges that there are three large groups of clusters (ending at the edge of the blue lines), which are further separated into three sub-clusters coloured orange, green, and red.

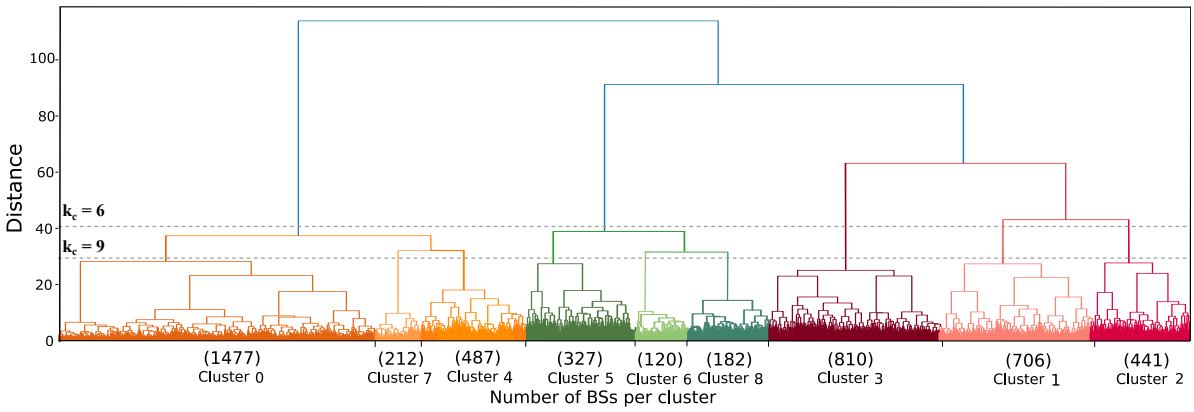


Figure 6.3: Dendrogram illustrating the iterative merging of antennas into clusters as returned by the hierarchical agglomerative algorithm run on RSCA features of individual antennas. Distance thresholds for $k_c = 6$ and $k_c = 9$ are highlighted. Colours tell apart the 9 clusters identified by the second threshold.

The first larger group comprises clusters 0, 7, and 4 (orange group), the second clusters 5, 6, and 8 (green group), whilst the third includes clusters 3, 1, and 2 (red group). The clusters found within the same group present a stronger similarity with each other, which is also corroborated by the RSCA heatmap in Figure 6.4, whilst they exhibit unalike mobile service utilisation patterns compared to the clusters of the other two groups. Out of the three groups, the orange one presents the more distinctive behaviour, found on a separate branch of the dendrogram. Furthermore, within each group, clusters found under the same branch yield a higher resemblance to each other compared to the clusters found in a separate branch, *e.g.*, clusters 1 and 2 for the red group. Finally, it should be noted that applying the hierarchical clustering with $k_c = 6$, corresponds to consolidating the clusters of the orange group into a single cluster, instead of diving it into 3 sub-clusters, and merging clusters 6 and 8 at the second branch of the green group.

Although the dendrogram allows identifying groups of clusters that are closer to each other, it cannot be used to delve into the feature importance of each cluster. Hence, an intriguing question that arises from the present analysis is: *which are the features, i.e., services, that impact the clustering decision the most, and consequently characterise each cluster?* To answer this question, the next sections discuss which are the user tendencies and mobile service utilisation patterns within each cluster that render them unique.

6.3 Understanding ICN Traffic

This section probes how explainable AI techniques can be leveraged to interpret the results of the previous clustering analysis and breakdown the application trends per cluster, unveiling that these trends are correlated with the type of indoor environment in which they occur. An alternative solution to using the SHAP framework to comprehend clustering results would be

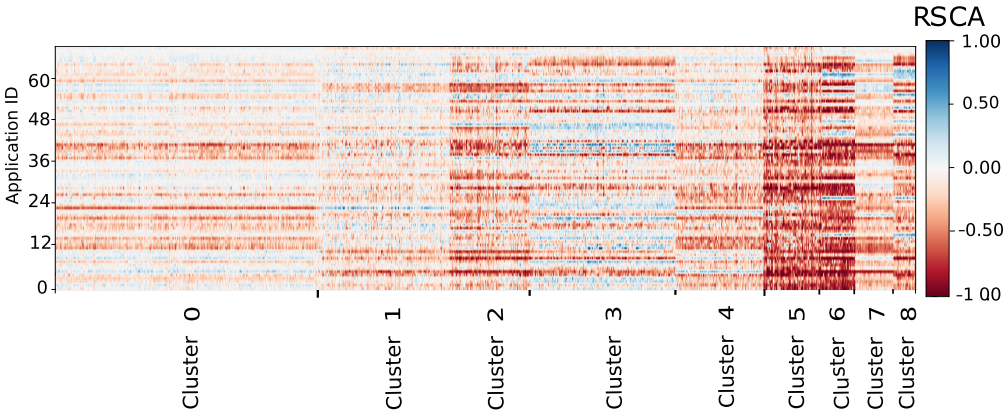


Figure 6.4: Heatmap of the mobile service RSCA (y-axis) for the clustered ICN antennas (x-axis), when $k = 9$.

to attempt to pronounce the influence of different services based on the service distance from the cluster centroids. The centroid position would indicate the usage disposition towards a mobile service by an antenna, *i.e.*, under- or over-utilisation depending on whether the centroid RSCA for this service is negative or positive, respectively. In addition, observing the average per service distance, one could deduce the service influence on the clustering decision; services with smaller distances from the centroid would be more important and vice versa. It is noted that while this approach was also explored, and yielded results that conformed with the SHAP-based clustering analysis, the latter aligned better with intuition and thus for brevity only the SHAP-based analysis is presented in the following subsections.

6.3.1 Interpreting Clustering Results

To interpret ICN antenna clustering results, first, the SHAP framework introduced in Section 2.4.1, is applied in the clustering algorithm of Section 6.2.2, and it is discussed how the results can help in understanding the under- and over-utilisation of mobile services in each cluster.

6.3.1.1 Interpreting Clustering with SHAP

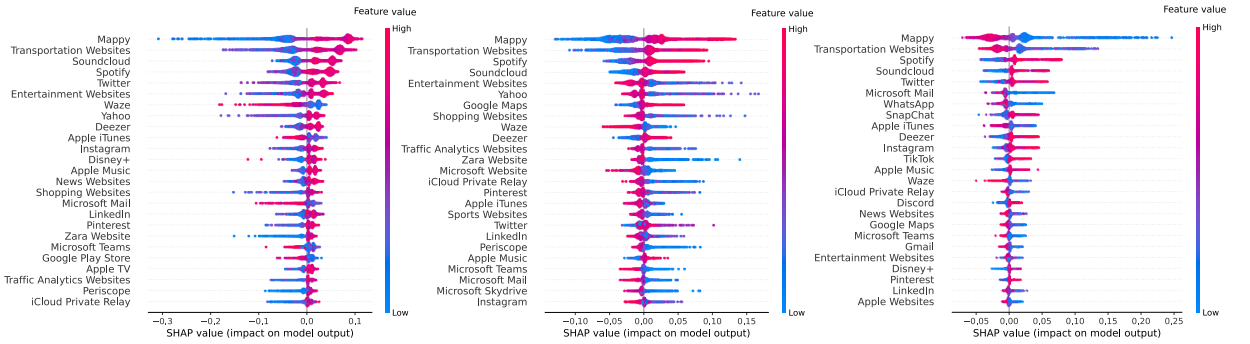
In this section, the SHAP framework is leveraged to interpret the agglomerative clustering decisions presented in Section 6.2.2 and shed light on the traffic demands of each cluster. The agglomerative algorithm forms the clusters by gradually linking samples together based on their distance from all the other samples, *i.e.*, the clustering is the result of consecutive iterations over the entire data set. It is important to note that the clustering is performed considering concurrently all the samples: therefore, the result cannot be used to infer the cluster of a single sample from the data set by solely observing its input features, or to predict the cluster of new samples not included in the data set. This implies that once the clustering is completed, there is no black-box ML response function f_{ML} to be interpreted.

A common approach to overcome this limitation is to build a surrogate supervised learning classifier on top of the unsupervised learning clustering results, treating the clusters as the target labels associated with each sample of the supervised learning problem [157, 158, 159, 160, 161]. The supervised learner is typically a decision trees-based model, and once it is trained to approximate the clustering results, one can instead interpret the predictions of the supervised learning classifier. Thence, considering as ground truth labels the clusters of Section 6.2.2, a random forest classifier with 100 trees is trained to infer the antenna cluster based on the mobile service RSCA, and it is then amalgamated with the SHAP framework presented earlier. That also allows the expediency of the TreeShap method to be exploited for the estimation of the Shapley values [69].

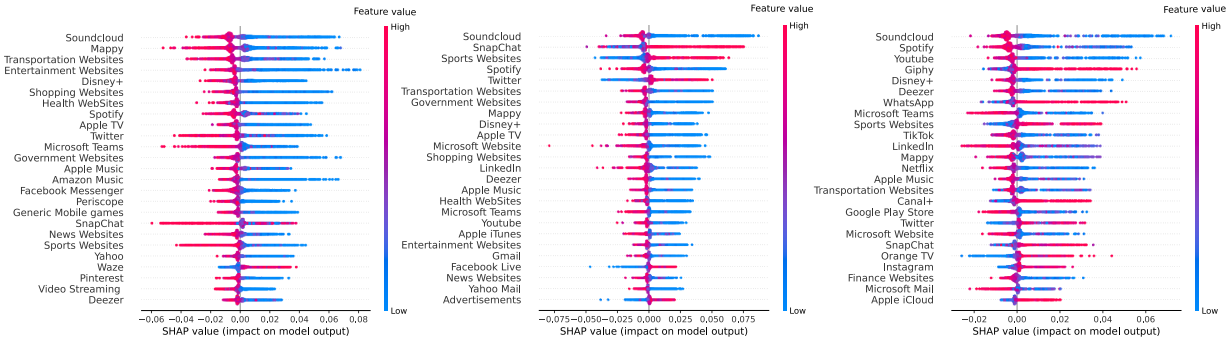
With the Shapley coefficients evaluated, one can inspect their values to comprehend the importance of each service in the cluster decision-making process, as well as whether the importance springs from the under- or over-utilisation of the service. The computed coefficients for each cluster are presented as beeswarm plots in Figs 6.5a to 6.5i, where the red and blue colour implies that the coefficient is associated with higher and lower feature values, respectively. A positive SHAP value associated with a high feature value signifies a positive impact on the prediction, leading the model to categorise the antenna in the respective cluster due to the corresponding application over-utilisation. Conversely, positive coefficients, combined with low feature values, indicate that the antenna belongs to the respective cluster if there is under-utilisation of a certain application.

To facilitate the analysis and the comprehension of the clustering criteria, the results are presented according to the groups identified in the dendrogram of Figure 6.3. The Internet services in each beeswarm plot are ranked in descending order according to their impact on the model's output. The impact of each service is measured as the mean absolute value of the service coefficients, *i.e.*, applications with high coefficient values influence cluster inference more. For each cluster, the 25 most influential services are presented. Remarkably, it can be observed that there are similarities in the mobile service utilisation patterns for the clusters falling under the same colour groups discussed in Section 6.2.2. For instance, the antennas of the orange group share in common that they over-utilise applications related to music, such as Spotify, Soundcloud, Deezer, and Apple Music. Likewise, applications related to Navigation, such as transportation websites, Mappy, and Google Maps, have a strong presence in clusters 0 and 4, and that in fact distinguishes them from cluster 7, which is characterised by under-utilisation of these applications. Finally, unlike cluster 0, the utilisation of entertainment services is scarce in cluster 4, *e.g.*, Yahoo and entertainment, shopping, or sports websites, which explains their separation into different clusters.

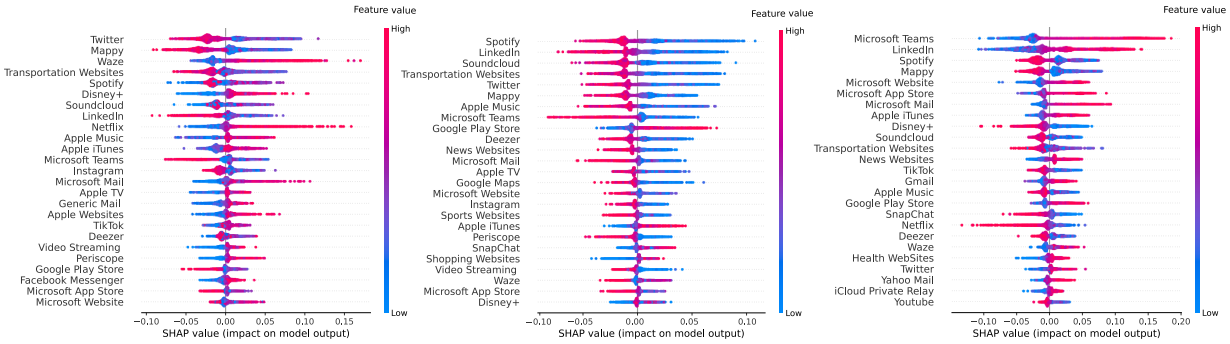
On the other hand, the main similarity between the green group clusters is that they exhibit strong under-utilisation for most of the mobile services. That is more apparent in cluster 5, which constitutes a distinct leaf in the dendrogram, whereas clusters 6 and 8 both



(a) Orange group, cluster 0. (b) Orange group, cluster 4. (c) Orange group, cluster 7.



(d) Green group, cluster 5. (e) Green group, cluster 6. (f) Green group, cluster 8.



(g) Red group, cluster 1. (h) Red group, cluster 2. (i) Red group, cluster 3.

Figure 6.5: SHAP values beeswarm plots for clusters 0 to 8, depicting the importance of each application in the clustering decision. The group colour refers to the branch colour of the dendrogram in Figure 6.3. Positive Shapley values matched with high feature values imply that for an antenna to belong to that cluster it should be overutilising the respective application, while low feature values with positive coefficients suggest under-utilisation.

belong in another branch. Moreover, both clusters 6 and 8 include antennas that over-utilise Snapchat, Twitter, and Sport websites. However, cluster 8 exhibits a larger diversity in terms of applications, which renders it unique: indeed, services such as Giphy, WhatsApp, and streaming such as Canal+ are absent in cluster 6.

The red group exhibits a substantially different behaviour compared to the orange group, as clusters 1, 2, and 3 demonstrate minor utilisation of music and navigation-related applications, including Spotify, SoundCloud, Mappy, and Transportation websites. Moreover, cluster 3, which is a distinct leaf in the dendrogram, deviates from clusters 1 and 2, since it is characterised by intense use of business-oriented applications, such as Microsoft Teams, LinkedIn, and emailing services. On the other hand, cluster 1 presents over-utilisation of diverse application types such as streaming services (Netflix, Disney+, Amazon Prime Video), vehicular navigation (Waze), and mailing applications. Finally, cluster 2 is characterised by services related to digital distribution services (Google Play Store) and Shopping websites.

The application utilisation analysis corroborates the sanity of the unsupervised learning approach followed in Section 6.2.2. Indeed, it is now evident that clusters found within a small distance, *i.e.*, in the same group, yield small differences in their mobile traffic demand patterns, whilst the trends among clusters of different groups are substantially different. Despite that though, an intuitive explanation and the roots of this phenomenon are still unidentified.

6.3.2 Indoor Environment Analysis

The data provided by the MNO is accompanied by additional information related to the antennas, such as their geographical location, the location area code, and the name of the BS. By inspecting the names of the antennas, applying simple string manipulation to extract keywords appearing within the names, and cross-checking the respective antenna location online, it was possible to identify specific types of indoor environments in which ICNs are deployed. In particular, the indoor locations are divided into eleven categories: (i) 17794 metro stations, (ii) 434 train stations, (iii) 187 airports, (iv) 774 workspaces, (v) 469 commercial centres and shopping stores, (vi) 451 stadiums, (vii) 230 expo centres, (viii) 28 hotels, (ix) 53 hospitals, (x) 220 tunnels, and (xi) 122 public buildings.

Consequently, apart from allocating the indoor BS to clusters, it is possible to associate them with a specific environment type. A logical next step is to quantify the type of indoor environments that reside within each cluster. A qualitative illustration of the correlation between the detected clusters and the indoor environment type is shown in the Sankey diagram of Figure 6.6, depicting how the samples of each cluster flow into the identified environments. Conspicuously, there are intense streams between certain clusters and environment types. For instance, it can be seen that the metro and train stations are monopolised by the orange group clusters, while the preponderance of stadiums appertains within one of the green

group clusters. Furthermore, it can be observed that the dominant flux towards workspaces originates from cluster 3, whilst clusters 1 and 2 populate the remaining environments.

A quantitative analysis is presented in Figure 6.7, which illustrates the percentage of each cluster occupied by the different environment types. In addition, Figure 6.8 reports how the different environment types are distributed among the various clusters for some non-obvious cases. As mentioned, a striking observation is that the orange group clusters (0, 4, and 7), shown in Figure 6.7a, comprise solely metro and train stations. That further rationalizes the mobile service utilisation patterns discussed in Section 6.3.1.1, and correlates them with the habits and the activities in which the users are engaged in relation to the environment in which they are found. Indeed, the orange group represents antennas that serve users while commuting, which explains why they are likely to listen to music, use navigation applications to make their way or check public transportation schedules, and be involved with entertainment-related applications while travelling. Notably, for clusters 0 and 4 more than 92% of the antennas are located in Paris and its suburbs, contrary to cluster 7 which consists solely of the Lille, Lyon, Rennes, and Toulouse metro antennas, *i.e.*, non-capital cities.

This further clarifies the distinction between the antennas of the orange group: non-popular navigation applications, such as Mappy and Transportation websites, are likely to fall in under-utilisation when compared to the usage patterns of metropolitan commuters. Indeed, the RSCA does relate to the absolute volume of data, rather it captures the peer-service usage between different antennas. Moreover, the comparative under-utilisation can be also attributed to the complexity of the Parisian metro system compared to that of smaller cities, *i.e.*, a single route might entail multiple station and line changes that are more complex to navigate and need the support of dedicated apps. Finally, the fact that Paris is a tourist destination fosters the use of navigation services by non-resident individuals.

Another interesting remark is that the preponderance of the antennas found in the green group is stadiums, as can be seen from Figures 6.7b and 6.8c. That justifies the high number of visits to sports websites and the use of content-sharing applications, such as Twitter and Snapchat via which one can upload photos and information relevant to sports events. This behaviour is more evident for the antennas in clusters 6 and 8, out of which more than 75% is occupied by stadiums. Remarkably, cluster 6 includes stadiums outside Paris, while approximately 60% of cluster 8 antennas are in Paris. On the other hand, stadiums make up only 35% of cluster 5, which also includes other diverse types of environments, such as expo centres, corporate offices, and commercial centres, equally distributed in Paris and other cities. Apart from the expo centres though, the remaining categories represent only a small fraction of their environment type, *e.g.*, only approximately 5% of the commercial centres and working environments belong to cluster 5, as can be seen from Figures 6.8a and 6.8c, respectively. Hence, given that cluster 5 is characterised by the under-utilisation of most

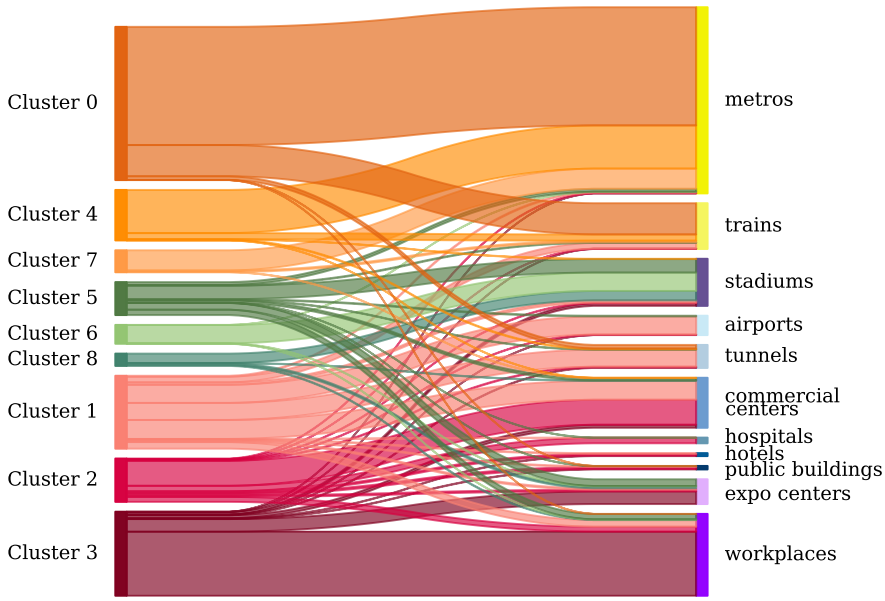
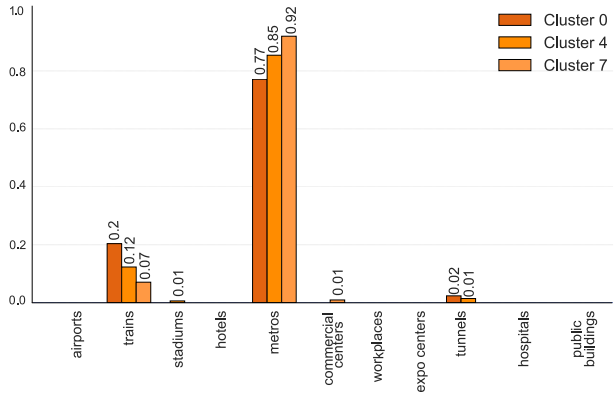


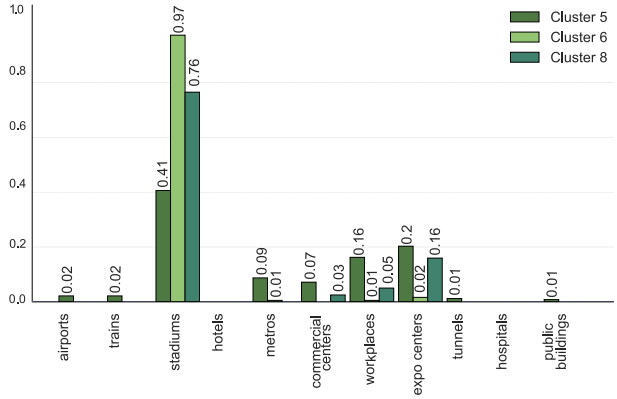
Figure 6.6: Sankey diagram depicting how the clusters flow into different environment types.

mobile services, it is considered to include antennas treating most of their Internet services equally, without demonstrating any advantages compared to other antennas in the data set. In other words, service usage is equally distributed at those antennas, yielding a similar, but small, numerator for all services in (6.1), compared to a larger denominator. Investigating the environments encountered in the red group, the most apparent connection is observed for cluster 3. As can be seen, more than 70% of cluster 3 antennas are workplaces, *e.g.*, corporate offices or industrial facilities, which clearly provokes a surge in the use of services such as Microsoft Teams, LinkedIn, and emailing applications. Expo centres also assume a strong presence, with more than 50% of them belonging to cluster 3 as seen in Figure 6.8c, likely due to holding corporate events, conventions, and conferences.

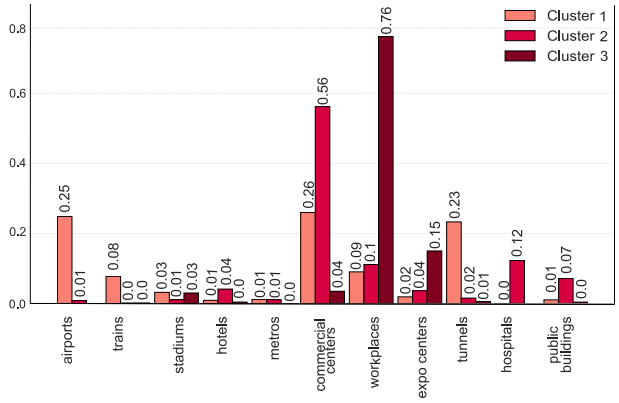
Interestingly, while cluster 3 constitutes a single branch of the red group, clusters 1 and 2, which belong to the other branch, host many commercial centres. In particular, cluster 2 hosts 50% of the commercial centres, most of the hotels and public buildings, as well as almost all the hospitals of the data set, as shown in Figure 6.8b. Notably, cluster 2 embodies all the small retail shops of the MNO, which clarifies its advantage in the use of Google Play Store for service provision. Cluster 1 is more diverse, and along with commercial centres, it also contains almost all airport and tunnel antennas, as illustrated in Figure 6.8a, and a small percentage of all the available environment types. Given the large variety of environments encountered in cluster 1 and the absence of a dominant environment type, as well as due to the fact that it contains messaging, streaming, and music services, it can be speculated that this serves as a general-use cluster. As a final note, we mention that while the antennas of cluster 1 are almost equally distributed between Paris and other cities, around 92% of the antennas of cluster 2 are found outside Paris, while 70% of cluster 3 antennas are located in



(a) Orange group.

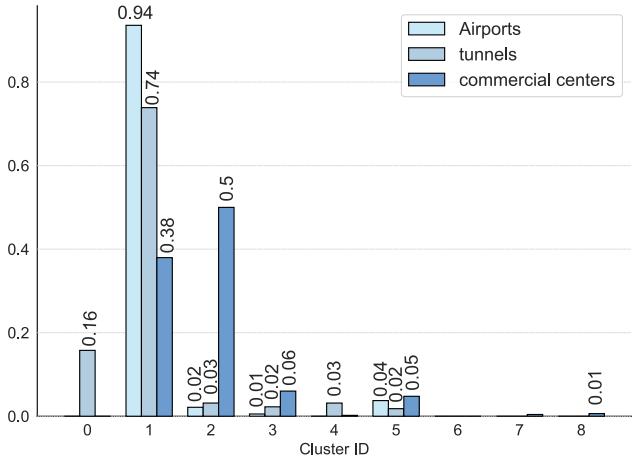


(b) Green group.

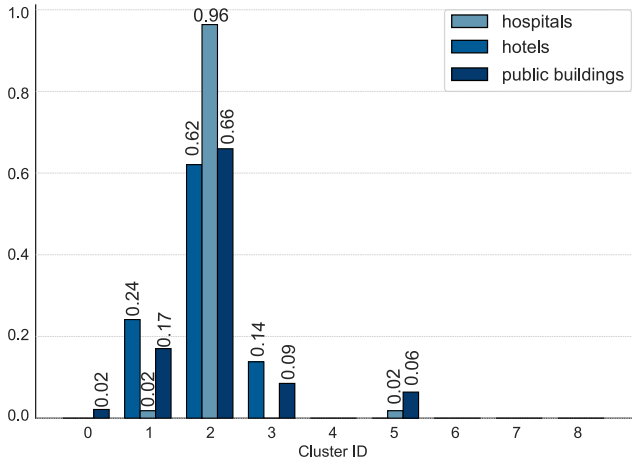


(c) Red group.

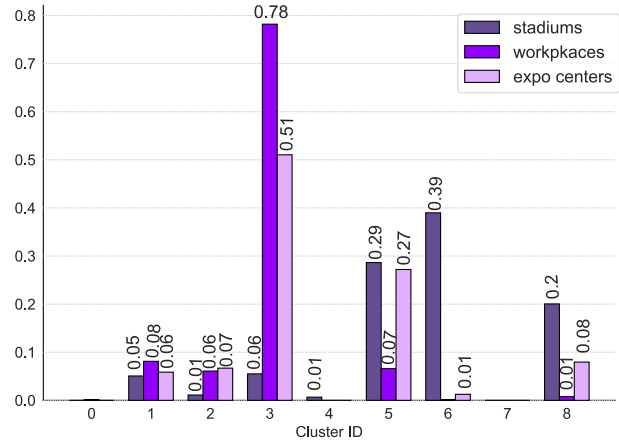
Figure 6.7: Types of indoor environments per cluster



(a) Airports, tunnels, and commercial centres distribution.



(b) Hotels, hospitals, and public buildings distribution.



(c) Stadiums, expo centres, and workplaces distribution.

Figure 6.8: Cluster distribution per indoor environment type.

Paris and its suburbs.

6.4 Comparison with Outdoor Antennas

From the analysis presented earlier, it clearly emerges that ICN mobile traffic demands intrinsically exhibit distinct patterns that highly depend on the type of indoor environment. A natural question is whether these patterns are also present in the components of legacy communication system radio access networks. This question can be answered, by probing the traffic generated by outdoor antennas found in proximity to the ICNs of the data set. In particular, the aim is to assess whether the nature of the ICN strictly defines and differentiates its service demands from that of close-by outdoor antennas, *e.g.*, if the service demands of the outdoor antennas nearby to corporate offices are drastically disparate from the in-building ones. Hence, for each indoor antenna, all the outdoor antennas found within a 1-kilometre radius are considered, and their RCA is computed as:

$$RCA_{out_{i,j}} = \frac{T_{out_{i,j}}/T_{out_i}}{T_{in_j}/T_{tot,in}}, \tag{6.5}$$

where $T_{out_{i,j}}$ represents the traffic recorded for the j-th service at the i-th neighbouring outdoor BS, T_{out_i} is the total traffic generated at the i-th outdoor BS for all the mobile services, while the ratio $T_{in_j}/T_{tot,in}$ expresses the level of utilisation the of j-th service over the entire indoor traffic. Then, the RSCA for the outdoor antennas can be computed straightforwardly via (6.2). It should be noticed that according to (6.5) the RCA for the outdoor antennas measures the level of utilisation for a certain service in an outdoor antenna compared to the level of the same service usage among all the indoor antennas. Indeed, rather than studying the traffic observed at outdoor antennas *per se*, the goal is to explore whether this traffic is innately different compared to that generated in indoor environments.

With the RSCA for all the neighbouring outdoor antennas estimated, their cluster can be inferred by feeding these values to the trained random forest classifier used in Section 6.3.1.1 to surrogate and generalise the unsupervised learning clustering results. The predicted cluster distribution for approximately 20,000 neighbouring outdoor antennas is presented in Figure 6.9. Evidently, the innate diversity encountered in the traffic demands of indoor antennas is absent for outdoor antennas. Indeed, almost 70% of the outdoor antennas appertains in cluster 1, which further corroborates the intuition that cluster 1 constitutes a general-use cluster. More importantly, though, it becomes clear that the distinct traffic behaviour of workplaces, stadiums, metro, and train stations is now almost absent, as only a negligible percentage of the outdoor antennas lies within the respective clusters.

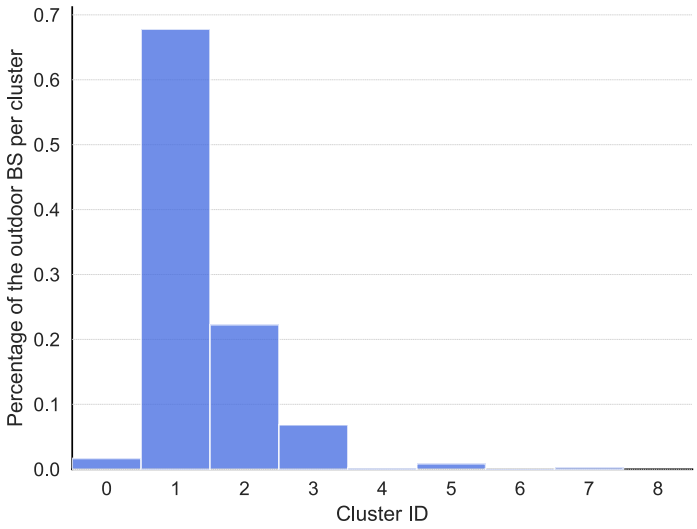


Figure 6.9: Distribution among the identified clusters, for 22,000 outdoor antennas located in close proximity of the ICN antennas included in the data set.

6.5 Temporal Analysis

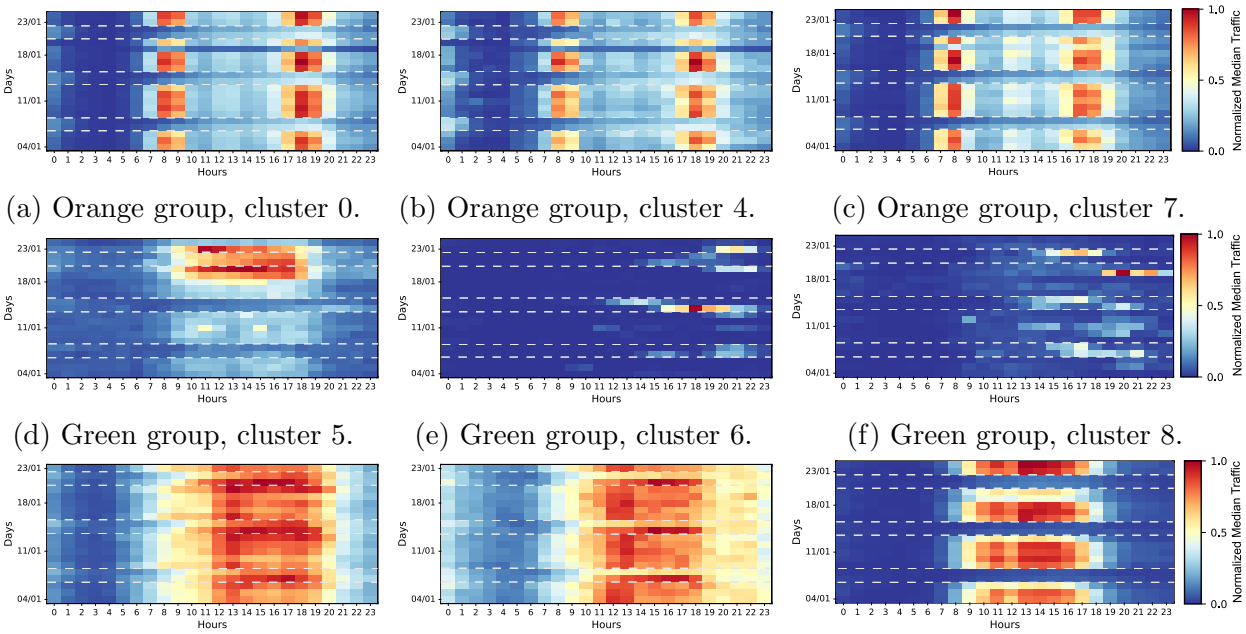
The next step in the analysis of ICN traffic is to uncover the patterns it exhibits over time. This analysis was conducted in collaboration with André Felipe Zanella from IMDEA Networks. Expectedly, one should observe different patterns for each cluster, as has been highlighted in previous research, though this work focused solely on the traffic recorded at outdoor antennas [38, 39]. To this end, Figure 6.10 provides heatmaps showing the evolution of the normalised median traffic per hour across all the antennas belonging to the same cluster, for the period of 04/01/2023 to 24/01/2023. Evidently, it can be pointed out that there is a strong correlation between the temporal patterns and the indoor environment type. Indeed, as can be seen in Figure 6.10a to 6.10c, the orange group clusters, which are populated with metro and train stations, demonstrate a traffic peak during the common weekly commuting hours in France, *i.e.*, 7.30 to 9.30 a.m. and 17.30 to 19.30 p.m. In the remaining hours of the day, the traffic volume is considerably smaller, while the same holds throughout the weekends, *e.g.*,, the 7th and 8th, or the 14th and 15th of January. Remarkably, there is another day with negligible traffic in the period under consideration; the 19th of January, which corresponds to a national general strike day. The strike's impact is not as severe for cluster 7, which includes metro antennas found in cities other than Paris, presumably due to the milder impact of the strike in these cities.

The green group clusters temporal patterns, presented in Figs. 6.10d to 6.10f, are characterised by sporadic, non-canonical bursts of data usage. Again, this is an expected behaviour for stadiums and convention centres, where unlike the other indoor environments discussed, surges of mobile subscribers appear on the premises and generate traffic only when events are taking place. For instance, in cluster 8 a traffic outbreak was observed only on the evening of January 19th in a cross-Atlantic NBA special event conducted at Accor Arena in

Paris, while the continuous burst in cluster 5 between the 19th and the 24th of January is a corollary of the 4-day Sirha Lyon event that took place at the Eurexpo Lyon convention centre. Another difference between the green group clusters is that cluster 5 records a low volume of traffic throughout the entire day. Indeed, cluster 5 does not include only stadiums but other environments as well, which as discussed in Section 6.3.1.1 all share in common a moderate application usage, as indicated by the mild blue colour.

As depicted in Figs 6.10g to 6.10i, the red group clusters present a more vivid and diurnal pattern compared to the other two groups, with the traffic being almost evenly distributed from 10 a.m. to 8 p.m. Interestingly, clusters 1 and 2, which comprise a large number of commercial antennas, record traffic throughout the weekdays and weekends, while cluster 3 consisting primarily of workspaces remains idle during weekends and after working hours, *i.e.*, 5.30 pm. That is a unique behaviour that clearly differentiates cluster 3 from the other two clusters. Delving further into the peculiarities of each cluster, cluster 2 yields a slight drop on Sundays, which can be attributed to the fact that it contains smaller stores, *e.g.*, the MNO agencies, which are not as active on those dates compared to larger commercial stores and malls. A further difference between clusters 1 and 2, is that the latter presents higher traffic during night time, likely due to the larger number of hotels and hospitals, as shown in Figure 6.8b, which are more active during these hours.

Finally, further insight can be gained by inspecting the temporal heatmaps for some mobile

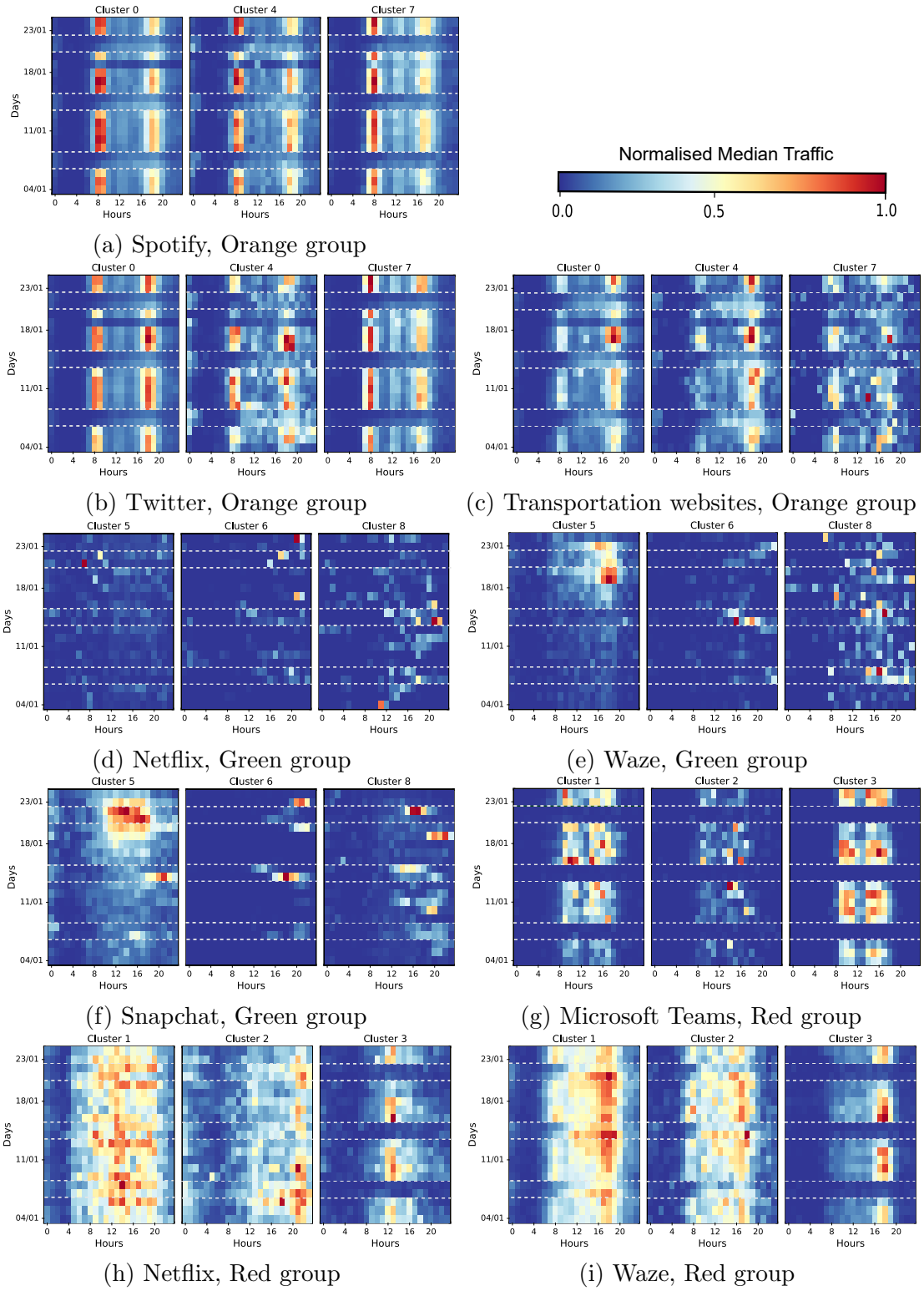


(a) Orange group, cluster 0. (b) Orange group, cluster 4. (c) Orange group, cluster 7.

(d) Green group, cluster 5. (e) Green group, cluster 6. (f) Green group, cluster 8.

(g) Red group, cluster 1. (h) Red group, cluster 2. (i) Red group, cluster 3.

Figure 6.10: Normalised median traffic heatmaps per hour for a period between 04/01/2023 and 24/01/2023. Each heatmap represents the median traffic of all antennas that belong to the specified cluster at a specific hour and day, while the light grey dashed lines indicate weekends.



(a) Spotify, Orange group

(b) Twitter, Orange group

(c) Transportation websites, Orange group

(d) Netflix, Green group

(e) Waze, Green group

(f) Snapchat, Green group

(g) Microsoft Teams, Red group

(h) Netflix, Red group

(i) Waze, Red group

Figure 6.11: Heatmaps of per hour normalised median traffic between 04/01/2023 and 24/01/2023, for the antennas of each cluster and for a selected set of services chosen based on their importance according to the SHAP values presented in Figure 6.5. The light grey dashed lines indicate weekends.

services that assumed a key role in the cluster decision-making process, as per Figure 6.5. Hence, for the orange group clusters Spotify, Twitter, and transport websites are selected, depicting their normalised median traffic as heatmaps in Figures 6.11a-6.11c. As it can be seen, Spotify assumes a strong presence for the entire group, with similar motifs for all clusters, demonstrating its traffic peaks during the morning commuting hours. On the other hand, the usage of transport websites is scattered for cluster 7, while clusters 0 and 4 preserve a more lively commuting hour pattern. Moreover, as indicated by Figure 6.5b, Twitter usage is comparatively diffuse for cluster 4, since both clusters 0 and 7 present a persistent peak during morning or evening commuting hours.

The traffic patterns of the green group clusters, comprising event-oriented venues, are also in alignment with the conclusions drawn based on the SHAP values. For instance, Snapchat in Figure 6.11f presents traffic patterns significantly similar to the ones seen for total traffic in Figure 6.10, indicating the substantial usage of social media apps throughout these events. Likewise, in Figure 6.11e, Waze which is used for driving navigation and obtaining live traffic maps and road alerts, also shows an interesting traffic pattern. In particular, for these antennas, Waze assumes its peak values a couple of hours after the peaks of total traffic, as well as those of social media apps, which suggest the usage of vehicular navigation apps to guide the event attendants back to their home destination. In addition, video streaming apps, such as Netflix, fall into under-utilisation in such venues, even on specific peak days and hours, as can be seen in Figure 6.11d.

For the red group, three very characteristic applications are selected aspiring to highlight their different utilisation during day hours; Microsoft Teams, Netfilx, and Waze. Microsoft Teams, which is a work-oriented service, attains very small peak values for clusters 1 and 2 which are composed mostly of non-working environments. On the contrary, cluster 3 records heavy traffic over working hours, or even during the lunch break, since it is populated with workspaces. Meanwhile, Netflix exhibits the opposite pattern, yielding a stronger usage in daytime and nighttime for clusters 1 and 2, respectively, but only being utilized during lunch hours for class 3. Again, this behaviour aligns with intuition since cluster 1 is of general use, cluster 2 includes most of the hotels where guests use streaming services during nighttime, while in working environments the use of streaming services is strictly limited during breaks. Finally, corroborating the findings of Figure 6.5g, out of the red group clusters, Waze attains the highest importance and recorded traffic for cluster 1, which aligns with intuition since as per Figure 6.8a it includes the majority of tunnels. Furthermore, the Waze traffic peaks in cluster 1 occur mostly on Saturdays, whereas for cluster 3 the heaviest traffic is recorded on weekdays after office hours when the employees return to their residences.

6.6 Summary

The proliferation of ICNs and the increasing mobile service demands generated by users in indoor environments call for an improved understanding of indoor traffic characteristics. This chapter constitutes a primer in this direction, unveiling the unique behaviours that are intrinsic to the traffic generated by ICNs. It is shown that a set of clusters, with distinct mobile service utilisation profiles, exists across a nationwide ICN deployment. Through explainable ML, a detailed analysis of the services that can identify these profiles is provided, and consequently, it is demonstrated that ICN traffic demands strongly relate to the type of indoor environment. These profiles are more diverse compared to those encountered in the traffic generated by conventional outdoor BSs, and they exhibit distinctive overall and per-application utilisation temporal patterns, influenced by the kind of activities that take place in the different indoor environments. Overall, this work paves the road to the characterisation of service-level mobile traffic demands in indoor environments, and offers a number of insights into patterns that were not observed or demonstrated quantitatively before.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis studied the applications of AI in electromagnetics and indoor wireless networks. Towards the first direction, the work presented advances the state-of-the-art by introducing a robust GNN-based CEM solver, called GEM, and a full-fledged data-driven propagation model. Specifically, GEM solves Maxwell's equations numerically through a non-trainable GNN message passing configuration, yielding the same accuracy compared to conventional transient CEM methods combined with an immense computational time gain. The validity of the introduced proof-of-concept framework was verified via diverse CEM simulations and two individual applications from the fields of biomedical engineering and photonics. This work sets the foundations for a new way of approaching computational sciences, where a differential equation is discretised through a numerical analysis method, the underlying discretised computational domain is perceived as a graph, and the message exchange between the graph nodes leads to the numerical solution of the initial differential equation.

In the second direction of the work, a data-driven model, called EM DeepRay, was trained with data from a ray-tracer and can accurately infer radio signal attenuation in any indoor environment. The fidelity of EM DeepRay was assessed in scenarios with multiple unknown parameters, *e.g.*, different geometries, frequency bands, and antenna radiation patterns, observing that the inferred PL distributions were consistent with the results of the ray tracer. This work paves the way for replacing legacy indoor wireless communication system simulators, with reliable AI-based radio propagation models. Consequently, the computational efficiency of the data-driven propagation model was leveraged for AI-assisted indoor wireless network planning, coupling EM DeepRay with AI optimisers to determine automatically the network topology that optimises some network KPIs. It was observed that the optimisation runs with EM DeepRay provided network deployments that attained the same performance as network topologies identified by equivalent experiments utilising a ray tracing simulator, yet the computational time required for the latter was higher by

two orders of magnitude. The drastic decrease in the optimisation time renders frameworks leveraging data-driven propagation models a key enabler for AI-driven planning solutions and the timely installation of indoor cellular and industrial private networks, which are expected to be extensively deployed in 5G/B5G systems.

Finally, AI was employed to probe the traffic dynamics of a nationwide ICN deployment. This analysis unveiled that the overall and per-application ICN mobile service demands are site-specific, highly dependent on the type of indoor environment, and more specialised than outdoor ones; therefore, ICN resource orchestration should not target overall capacity, as in outdoor environments, but must take into account the most important application usage per indoor environment. In a sense, this work fosters adopting a distinct network slicing dimension for indoor network resource planning, where the indoor slices will be tuned based on the characterising applications for that specific indoor environment.

7.2 Future Work

GNN-based PDE resolution. The approach for solving Maxwell's equations with non-trainable GNNs is directly transferable to other systems of differential equations arising in other sciences. This includes the Navier-Stokes, Burger's, Heat, Laplace, and Schrodinger's equations, or other partial differential equations used for example in neuroscience or structural engineering. Furthermore, the proposed approach is not applicable only to the finite difference method but can be extended to other numerical analysis techniques, such as the finite element method. Hence, for any such partial differential equation and numerical analysis method, it is possible to construct a distinct GNN for which the weighted message exchange depicts the resolution of the target mathematical expression.

Furthermore, one could also include non-linearities and learnable components in the graph-driven solvers, to account either for real-world measured data (in cases where the fidelity of partial differential equation is not guaranteed) or to operate outside the stability limits of conventional computational science methods. In this case, the node messages will be transformed into non-linear functions with learnable weights that depend on the graph node values, the pre-defined edge connections and weights (that allow the GNN to attain an equivalent solution in the first place), the values of the neighboring feature nodes, as well as the properties of the PDE and the simulation domain. Infusing learnable components may enable the equivalent GNN representation to overcome the sampling and stability constraints of the corresponding numerical method, inferring the PDE solution for an arbitrary discretization of the PDE domain, or in case of time-dependent PDEs to emulate the entire trajectory of the PDE solution at non-uniformly spaced points in time.

Data-Driven Radio Propagation Models. Future research on data-driven propagation models can further empower their role in wireless network applications. Specifically, most

of the existing generalisable data-driven propagation models are trained to predict physical quantities related to the signal level. On the contrary, data-driven models aiming at inferring other wireless channel QoIs, e.g., the phase, angle of arrival/departure or the delay spread, currently exhibit limited scalability and they cannot generalise to unseen geometries or frequency bands. These characteristics are important for many applications of wireless communication systems, such as dynamic beamforming and beam steering, that can enable real-time network optimisation to accommodate user requirements. However, recent work has shown [162] that such QoIs exhibit rapid fluctuations in space and can phenomenally be perceived as noise (see Fig. 3b in [162]). Hence, in contrast to the smoother PL spatial distributions, such QoIs manifest noisy patterns that are unlikely to be learned by encoding-decoding architectures like those presented in Chapter 4. A potential solution to that would be to forge data-driven models that learn tracing rays per se, rather than simply inferring a channel characteristic, *i.e.*, train models that learn to predict the entire ray trajectory instead of just a single QoI. Then, the attributes (material parameters, distances, phase difference, ray AoA) of the points along the ray trajectory could be non-linearly transformed to infer concurrently various correlated measures, such as multipath delay, and AoA/AoD.

Another noteworthy direction would be to explore the use of large language models (LLMs) in radio propagation modeling. These models have demonstrated a unique potential to process large datasets of images, identify patterns and correlations within them, and consequently create new realistic images with remarkable faithfulness. Hence, one could perceive radio map generation as an image generation problem, and expose an LLM to images depicting radio propagation environments and the respective radio map images. Multi-modal information related to the nature of radio propagation could also be contextualized and provided to the LLM in the form of prompts, *e.g.*, "When electromagnetic waves impinge onto walls you should observe a reflected and a transmitted component attenuated by:" and then provide as additional information the tokenised Fresnel transmission and reflection coefficients as a function of the angle of incidence.

Network Planning, UQ, and ICN Traffic Analysis. Future research on network planning with data-driven models can explore multi-objective optimisation problems where a set of conflicting target criteria ought to be simultaneously met, e.g., having KPIs related to latency or meeting energy and budget constraints associated with the fBS transmit power and their number. Moreover, the use of advanced optimisation algorithms, such as the simultaneous perturbation stochastic approximation or zeroth-order optimisation, is worth investigating and their performance compared to that of meta-heuristics. Additionally, one could consider the distinct temporal behaviours of the ICN traffic to configure network operations optimally. In a sense, the conclusions of Chapter 6 foster adopting a distinct network slicing dimension for indoor network resource planning, where the indoor slices will be tuned based on the characterising applications for that specific indoor environment.

Applications of such slicing could include adaptive power transmission control or content caching according to the insights and patterns provided by the analysis of Chapter 6, *e.g.*, the transmitting power could be reduced in metro antennas during off-peak hours.

Finally, another interesting direction would be to bring together all these three components (planning, UQ, traffic analysis) into a single framework implementation. Specifically, the interplay between UQ and network planning is inextricably linked to the reliability-based design optimisation problems [163]. In such cases, the goal is to optimise a cost function depicting the system's performance, while taking into account various constraints that may be subject to uncertainty occurring due to randomness in designing or environmental parameters. Consequently, one could attempt to design a network and be confident that a target metric, *e.g.*, coverage, will remain higher than a certain threshold despite the impact of any uncertainties. The uncertainty would not necessarily stem from the propagation environment, but could also be related to the user traffic demands. Hence the analysis of Chapter 6 can be leveraged to extract lower and upper limits of the mobile traffic demands for different indoor environments that can be consequently used as constraints in the optimisation problem, *e.g.*, the throughput should remain higher a percentile of the traffic demand lower limit coming from Chapter 6 analysis.

Bibliography

[1] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Computer Networks*, vol. 144, pp. 17–39, 2018.

[2] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[3] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78729–78757, 2019.

[4] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, 2019.

[5] B. Ma, W. Guo, and J. Zhang, "A survey of online data-driven proactive 5G network optimisation using machine learning," *IEEE Access*, vol. 8, pp. 35606–35637, 2020.

[6] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 2019.

[7] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, 2020.

[8] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 2019.

[9] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 2019.

[10] A. Seretis and C. D. Sarris, "An overview of machine learning techniques for radiowave propagation modeling," *IEEE Trans. Antennas and Propag.*, vol. 70, pp. 3970–3985, 6 2022.

[11] C. Huang, R. He, B. Ai, A. F. Molisch, B. K. Lau, K. Haneda, B. Liu, C.-X. Wang, M. Yang, C. Oestges, *et al.*, "Artificial intelligence enabled radio propagation for communications—part

ii: Scenario identification and channel modeling," *IEEE Trans. Antennas and Propag.*, vol. 70, pp. 3955–3969, 6 2022.

[12] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong, and T. Kürner, "The design and applications of high-performance ray-tracing simulation platform for 5G and beyond wireless communications: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 10–27, 1st Quart. 2018.

[13] S. F. Yunas, M. Valkama, and J. Niemelä, "Spectral and energy efficiency of ultra-dense networks under different deployment strategies," *IEEE Com. Mag.*, vol. 53, pp. 90–100, 01 2015.

[14] S. F. Yunas, A. Asp, J. Niemela, and M. Valkama, "Deployment strategies and performance analysis of macrocell and femtocell networks in suburban environment with modern buildings," in *Proc. IEEE 39th Conf. Local Comput. Netw.*, pp. 643–651, 09 2014.

[15] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, pp. 72–79, 2 2016.

[16] "Cisco vision: 5G-thriving indoors," white paper, Cisco, San Jose, CA, USA, 2019.

[17] Ericsson, "White paper: Bringing 5G networks indoors," tech. rep., Ericsson, 2017.

[18] Huawei, "White paper: Indoor 5G networks," tech. rep., Huawei, 2018.

[19] ZTE, "White paper: 5G indoor," tech. rep., ZTE, 2019.

[20] A. Aijaz, "Private 5G: The future of industrial wireless," *IEEE Ind. Electron. Mag.*, vol. 14, no. 4, pp. 136–145, 2020.

[21] X. Zhang, X. Shu, B. Zhang, J. Ren, L. Zhou, and X. Chen, "Cellular network radio propagation modeling with deep convolutional neural networks," *in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining,*, pp. 2378–2386, 8 2020.

[22] O. Perrault, J.-P. Rossi, and T. Balandier, "Predicting field strength with a neural ray-tracing model," in *Proc. IEEE Global Commun. Conf.*, vol. 2, pp. 1167–1171, IEEE, 1996.

[23] S. P. Sotiroudis, S. K. Goudos, and K. Siakavara, "Deep learning for radio propagation: Using image-driven regression to estimate path loss in urban areas," *ICT Express*, vol. 6, no. 3, pp. 160–165, 2020.

[24] S. P. Sotiroudis, P. Sarigiannidis, S. K. Goudos, and K. Siakavara, "Fusing diverse input modalities for path loss prediction: A deep learning approach," *IEEE Access*, vol. 9, pp. 30441–30451, 2021.

[25] V. V. Ratnam, H. Chen, S. Pawar, B. Zhang, C. J. Zhang, Y.-J. Kim, S. Lee, M. Cho, and S.-R. Yoon, "FadeNet: Deep learning-based mm-wave large-scale channel fading prediction and its applications," *IEEE Access*, vol. 9, pp. 3278–3290, 2020.

[26] L. Wu, D. He, B. Ai, J. Wang, H. Qi, K. Guan, and Z. Zhong, "Artificial neural network based path loss prediction for wireless communication network," *IEEE access*, vol. 8, pp. 199523–199538, 2020.

[27] J. Thrane, D. Zibar, and H. L. Christiansen, "Model-aided deep learning method for path loss prediction in mobile communication systems at 2.6 Ghz," *IEEE Access*, vol. 8, pp. 7925–7936, 2020.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[29] S. Wang, S.-Y. Hu, E. Cheah, X. Wang, J. Wang, L. Chen, M. Baikpour, A. Ozturk, Q. Li, S.-H. Chou, *et al.*, "U-Net using stacked dilated convolutions for medical image segmentation," *arXiv preprint arXiv:2004.03466*, 2020.

[30] L. Azpilicueta, M. Rawat, K. Rawat, F. M. Ghannouchi, and F. Falcone, "A ray launching-neural network approach for radio wave propagation analysis in complex indoor environments," *IEEE Trans. Antennas Propag.*, vol. 62, no. 5, pp. 2777–2786, 2014.

[31] L. Bai, C.-X. Wang, J. Huang, Q. Xu, Y. Yang, G. Goussetis, J. Sun, and W. Zhang, "Predicting wireless mmWave massive MIMO channel characteristics using machine learning algorithms," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[32] S. Bakirtzis, K. Qiu, J. Zhang, and I. Wassell, "DeepRay: Deep learning meets ray-tracing," in *Proc. 16th Eur. Conf. Antennas Propag. (EuCAP)*, pp. 1–5, 03 2022.

[33] S. Bakirtzis, J. Chen, K. Qiu, J. Zhang, and I. Wassell, "EM DeepRay: An expedient, generalizable, and realistic data-driven indoor propagation model," *IEEE Trans. on Antennas and Propag.*, vol. 70, no. 6, pp. 4140–4154, 2022.

[34] S. Bakirtzis, I. Wassell, M. Fiore, and J. Zhang, "Stochastic evaluation of indoor wireless network performance with data-driven propagation models," in *Proc. IEEE Global Commun. Conf.*, pp. 3587–3592, 12 2022.

[35] A. C. Austin, N. Sood, J. Siu, and C. D. Sarris, "Application of polynomial chaos to quantify uncertainty in deterministic channel models," *IEEE Trans. Antennas Propag.*, vol. 61, no. 11, pp. 5754–5761, 2013.

[36] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2190–2202, 2018.

[37] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, 2019.

[38] A. Furno, M. Fiore, R. Stanica, C. Ziemlicki, and Z. Smoreda, "A tale of ten cities: Characterizing signatures of mobile traffic in urban areas," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2682–2696, 2016.

[39] A. Furno, M. Fiore, and R. Stanica, "Joint spatial and temporal classification of mobile traffic demands," in *IEEE INFOCOM 2017*, (New York City at 3 Park Ave), pp. 1–9, IEEE, 2017.

[40] Y. Jin, N. Duffield, A. Gerber, P. Haffner, W.-L. Hsu, G. Jacobson, S. Sen, S. Venkataraman, and Z.-L. Zhang, "Characterizing data usage patterns in a large cellular network," in *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, (New York, NY, USA), pp. 7–12, Association for Computing Machinery, 2012.

[41] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, (57 Morehouse LaneRed HookNYUnited States), pp. 4768–4777, Curran Associates Inc., 2017.

[42] S. Bakirtzis, A. F. Zanella, S. Rubrichi, C. Ziemlicki, Z. Smoreda, I. Wassell, J. Zhang, and M. Fiore, "Characterizing mobile service demands at indoor cellular networks," in *Proceedings of the 2023 ACM on Internet Measurement Conference*, pp. 645–659, 2023.

[43] S. Bakirtzis, M. Fiore, I. Wassell, and J. Zhang, "Expedient AI-assisted indoor wireless network planning with data-driven propagation models." TechRxiv, 2023.

[44] S. Bakirtzis, K. Qiu, I. Wassell, M. Fiore, and J. Zhang, "Deep-learning-based multivariate time-series classification for indoor/outdoor detection," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24529–24540, 2022.

[45] S. Bakirtzis, M. Fiore, and I. Wassell, "Towards graph neural network-based computational electromagnetics," in *2024 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, IEEE, 2024.

[46] S. Bakirtzis, M. Fiore, J. Zhang, and I. Wassell, "Solving maxwell's equations with non-trainable graph neural network message passing," *arXiv preprint arXiv:2405.00814*, 2024.

[47] S. Bakirtzis, C. Yapar, M. Fiore, J. Zhang, and I. Wassell, "Empowering wireless network applications with deep learning-based radio propagation models," *arXiv preprint arXiv:2408.12193*, 2024.

[48] E. Dahlman, G. Mildh, S. Parkvall, J. Peisa, J. Sachs, and Y. Selén, "5g radio access," *Ericsson review*, vol. 6, no. 1, 2014.

[49] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Com. Mag.*, vol. 57, no. 8, pp. 84–90, 2019.

[50] T. K. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, "A survey of various propagation models for mobile communication," *IEEE Antennas Propag. Mag.*, vol. 45, pp. 51–82, 6 2003.

[51] R. Janaswamy, *Radiowave propagation and smart antennas for wireless communications.* Springer Science & Business Media, 2001.

[52] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, "Overview of millimeter wave communications for fifth-generation (5G) wireless networks with a focus on propagation models," *IEEE Trans. Antennas Propag.*, vol. 65, no. 12, pp. 6213–6230, 2017.

[53] V. Abhayawardhana, I. Wassell, D. Crosby, M. Sellars, and M. Brown, "Comparison of empirical propagation path loss models for fixed wireless access systems," in *2005 IEEE 61st Vehicular Technology Conference*, vol. 1, pp. 73–77, IEEE, 2005.

[54] E. Damosso, L. M. Correia, *et al.*, "COST action 231: Digital mobile radio towards future generation systems: Final report," *European commission*, 1999.

[55] K. Hari, D. Baum, A. Rustako, R. Roman, and D. Trinkwon, "Channel models for fixed wireless applications," *IEEE 802.16 Broadband wireless access working group*, 2003.

[56] F. Fuschini, E. M. Vitucci, M. Barbiroli, G. Falciasecca, and V. Degli-Esposti, "Ray tracing propagation modeling for future small-cell and indoor applications: A review of current techniques," *Radio Sci.*, vol. 50, pp. 469–485, 6 2015.

[57] M. F. Cátedra and J. Perez, *Cell planning for wireless communications.* Artech House, Inc., 1999.

[58] H. Ling, R.-C. Chou, and S.-W. Lee, "Shooting and bouncing rays: Calculating the RCS of an arbitrarily shaped cavity," *IEEE Trans. Antennas Propag.*, vol. 37, no. 2, pp. 194–205, 1989.

[59] A. Taflove, S. C. Hagness, and M. Piket-May, "Computational electromagnetics: the finite-difference time-domain method," *The Electrical Engineering Handbook*, vol. 3, no. 629-670, p. 15, 2005.

[60] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, vol. 114, no. 2, pp. 185–200, 1994.

[61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[62] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[63] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[64] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Int. Conf. Mach. Learn.*, pp. 1263–1272, PMLR, 2017.

[65] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89, IEEE, 2018.

[66] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2020.

[67] L. S. Shapley, H. Kuhn, and A. Tucker, "Contributions to the theory of games," *Annals of Mathematics studies*, vol. 28, no. 2, pp. 307–317, 1953.

[68] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowl. Inf. Sys.*, vol. 41, pp. 647–665, 2014.

[69] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56–67, 2020.

[70] A. Bondeson, T. Rylander, and P. Ingelström, *Computational electromagnetics.* New York City, New York, USA: Springer, 2012.

[71] D. E. McCoy, A. V. Shneidman, A. L. Davis, and J. Aizenberg, "Finite-difference time-domain (FDTD) optical simulations: a primer for the life sciences and bio-inspired engineering," *Micron*, vol. 151, p. 103160, 2021.

[72] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade, *Photonic crystals: Molding the flow of light.* Princeton, New Jersey, USA: Princeton University Press, 2008.

[73] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis.* North Chelmsford, Massachusetts, USA: Courier Corporation, 2012.

[74] W. C. Gibson, *The method of moments in electromagnetics.* Boca Raton, Florida, USA: CRC Press, 2021.

[75] F. Teixeira, C. Sarris, Y. Zhang, D.-Y. Na, J.-P. Berenger, Y. Su, M. Okoniewski, W. Chew, V. Backman, and J. Simpson, "Finite-difference time-domain methods," *Nat. Rev. Methods Primers*, vol. 3, no. 1, p. 75, 2023.

[76] S. Gedney, *Introduction to the finite-difference time-domain (FDTD) method for electromagnetics.* New York City, New York, USA: Springer Nature, 2022.

[77] C. Guiffaut and K. Mahdjoubi, "A parallel FDTD algorithm using the MPI library," *Antennas Propag. Mag.*, vol. 43, no. 2, pp. 94–103, 2001.

[78] H. Bao and R. Chen, "An efficient domain decomposition parallel scheme for leapfrog ADI-FDTD method," *IEEE Trans. Antennas Propag.*, vol. 65, no. 3, pp. 1490–1494, 2017.

[79] S. E. Krakiwsky, L. E. Turner, and M. M. Okoniewski, "Acceleration of finite-difference time-domain (FDTD) using graphics processor units (GPU)," *in IEEE MTT-S Int. Microwave Symp. Digest*, vol. 2, pp. 1033–1036, 6 2004.

[80] C. Warren, A. Giannopoulos, A. Gray, I. Giannakis, A. Patterson, L. Wetter, and A. Hamrah, "A CUDA-based GPU engine for gprMax: Open source FDTD electromagnetic simulation software," *Comput. Phys. Commun.*, vol. 237, pp. 208–218, 2019.

[81] Z. Bo, X. Zheng-hui, R. Wu, L. Wei-ming, and S. Xin-qing, "Accelerating FDTD algorithm using GPU computing," *in Proc. IEEE Int. Conf. Microw. Technol. Comput. Electromagn.*, pp. 410–413, 5 2011.

[82] L. Guo, M. Li, S. Xu, F. Yang, and L. Liu, "Electromagnetic modeling using an FDTD-equivalent recurrent convolution neural network: Accurate computing on a deep learning framework," *IEEE Antennas Propag. Mag.*, 2021.

[83] H. M. Yao, L. Jiang, H. H. Zhang, and E. Wei, "Machine learning methodology review for computational electromagnetics," *in Proc. Int. Appl. Comput. Electromagn. Soc. Symp. China (ACES)*, vol. 1, pp. 1–4, 8 2019.

[84] S. D. Campbell and D. H. Werner, *Advances in Electromagnetics Empowered by Artificial Intelligence and Deep Learning.* Hoboken, New Jersey , USA: John Wiley & Sons, 2023.

[85] A. Seretis and C. D. Sarris, "Towards physics-based generalizable convolutional neural network models for indoor propagation," *IEEE Trans. Antennas Propag.*, 2022.

[86] R. Levie, Ç. Yapar, G. Kutyniok, and G. Caire, "Radiounet: Fast radio map estimation with convolutional neural networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 4001–4015, 2021.

[87] H. M. Yao, L. J. Jiang, and Y. W. Qin, "Machine learning based method of moments (ML-MoM)," *in Proc. IEEE-APS*, pp. 973–974, 7 2017.

[88] S. Barmada, N. Fontana, L. Sani, D. Thomopulos, and M. Tucci, "Deep learning and reduced models for fast optimization in electromagnetics," *IEEE Trans. Magn*, vol. 56, no. 3, pp. 1–4, 2020.

[89] R. Phellan, B. Hachem, J. Clin, J.-M. Mac-Thiong, and L. Duong, "Real-time biomechanics using the finite element method and machine learning: Review and perspective," *Med. Phys.*, vol. 48, no. 1, pp. 7–18, 2021.

[90] T. Shan, J. Zeng, X. Song, R. Guo, M. Li, F. Yang, and S. Xu, "Physics-informed supervised residual learning for electromagnetic modeling," *IEEE Trans. Antennas Propag.*, vol. 71, no. 4, pp. 3393–3407, 2023.

[91] H. M. Yao and L. J. Jiang, "Machine learning based neural network solving methods for the FDTD method," *in Proc. IEEE-APS*, pp. 2321–2322, July 2018.

[92] D. Zhu, Q. Zhao, Y. Bo, W. Chen, and L. Yang, "Application of deep learning in FDTD method," *in Proc. 13th Int. Symp. Antennas, Propag. EM Theor*, pp. 1–3, 2021.

[93] L. Guo, M. Li, S. Xu, and F. Yang, "Study on a recurrent convolutional neural network based fdtd method," *in Proc. Int. Appl. Comput. Electromagn. Soc. Symp.-China (ACES)*, vol. 1, pp. 1–2, 8 2019.

[94] L. Kuhn, T. Repän, and C. Rockstuhl, "Exploiting graph neural networks to perform finite-difference time-domain based optical simulations," *APL Photonics*, vol. 8, no. 3, 2023.

[95] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," *in Proc. ICLR Workshop Representation Learn. Graphs Manifolds*, 5 2019.

[96] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, 2021.

[97] Z. Hao, Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song, and J. Zhu, "Gnot: A general neural operator transformer for operator learning," in *Int. Conf. Mach. Learn*, pp. 12556–12569, PMLR, 2023.

[98] Z. Li, K. Meidani, and A. B. Farimani, "Transformer for partial differential equations' operator learning," *arXiv preprint arXiv:2205.13671*, 2022.

[99] B. Raonic, R. Molinaro, T. De Ryck, T. Rohner, F. Bartolucci, R. Alaifari, S. Mishra, and E. de Bézenac, "Convolutional neural operators for robust and accurate learning of pdes," *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

[100] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.

[101] D. J. Ozer, "Correlation and the coefficient of determination.," *Psychol. Bull.*, vol. 97, no. 2, p. 307, 1985.

[102] H. Alisoy, S. B. Us, and B. Alagoz, "An FDTD-based numerical analysis of microwave propagation properties in a skin–fat tissue layers," *Optik*, vol. 124, no. 21, pp. 5218–5224, 2013.

[103] S. Gabriel, R. Lau, and C. Gabriel, "The dielectric properties of biological tissues: Parametric models for the dielectric spectrum of tissues," *Phys. Med. Biol.*, vol. 41, no. 11, p. 2271, 1996.

[104] A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. D. Joannopoulos, and S. G. Johnson, "MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method," *Comput. Phys. Commun.*, vol. 181, no. 3, pp. 687–702, 2010.

[105] K. Qiu, S. Bakirtzis, H. Song, J. Zhang, and I. Wassell, "Pseudo ray-tracing: Deep leaning assisted outdoor mm-wave path loss prediction," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1699–1702, 2022.

[106] ITU-Rec, "Recommendation P.2040-1, Effects of building materials and structures on radiowave propagation above about 100 MHz," *Int. Telecommun. Union, Geneva, Switzerland*, 2015.

[107] (2021) Ranplan Wireless, Ranplan Professional.

[108] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing.* Cambridge university press, 2007.

[109] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3431–3440, 2015.

[110] Z. Zeng, X. Li, Y. K. Yu, and C.-W. Fu, "Deep floor plan recognition using a multi-task network with room-boundary-guided attention," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9096–9104, 2019.

[111] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Proc. Conf. Neural Informat. Process. Syst.*, vol. 34, pp. 12077–12090, 2021.

[112] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[113] J. C. Baez and J. P. Muniain, *Gauge fields, knots and gravity*, vol. 4. World Scientific Publishing Company, 1994.

[114] S. Hosseinzadeh, H. Larijani, K. Curtis, A. Wixted, and A. Amini, "Empirical propagation performance evaluation of LoRa for indoor environment," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pp. 26–31, IEEE, 2017.

[115] J. Zhang, A. A. Glazunov, W. Yang, and J. Zhang, "Fundamental wireless performance of a building," *IEEE Wirel. Commun.*, 2021.

[116] J. Zhang, A. A. Glazunov, and J. Zhang, "Wireless performance evaluation of building layouts: Closed-form computation of figures of merit," *IEEE Trans. Commun.*, vol. 69, pp. 4890–4906, 07 2021.

[117] 3GPP Technical Specification 38.306 Release 15, "NR: User equipment (UE) radio access capabilities," tech. rep., ETSI, 2017.

[118] A. W. Reza, M. S. Sarker, and K. Dimyati, "A novel integrated mathematical approach of ray-tracing and genetic algorithm for optimizing indoor wireless coverage," *Prog. Electromagn. Res.*, vol. 110, pp. 147–162, 2010.

[119] A. Oustry, M. Le Tilly, T. Clausen, C. d'Ambrosio, and L. Liberti, "Optimal deployment of indoor wireless local area networks," *Networks*, vol. 81, no. 1, pp. 23–50, 2023.

[120] M. J. Nassiri and S. Etemadi Borujeni, "A novel approach on femtocell placement in the commercial buildings using genetic algorithm," *Trans. Emerg. Telecommun. Technol*, vol. 32, no. 9, p. e4285, 2021.

[121] A. P. Engelbrecht, *Computational intelligence: an introduction.* Hoboken, NJ, USA: John Wiley & Sons, 2007.

[122] P. I. Frazier, "A tutorial on Bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[123] V. Feoktistov and S. Janaqi, "Generalization of the strategies in differential evolution," in *Proc. 18th IPDPS*, p. 165, IEEE, 2004.

[124] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," *Computational intelligence for multimedia big data on the cloud with engineering applications*, pp. 185–231, 2018.

[125] S. Stavrou and S. Saunders, "Review of constitutive parameters of building materials," in *Twelfth Int. Conf. Antennas Propag. , 2003 (ICAP 2003).(Conf. Publ. No. 491)*, vol. 1, pp. 211–215, IET, 2003.

[126] R. Rudd, K. Craig, M. Ganley, and R. Hartless, "Building materials and propagation," *Final Report, Ofcom*, vol. 2604, 2014.

[127] O. Le Maitre and O. M. Knio, *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics.* Springer Science & Business Media, 2010.

[128] D. Xiu, *Numerical methods for stochastic computations: a spectral method approach.* Princeton university press, 2010.

[129] S. Marelli and B. Sudret, "Uqlab: A framework for uncertainty quantification in matlab," in *Vulnerability, uncertainty, and risk: quantification, mitigation, and management*, 2014.

[130] D. Xiu and G. E. Karniadakis, "Modeling uncertainty in flow simulations via generalized polynomial chaos," *J. Comput. Phys.*, vol. 187, no. 1, pp. 137–167, 2003.

[131] T. Crestaux, O. Le Maıtre, and J.-M. Martinez, "Polynomial chaos expansion for sensitivity analysis," *Reliab. Eng. Syst. Saf.*, vol. 94, no. 7, pp. 1161–1172, 2009.

[132] S. Bakirtzis, X. Zhang, and C. D. Sarris, "Stochastic modeling of wave propagation in waveguides with rough surface walls," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 1, pp. 500–508, 2020.

[133] S. Marelli and B. Sudret, "Uqlab user manual–polynomial chaos expansions," *Chair of risk, safety & uncertainty quantification, ETH Zürich, 0.9-104 edition*, pp. 97–110, 2015.

[134] Access Evolved Universal Terrestrial Radio, "User equipment (UE) radio access capabilities," *Standard ETSI TS*, vol. 136, no. 306, p. V14, 2015.

[135] S. Sesia, I. Toufik, and M. Baker, *LTE-the UMTS long term evolution: from theory to practice.* John Wiley & Sons, 2011.

[136] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 905–929, 2020.

[137] M. Ghahramani, M. Zhou, and G. Wang, "Urban sensing based on mobile phone data: Approaches, applications, and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 627–637, 2020.

[138] A. Llorente, M. Garcia-Herranz, M. Cebrian, and E. Moro, "Social media fingerprints of unemployment," *PloS one*, vol. 10, no. 5, p. e0128692, 2015.

[139] N. Pokhriyal and D. C. Jacques, "Combining disparate data sources for improved poverty prediction and mapping," *Proc. Natl. Acad. Sci.*, vol. 114, no. 46, pp. E9783–E9792, 2017.

[140] J. E. Steele, P. R. Sundsøy, C. Pezzulo, V. A. Alegana, T. J. Bird, J. Blumenstock, J. Bjelland, K. Engø-Monsen, Y.-A. de Montjoye, A. M. Iqbal, K. N. Hadiuzzaman, X. Lu, E. Wetter, A. J. Tatem, and L. Bengtsson, "Mapping poverty using mobile phone and satellite data," *J. R. Soc. Interface*, vol. 14, no. 127, p. 20160690, 2017.

[141] J. Blumenstock, G. Cadamuro, and R. On, "Predicting poverty and wealth from mobile phone metadata," *Sci.*, vol. 350, no. 6264, pp. 1073–1076, 2015.

[142] S. Mishra, Z. Smoreda, and M. Fiore, "Second-level digital divide: A longitudinal study of mobile traffic consumption imbalance in France," in *Proceedings of the ACM Web Conference 2022*, WWW '22, (New York, NY, USA), p. 2532–2540, Association for Computing Machinery, 2022.

[143] I. Ucar, M. Gramaglia, M. Fiore, Z. Smoreda, and E. Moro, "News or social media? socio-economic divide of mobile service consumption," *J. R. Soc. Interface*, vol. 18, no. 185, p. 20210350, 2021.

[144] S. Dujardin, D. Jacques, J. Steele, and C. Linard, "Mobile phone data for urban climate change adaptation: Reviewing applications, opportunities and key challenges," *Sustainability*, vol. 12, no. 4, p. 1501, 2020.

[145] W. Chen, Y. He, and S. Pan, "Impact of air pollution on human activities: Evidence from nine million mobile phone users," *PLoS ONE*, vol. 16, p. e0251288, 2021.

[146] K. H. Grantz, H. R. Meredith, D. A. Cummings, C. J. E. Metcalf, B. T. Grenfell, J. R. Giles, S. Mehta, S. Solomon, A. Labrique, N. Kishore, *et al.*, "The use of mobile phone data to inform analysis of covid-19 pandemic epidemiology," *Nat. Commun.*, vol. 11, no. 1, p. 4961, 2020.

[147] N. Oliver, B. Lepri, H. Sterly, R. Lambiotte, S. Deletaille, M. D. Nadai, E. Letouzé, A. A. Salah, R. Benjamins, C. Cattuto, V. Colizza, N. de Cordes, S. P. Fraiberger, T. Koebe, S. Lehmann, J. Murillo, A. Pentland, P. N. Pham, F. Pivetta, J. Saramäki, S. V. Scarpino, M. Tizzoni, S. Verhulst, and P. Vinck, "Mobile phone data for informing public health actions across the covid-19 pandemic life cycle," *Sci. Adv.*, vol. 6, no. 23, p. eabc0764, 2020.

[148] A. F. Zanella, O. E. Martinez-Durive, S. Mishra, Z. Smoreda, and M. Fiore, "Impact of later-stages covid-19 response measures on spatiotemporal mobile service usage," in *IEEE INFOCOM 2022*, (New York, NY, USA), pp. 970–979, IEEE, 2022.

[149] European Union, "EU general data protection regulation (GDPR): Regulation (EU) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (General Data Protection Regulation)," june 2016.

[150] B. Balassa, "Trade liberalisation and "revealed" comparative advantage 1," *The manchester school*, vol. 33, no. 2, pp. 99–123, 1965.

[151] K. Laursen and C. Engedal, "The role of the technology factor in economic growth: A theoretical and empirical inquiry into new approaches to economic growth," *Unpublished MA dissertation*, 1995.

[152] K. Laursen, "Revealed comparative advantage and the alternatives as measures of international specialization," *Eurasian business review*, vol. 5, pp. 99–115, 2015.

[153] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[154] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *J. Am. Stat. Assoc*, vol. 58, no. 301, pp. 236–244, 1963.

[155] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.

[156] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.

[157] M. Moshkovitz, S. Dasgupta, C. Rashtchian, and N. Frost, "Explainable k-means and k-medians clustering," in *Int. Conf. Mach. Learn.*, pp. 7055–7065, PMLR, 2020.

[158] R. Fraiman, B. Ghattas, and M. Svarc, "Interpretable clustering using unsupervised binary trees," *Adv. Data Anal. Classif*, vol. 7, pp. 125–145, 2013.

[159] D. Bertsimas, A. Orfanoudaki, and H. Wiberg, "Interpretable clustering: an optimization approach," *Machine Learning*, vol. 110, pp. 89–138, 2021.

[160] B. Ghattas, P. Michel, and L. Boyer, "Clustering nominal data using unsupervised binary decision trees: Comparisons with the state of the art methods," *Pattern Recognition*, vol. 67, pp. 177–185, 2017.

[161] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using shapley additive explanations," *Expert systems with applications*, vol. 186, p. 115736, 2021.

[162] A. Seretis, C. Xu, and C. Sarris, "Fast selection of indoor wireless transmitter locations with generalizable neural network propagation models," *Authorea Preprints*, 2023.

[163] M. Moustapha, S. Marelli, and B. Sudret, "Uqlab user manual–reliability-based design optimization," tech. rep., Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH . . . , 2019.