ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

INGEGNERIA ELETTRONICA, TELECOMUNICAZIONI E TECNOLOGIE DELL'INFORMAZIONE

Ciclo 38

**Settore Concorsuale:** 09/F1 - CAMPI ELETTROMAGNETICI

**Settore Scientifico Disciplinare:** ING-INF/02 - CAMPI ELETTROMAGNETICI

# MACHINE LEARNING MODELS FOR WIRELESS CHANNEL MODELLING

**Presentata da:** Mohammad Hossein Zadeh

**Coordinatore Dottorato**

Davide Dardari

**Supervisore**

Franco Fuschini

**Co-supervisore**

Marina Barbiroli

Esame finale anno 2026

# Contents

# Introduction

The continuous evolution of wireless communication systems has led to an ever-growing demand for reliable, scalable, and efficient wireless connectivity across diverse environments. From personal and industrial communications to intelligent transportation systems and non-terrestrial networks, wireless technologies now form the backbone of modern society. The design, optimization, and deployment of such systems fundamentally requires an accurate characterization of the radio propagation channel, which describes how electromagnetic waves spread by the transmitting device can reach the receiver(s) interacting with the surrounding environment.

Radio propagation modeling plays a crucial role in wireless system design, as it provides the analytical and computational framework for predicting received signal properties, propagation losses, and other key channel characteristics. Over the years, several modeling approaches have been developed to describe the propagation channel. Empirical models, such as the Okumura–Hata and COST-231 formulations, offer simple and computationally efficient solutions based on large-scale measurement campaigns. Although these models have been widely adopted in system-level studies and standardization, they aim at the description of average trends, e.g the average range dependence of the received signal strength. Deterministic approaches, including Ray Tracing (RT) and Ray Launching (RL), explicitly account for electromagnetic interactions such as reflection, diffraction, and scattering, enabling site-specific predictions of signal propagation and channel behavior. However, these methods are computationally intensive and require detailed geometric and material information, which limits their applicability in large or dynamic scenarios.

The emergence of new communication paradigms such as 5G, 6G, and the Internet of Things (IoT), together with the increasing complexity of propagation environments, calls for more flexible and fast but still accurate tools. To this aim, the exploration of data-driven methodologies capable of learning the underlying relationships between environmental features and propagation outcomes directly from data has been gaining increasing attention over the last years.

Machine Learning (ML) offers a powerful framework to overcome these challenges. By leveraging large datasets and advanced learning algorithms, ML models can capture complex relationships between environmental, geometrical, and system parameters and the resulting propagation characteristics. Once trained, these models can provide ac-

curate and fast predictions of radio channel behavior and propagation characteristics with significantly reduced computational cost compared to RT. Moreover, ML-based models can generalize across different environments, offering an efficient and adaptive approach to wireless channel characterization.

The main objective of this thesis is to develop and validate ML models that can help tackling electromagnetic problems, ranging from wireless channel modeling and Line of Sight (LoS) assessment to Object Recognition (OR). The ML models described in this work are aimed at (i) prediction of key propagation parameters, like *received power* and *Path Loss (PL)*, which describe the signal attenuation over distance; the *shadowing* and *fast fading statistics*, which capture large- and small-scale variations of the signal due to obstacles and multipath propagation; and the *Delay Spread (DS)*, which quantifies the temporal dispersion of the transmitted signal caused by interactions within the environment or (ii) classification tasks like LoS detection and OR through backscattering analysis. The study investigates both tabular-data models, such as eXtreme Gradient Boosting (XGBoost) and Multilayer Perceptron (MLP), and image-based models, such as the Convolutional Neural Network (CNN)s, specifically 1D CNNs and UNet architecture, for predicting site-specific coverage maps. The thesis also compares ML-based predictions with classical empirical formulations to assess their relative accuracy, interpretability, and computational efficiency.

The adopted methodology combines physical modeling with data-driven learning. Deterministic RT simulations are employed to generate ground-truth datasets for multiple environments, including industrial and urban scenarios. The resulting data are preprocessed, structured, and augmented to form suitable training and testing sets for the selected ML models. This framework enables a systematic investigation of the predictive capability and scalability of ML models in wireless propagation modeling.

The remainder of this thesis is organized as follows. Chapter 1 introduces the fundamentals of wireless propagation and channel modeling, emphasizing the physical mechanisms affecting radio wave propagation. Chapter 2 presents the core concepts of machine learning and the models adopted in this work. Chapter 3 discusses the application of ML techniques to wireless propagation problems, including dataset generation. Chapter 4 reports and analyzes the obtained results, comparing ML-based predictions with empirical and deterministic approaches. Finally, the conclusions summarize the main findings and outline future research directions.

# 1

# Propagation in Wireless Systems

Wireless communication systems rely on the transmission and reception of Electromagnetic (EM) waves between spatially separated devices. Unlike wired systems, where the transmission medium is controlled and well-defined, in wireless channels the medium is open, shared, and variable, leading to signal modifications that must be properly understood and modeled [1].

The wireless propagation channel can be generally introduced as everything in between a Transmitting Antenna (Tx) and a Receiving Antenna (Rx). Inside the propagation channel, wireless signals undergo attenuation, distortion, and time, space, and frequency variations due to the surrounding environment. These effects determine the quality, capacity, and reliability of the communication system [2].

## 1.1 Free Space

The simplest case of wireless propagation is the free space radio link, where the medium is assumed to be homogeneous, isotropic, and lossless [1–4]. In such conditions, the Tx and Rx are in unobstructed LoS, as illustrated in Fig. 1.1.


**Figure 1.1:** LoS free space propagation between Tx and Rx along a direct path.

In this case, the received power is described by the Friis formula [5], which establishes

an inverse-square law between received power and distance as shown in Eq. 1.1.

$$P_r = P_t \cdot g_{tx}(\theta_{tx}, \phi_{tx}) \cdot g_{rx}(\theta_{rx}, \phi_{rx}) \cdot (\frac{\lambda}{4\pi d})^2 \cdot \rho_d \cdot \tau_P \tag{1.1}$$

where $d$ is the link distance, $P_t$ and $P_r$ denote the transmitted and received power, $g_{tx}$ and $g_{rx}$ are the antenna gains in the propagation directions, and $\lambda$ is the wavelength. The terms $\rho_d$ and $\tau_P$ account for power and polarization mismatches, respectively.

From a general perspective, the propagation channel can be characterized through several key parameters:

- **Path Gain (PG):** describes the average fraction of transmitted power that is received over a wireless link, accounting for the propagation effects between transmitter and receiver. In the free space case, assuming isotropic antennas and perfect impedance and polarization matching, the PG is given by the Friis transmission formula:
$$PG_{FS} = \left(\frac{\lambda}{4\pi d}\right)^2, \tag{1.2}$$
where $\lambda$ is the wavelength and $d$ is the link distance.
- **Path Loss (PL):** is defined as the multiplicative inverse of the path gain and provides an equivalent description of the same propagation phenomenon in terms of signal attenuation. It is commonly adopted for practical and logarithmic representations. In free space, the path loss is therefore expressed as:
$$PL_{FS} = \frac{1}{PG_{FS}} = \left(\frac{4\pi d}{\lambda}\right)^2. \tag{1.3}$$

In free space, the Friis transmission equation provides an exact description of the distance-dependent attenuation, showing that the received power decays proportionally to $d^{-2}$. This behavior corresponds to a path loss exponent equal to 2. In real propagation environments, however, the assumptions underlying the Friis model are no longer satisfied due to the presence of obstacles, reflections, diffraction, scattering, and absorption phenomena. These mechanisms alter the effective rate at which the average received power decreases with distance. To account for such effects, propagation models commonly introduce Path Loss Exponent (PLE), which generalizes the distance-dependent attenuation observed in free space and captures the impact of the surrounding environment on signal propagation.

- **PLE:** characterizes the rate at which the average received power (or equivalently the path gain) decays with distance in a given environment, as illustrated in Fig. 1.2. In free space, the PLE is equal to 2, corresponding to the inverse-square law of propagation. In real environments, its value depends on the surrounding geometry and obstruction level and typically ranges between 2 and 6.

The Friis equation is very simple and ready-to-use, and this made its success and popularity. However, it is often too simple, i.e. it can reliably describe EM propagation

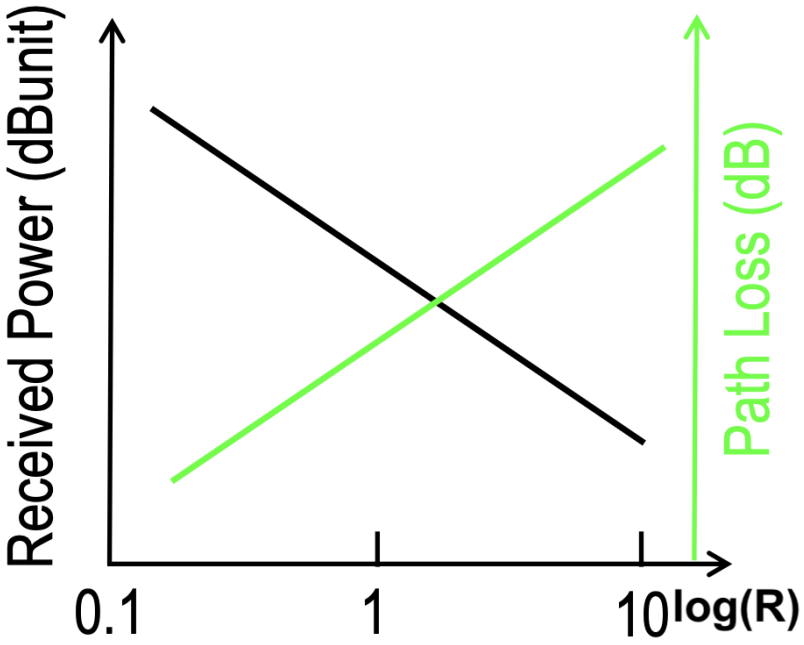



**Figure 1.2:** Received Power and PL vs. distance.

in real environment only to a rough extent. The main issue is that the assumptions the Friis equation relies on are usually not satisfied in practice:

- The medium is not homogeneous (the air itself is not perfectly homogeneous, and objects made of different materials are always present in practice).
- The medium is never completely loss-less.

## 1.2 Multipath

In practical scenarios, wireless communications almost always occur in the presence of multiple objects located around the transmitting and receiving antennas. When these objects are situated in the far-field regions of the antennas, which is often the case in real deployments, the electromagnetic wave emitted by the transmitting antenna is fundamentally spherical in nature, but can be locally approximated as a plane wave due to the negligible curvature of the wavefront at large distances. As this wavefront propagates, it inevitably interacts with surrounding objects, since most materials are not perfectly transparent to radio waves [1–4, 6].

The nature of these interactions depends on the physical characteristics of the illuminated objects, such as their shape, size, and material composition, as well as on the wavelength of the impinging wave. Different interaction mechanisms can therefore be distinguished:

- **Reflection** occurs when the wave encounters large, smooth, and homogeneous surfaces, such as walls or the ground.

- **Diffraction** is observed when the wave encounters sharp edges, tips, or apertures whose dimensions are small compared to the wavelength, allowing the wave to bend around obstacles and reach shadowed regions
- **Scattering** arises when the wave interacts with rough or inhomogeneous objects, producing multiple secondary waves that propagate in different directions.
- **Transmission/refraction** take place when the wave partially or fully penetrates a material with different electromagnetic properties, letting EM waves propagating through and beyond obstacles.
- **Absorption:** occurs when part of the electromagnetic energy is captured by the atomic or molecular structure of the material. This interaction may excite electrons to higher energy levels or trigger molecular vibrations and rotations, converting part of the wave energy into other forms such as heat or internal energy. As a result, the transmitted or reflected signal strength is reduced.

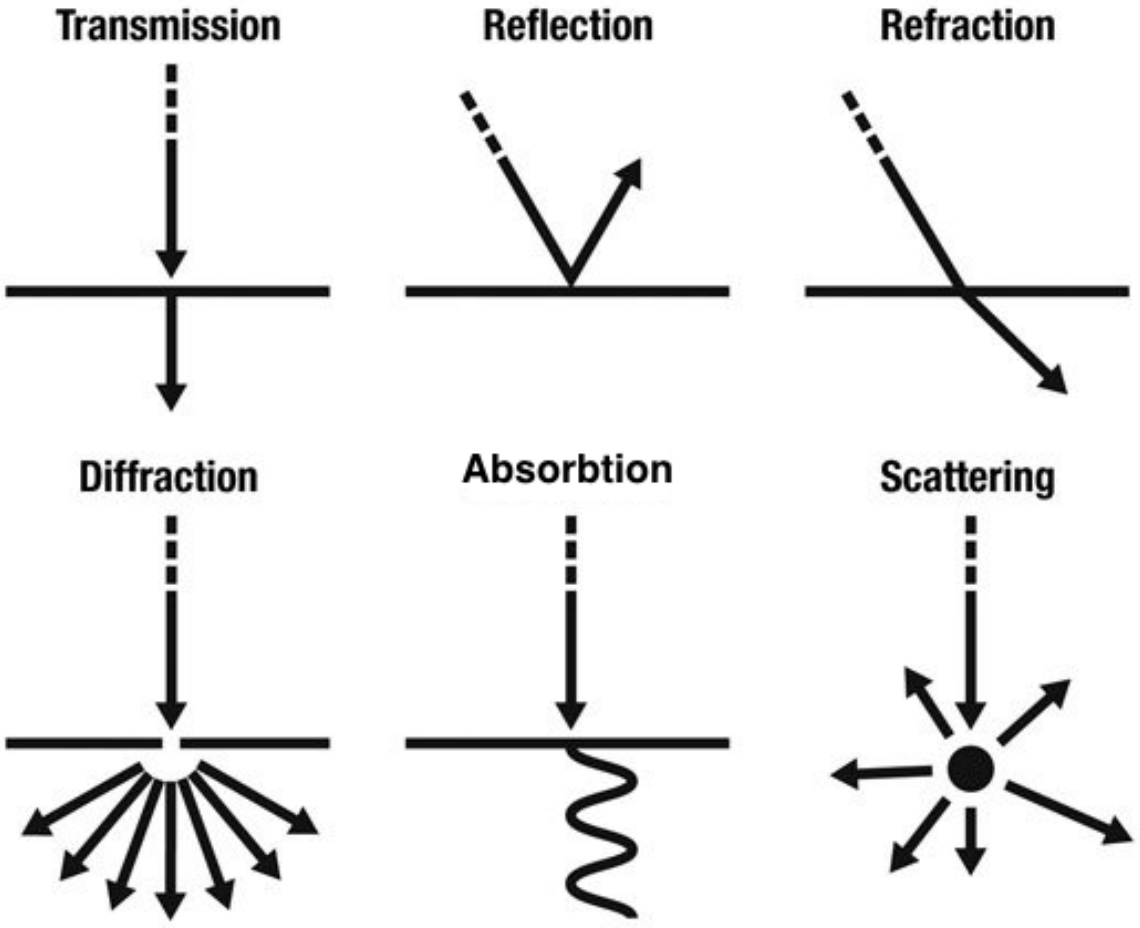



**Figure 1.3:** Wave Interactions.

In real environments, these interactions give rise to a phenomenon known as multipath propagation. The scattering of waves by surrounding objects generates a rich set of propagation paths between the transmitter and receiver. As a result, the received signal is not a single wavefront but rather the superposition of multiple replicas, each

arriving with different amplitudes, phases, and delays. Depending on their relative characteristics, these signal components may interfere constructively or destructively, leading to distortion, additional losses, and fluctuations in the received power. Multipath propagation is therefore a central feature of wireless channels and represents the starting point for understanding large-scale and small-scale fading phenomena, which will be discussed in the following sections [7, 8].

## 1.3 Fading Effects

The complex nature of multipath propagation can be conveniently described in terms of three mutually independent, multiplicative, and approximately separable components as shown in Fig. 1.4:

1. The overall, average range dependence of the received signal envelope, previously referred to as PG, or the dominant component (black dash dash dot line in Fig 1.4). The envelope denotes the instantaneous amplitude of the received signal after removal of the carrier, and it represents the slowly varying magnitude of the received waveform;
2. Fluctuations of the received signal levels occurring over distances much larger than the wavelength $\lambda$, referred to as large-scale fading or shadowing (red curve in Fig. 1.4);
3. Additional fluctuations superimposed on shadowing, observed over distances of the order of $\lambda/2$, called small-scale fading or fast fading (green curve in Fig. 1.4).

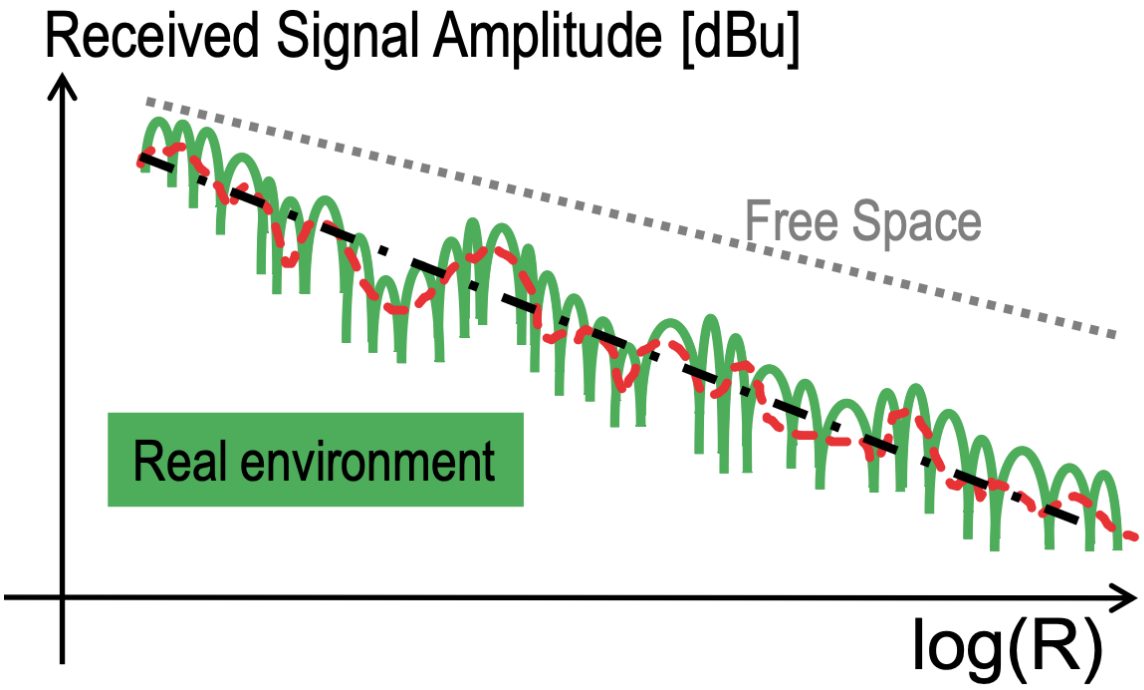



**Figure 1.4:** 3 main components.

This decomposition provides a practical framework to study radio propagation effects at different scales.

### 1.3.1 Dominant Component

The slope of the PG curve in real environments is generally steeper than in free space propagation (e.g. $PLE > 2$), due to the presence of obstacles that hinder the direct wave. The average received signal amplitude at distance $R$ reflects the mean obstruction experienced at that distance.

### 1.3.2 Shadowing

The actual obstruction at a given receiver location may differ from the average obstruction level experienced for the same distance. Such deviations are known as shadowing, as they capture the effect of specific obstacles between the transmitter and the receiver. Shadowing is considered a large-scale effect, since the receiver must typically be displaced by a distance comparable to the size of surrounding objects to experience uncorrelated shadowing. In urban environments, and especially at frequencies above UHF, this decorrelation distance is much larger than the wavelength.

### 1.3.3 Fast Fading

Superimposed on shadowing are rapid oscillations in the received power caused by the interference among multipath components. These oscillations are referred to as fast fading, since the relative phases of multipath contributions may vary significantly over distances comparable to the wavelength. Fast fading is thus a small-scale effect, directly related to the constructive and destructive interference patterns generated by multipath propagation.

Although different measurement routes in the same environment produce different detailed results, due to the point-specific nature of shadowing and multipath interference, they share the same average path gain trend with distance. By definition, this trend is obtained from measurements over many receiver locations within the environment. Consequently, the large- and small-scale fading fluctuations can be regarded as realizations of the same random processes, characterized by well-defined statistical distributions.

To distinguish between large-scale and small-scale fading, the concept of the small area average is often introduced [1]. It corresponds to the spatial average of the received signal magnitude over an area with linear dimensions of several tens of wavelengths.

$$V_{SA}(\vec{r}) = \frac{1}{A} \int_S |v(\vec{r} + \vec{s})| \, dS \tag{1.4}$$

where $V_{SA}(\vec{r})$ represents the *spatially averaged* value of the received signal amplitude around the point $\vec{r}$, $v(\vec{r} + \vec{s})$ denotes the instantaneous received signal amplitude at

a position displaced by the vector $\vec{s}$ from $\vec{r}$, $S$ is the spatial averaging region, and $A = \int_S dS$ denotes the total surface area of $S$.

This averaging filters out the rapid oscillations due to fast fading, isolating the contributions of path gain and shadowing. The fast fading component at a given location can then be defined as the ratio between the instantaneous received signal and the small-area average.

$$\rho = \frac{|v(\vec{r})|}{V_{SA}(\vec{r})} \tag{1.5}$$

This separation enables a statistical characterization of fast fading based on experimental data.

#### 1.3.3.1 Rayleigh Fading

The statistical behavior of fast fading depends on the relative contributions of the multipath components. In the absence of a dominant component, such as when no direct path exists between transmitter and receiver, the fast fading component is well described by a Rayleigh distribution. This distribution accurately models the case where many multipath components with comparable amplitudes arrive at the receiver. In this scenario, the received power follows an exponential distribution [9, 10].

The Probability Density Function (PDF) of the received signal envelope $\rho$ in Rayleigh fading is:

$$f_\rho(\rho) = \begin{cases} \frac{\rho}{\sigma^2}\, e^{-\frac{\rho^2}{2\sigma^2}}, & \rho \geq 0 \\ 0, & \rho < 0 \end{cases} \tag{1.6}$$

where $\sigma$ is the scale parameter, related to the root mean square (RMS) value of the scattered components.

#### 1.3.3.2 Rice Fading

When a dominant component is present, for example in line-of-sight conditions where the direct contribution is stronger than the scattered components, the received envelope follows a Rice distribution. The ratio between the power of the dominant component and the scattered multipath contributions is quantified by the Rice factor $K$. A high $K$-factor corresponds to strong LoS conditions, whereas a low $K$-factor indicates a channel dominated by multipath [11].

The PDF of the received envelope $\rho$ in Rice fading is:

$$f_\rho(\rho) = \begin{cases} \frac{\rho}{\sigma^2}\, e^{-\frac{\rho^2+A^2}{2\sigma^2}} I_0\left(\frac{\rho A}{\sigma^2}\right), & \rho \geq 0 \\ 0, & \rho < 0 \end{cases} \tag{1.7}$$

where $A$ is the amplitude of the dominant component, $\sigma$ is the scale parameter of the scattered components, and $I_0(\cdot)$ is the modified Bessel function of the first kind and zero order. The Rice factor is defined as

$$K = \frac{A^2}{2\sigma^2}, \tag{1.8}$$

expressing the power ratio between the deterministic (dominant) and the random (scattered) signal contributions.

## 1.4 Dispersive effects

In addition to attenuation, multipath propagation also produces dispersive effects, since the replicas of the transmitted signal travel along paths of different lengths and therefore arrive at the receiver with different delays. Moreover, in the presence of mobility within the propagation channel, these multipath components may also appear at the Rx with different frequencies due to the doppler effect [12]. This temporal and frequency spreading of the signal introduces distortion, which becomes particularly critical in wideband systems. Unlike PL and fading, which mainly affects the amplitude of the received signal, dispersive effects describe the temporal and spectral structure of the received waveform [4, 7, 13].

### 1.4.1 Delay Spread

A common metric to quantify temporal dispersion is the Root Mean Square (RMS) delay spread $\sigma_\xi$, which represents the standard deviation of the multipath Power Delay Profile (PDP). If $P(\xi)$ denotes the average received power at delay $\xi$, the RMS delay spread is defined as:

$$\sigma_\xi = \sqrt{\frac{\sum_i P(\xi_i)(\xi_i - \bar{\xi})^2}{\sum_i P(\xi_i)}} \tag{1.9}$$

where $\bar{\xi}$ is the mean excess delay, given by:

$$\bar{\xi} = \frac{\sum_i P(\xi_i)\xi_i}{\sum_i P(\xi_i)}. \tag{1.10}$$

The delay spread is inversely related to the coherence bandwidth $B_c$, which defines the frequency range over which the channel frequency response can be considered approximately flat [2]:

$$B_c \approx \frac{1}{\sigma_\xi}. \tag{1.11}$$

Small delay spreads imply that the channel behaves as frequency-flat, whereas large delay spreads result in frequency-selective fading, leading to possible distortion effects

such as Inter Symbol Interference (ISI) in digital systems. In industrial environments, the presence of large metallic machinery and complex geometries often results in significant delay spread values, as shown in recent measurement campaigns. Such conditions must be properly accounted for in system design, particularly for broadband wireless networks.

In practice, the PDP is often observed to exhibit an exponentially decaying behavior with delay, reflecting the reduced power of later multipath components. A common model expresses the PDP as:

$$Q(\xi) = Ae^{-\alpha\xi}, \quad \xi \geq 0, \tag{1.12}$$

where $A$ is a normalization constant and $\alpha$ is the decay factor, which depends on the propagation environment.

## 1.5 Channel Modelling

In order to design and evaluate wireless systems, accurate channel models are required. These models aim at describing the propagation mechanisms in a compact form suitable for link-level and system-level analysis. Channel models can be broadly categorized into two main classes: statistical models and deterministic models.

### 1.5.1 Statistical Models

Statistical models are based on empirical data collected from extensive measurement campaigns, or can be derived from general EM theory applied to simplified descriptions of the environment. They provide closed-form expressions for average propagation quantities, such as PL, shadowing, and fading statistics, typically as functions of frequency, distance, and environment type. The most widely known examples are the Okumura-Hata model [14] and its extensions, such as the COST-231 Hata model [15]for frequencies up to 2 GHz. Other well-known approaches include the Walfisch-Bertoni model [16] for urban microcells and the more recent 3GPP [17] and WINNER [18] models for cellular and vehicular scenarios.

These models are simple to use and computationally efficient, making them suitable for large-scale network planning. However, they often fail to capture environment-specific peculiarities, especially in complex indoor or industrial settings. Recent studies have highlighted their limitations in such environments, where PLE and shadowing deviations significantly depart from classical assumptions [1, 4, 19, 20].

### 1.5.2 Deterministic Models

Deterministic models explicitly account for the geometry and electromagnetic properties of the environment. Among them, Ray-based (Ray Tracing or Ray Launching)

is the most popular approach, as it simulates the propagation of electromagnetic rays by considering reflection, diffraction, and scattering based on geometrical optics (GO) and the uniform theory of diffraction (UTD). Ray tracing can provide highly accurate predictions of received power, delay spread, and angular characteristics, provided that detailed environmental maps are available [20, 21], as well as accurate description of the antennas radiation patterns.

Although computationally intensive, deterministic models have become increasingly feasible thanks to advances in computing power and efficient algorithms. In recent years, they have also served as a foundation for generating datasets used in machine learning-based channel modeling, where neural networks learn to approximate the mapping between environmental features and propagation outcomes.

# 2

# Machine Learning

Artificial Intelligence (AI) is a broad field of computer science concerned with designing systems that are capable of performing tasks that normally require human intelligence, such as reasoning, perception, decision-making, and learning [22, 23]. Within this field, Machine Learning (ML) has emerged as the most successful paradigm, focusing on algorithms that learn directly from data instead of being explicitly programmed with handcrafted rules [24, 25]. A further specialization is Deep Learning (DL), which relies on multi-layer neural networks capable of automatically extracting hierarchical feature representations [26, 27].

The relationship between AI, ML, and DL is illustrated in Figure 2.1. AI is the overarching field, ML is a subfield of AI, and DL is a subfield of ML.

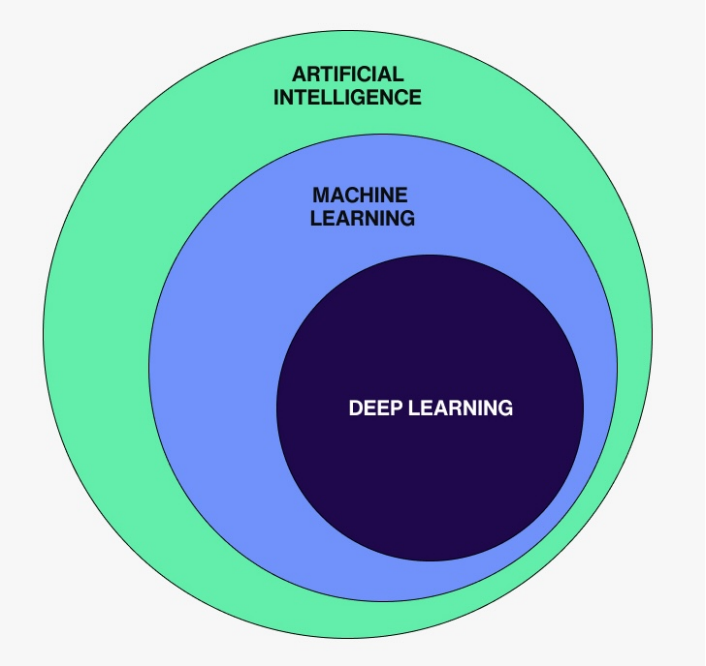



**Figure 2.1:** Relationship between AI, ML and DL

## 2.1 Machine Learning

ML is the study of algorithms and computational methods that enable computer systems to learn patterns from data and make predictions or decisions without being explicitly programmed. Instead of relying on handcrafted rules, ML algorithms improve their performance through experience.

A widely cited definition, due to Tom Mitchell, states [24]:

> "A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$."

Formally, given a dataset

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \tag{2.1}$$

where $x_i \in \mathbb{R}^d$ are input features and $y_i$ are target outputs (labels), a common goal of ML in predictive tasks is to find a function $f$ such that:

$$f(x_i) \approx y_i, \quad \forall (x_i, y_i) \in \mathcal{D}. \tag{2.2}$$

The prediction $\hat{y}_i = f(x_i)$ is evaluated through a loss function $\ell(y_i, \hat{y}_i)$, and learning corresponds to minimizing the empirical risk:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, \hat{y}_i). \tag{2.3}$$

Depending on the task, the algorithm is trained by minimizing a loss function appropriate for regression (e.g., squared error) or classification (e.g., cross-entropy) [28, 29].

This formulation mainly applies to supervised learning problems involving interpolation or pattern recognition. In decision-making contexts, such as reinforcement learning, the objective instead involves learning a policy that maximizes an expected reward over time rather than minimizing a prediction error.

### 2.1.1 Types of Learning

Machine learning tasks can be grouped into three major categories [24, 25, 30]:

- **Supervised learning:** the algorithm is trained on labeled data, where both inputs and desired outputs are known. Examples include regression (predicting continuous values) and classification (predicting categories).
- **Unsupervised learning:** the algorithm operates on unlabeled data to discover hidden structure, such as clusters or latent factors. Examples include clustering and dimensionality reduction.
- **Reinforcement learning:** an agent interacts with an environment, choosing actions to maximize a cumulative reward signal. Unlike supervised learning, feedback is not given as explicit labels but as rewards or penalties.

Figure 2.2 provides an overview of these learning types.

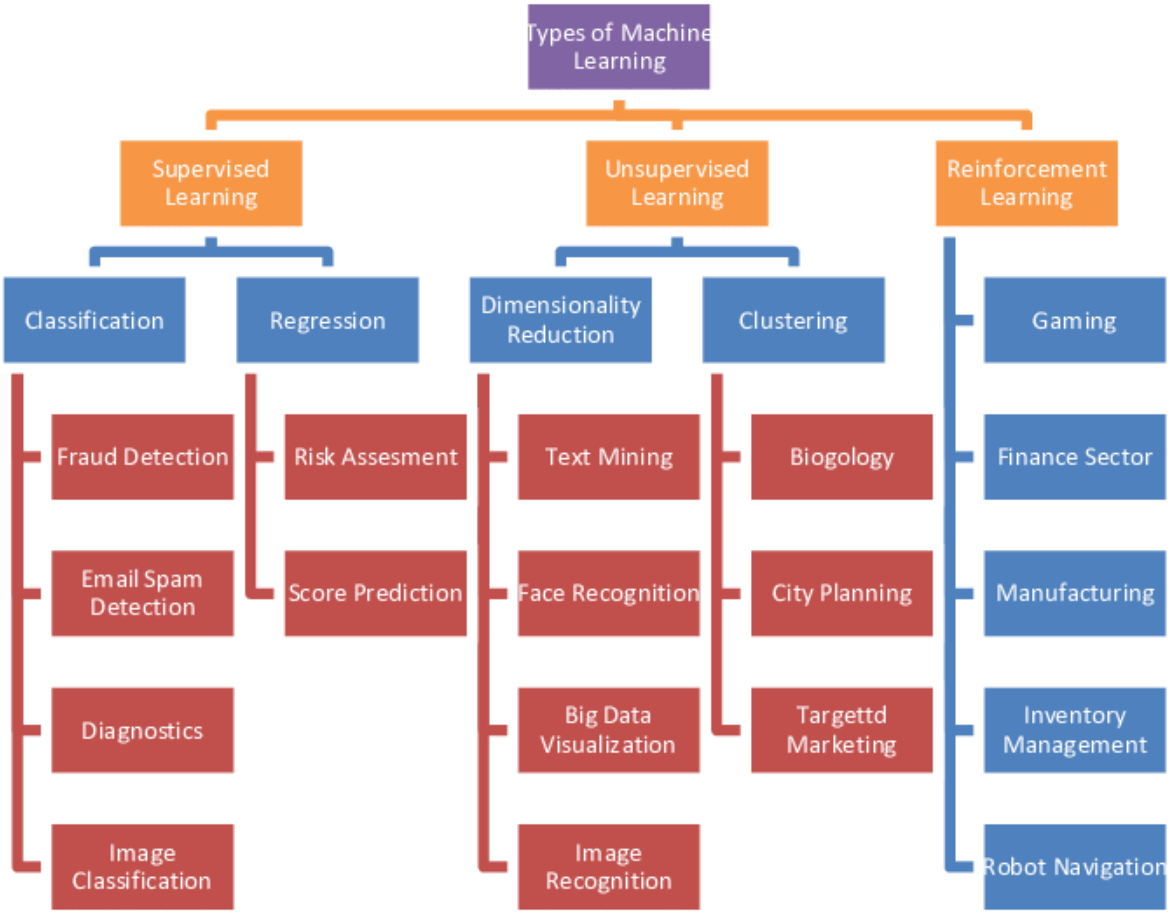



**Figure 2.2:** ML categories.

### 2.1.2 Core Concepts

Several concepts are central to the success of ML, including generalization, overfitting, underfitting, and the bias–variance tradeoff [25, 26, 31].

- **Generalization:** the ability of a model to perform well on unseen data, beyond the training examples.
- **Overfitting:** occurs when the model learns noise and details of the training set, losing generalization capability.
- **Underfitting:** happens when the model is too simple to capture the underlying patterns of the data.
- **Bias–Variance tradeoff:** the decomposition of expected error into bias, variance, and irreducible noise:

$$\mathbb{E}[(y - \hat{y})^2] = \text{Bias}^2 + \text{Variance} + \sigma^2, \tag{2.4}$$

where:

- *Bias* measures the systematic error introduced by approximating a complex real-world process with a simplified model;
- *Variance* quantifies the sensitivity of the model predictions to variations in the training data;
- $\sigma^2$ represents the *irreducible noise*, i.e., the intrinsic variability of the data that cannot be reduced by any model.

### 2.1.3 Training and Validation

To evaluate performance, data are usually split into three subsets as shown in Fig. 2.3:

- **Training set:** used to fit the model parameters.
- **Validation set:** used for model selection and hyperparameter tuning.
- **Test set:** used for final performance evaluation on unseen data.

Cross-validation techniques, such as $k$-fold cross-validation, are often used to provide more robust estimates of generalization [32].

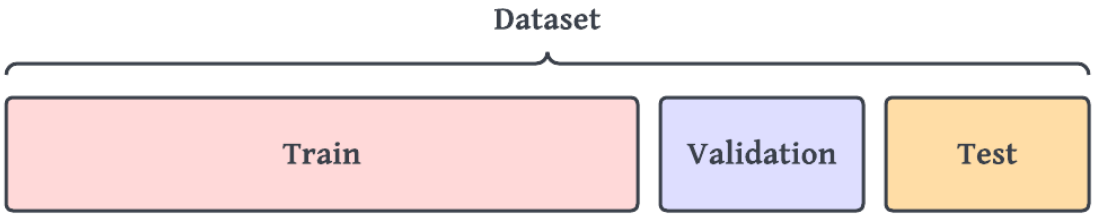



**Figure 2.3:** Dataset division

## 2.2 Supervised Learning

The objective is to learn a function that generalizes well, i.e., produces accurate predictions for new inputs, while avoiding an excessive reliance on memorization of the training set alone.

### 2.2.1 Regression vs. Classification

Supervised learning tasks are divided into two main categories:

- **Regression:** the output variable is continuous. The goal is to predict numerical values, such as temperature, stock prices, or received signal power.
- **Classification:** the output variable is categorical. The goal is to assign inputs to one of a finite number of classes, for example distinguishing spam from non-spam emails or identifying objects in an image.

### 2.2.2 Models for Supervised Learning

Different algorithms can be used to tackle supervised learning problems. While some are more naturally suited to regression or to classification, many can be adapted to both.

- **Linear Models:** Among the simplest approaches, these models assume a linear relationship between input features and the target [33].
  - *Regression:* Predictions are obtained as a weighted sum of the inputs $\bar{x} = (x_1, x_2, \ldots, x_j, \ldots, x_d)$:
    $$\hat{y} = \beta_0 + \sum_{j=1}^{d} \beta_j x_j, \tag{2.5}$$
    where $\beta_j$ are the model coefficients and $\beta_0$ is the intercept. The coefficients are typically estimated by minimizing the mean squared error between predictions and true values.
  - *Classification:* The same linear combination is transformed into class probabilities. For binary classification, logistic regression applies the sigmoid function:
    $$P(y = 1|\bar{x}) = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad z = \beta_0 + \sum_{j=1}^{d} \beta_j x_j. \tag{2.6}$$
    Where $\bar{x} = (x_1, x_2, \ldots, x_d)$ denotes the input feature vector. For multi-class classification, the softmax function generalizes this idea:
    $$P(y = k|\bar{x}) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}, \quad z_k = \beta_{0k} + \sum_{j=1}^{d} \beta_{jk} x_j, \tag{2.7}$$
    where $K$ is the number of classes. This formulation provides normalized probabilities that sum to 1, allowing the model to decide the most likely class while directly interpreting the output probability as a measure of classification confidence.
- **Support Vector Machines (SVMs)):** work by finding the boundary that best separates data points of different classes. This boundary is chosen to maximize the margin, i.e., the distance between the closest points from each class and the separating hyperplane, which helps improve robustness. By using kernels, SVMs can also handle cases where the classes are not linearly separable, effectively projecting the data into a higher-dimensional space where separation is easier. In regression (Support Vector Regressors (SVRs)), the method instead fits a function that predicts values within a tolerance band, focusing on capturing the general trend rather than exact matches [34].
- **Decision Trees:** partition the input space by asking a series of questions about the features, similar to a flowchart. Each internal node corresponds to a decision rule (e.g., "is temperature > 20°C?"), and each leaf node corresponds to a prediction. In regression, predictions at the leaves are the average of target values

in that region. In classification, predictions are based on the majority class in that region. Trees are easy to interpret, but if grown too deep, they can overfit the training data [35].

- **Ensemble Methods:** Ensembles improve predictive performance by combining the outputs of multiple base learners. In this work, the focus is on tree-based ensembles, which are among the most widely used and effective approaches.
  - *Random Forests* train multiple trees on randomly chosen subsets of the data and features. The final prediction is obtained by averaging the outputs (regression) or taking a majority vote (classification). This reduces variance and improves generalization [36].
  - *Gradient Boosting*, exemplified by algorithms such as eXtreme Gradient Boosting (XGBoost), builds trees sequentially. Each new tree focuses on correcting the errors of the previous ones, and the combination results in a highly accurate model.

## 2.3 Unsupervised Learning

Unlike supervised learning, where labels are available, unsupervised learning works with unlabeled data $\{x_i\}_{i=1}^{N}$ and aims to discover hidden patterns or structures. The objective is not to predict explicit outputs but to organize the data in a meaningful way.

### 2.3.1 Clustering

Clustering groups similar data points into clusters without requiring labeled data. The most widely used approach is $k$**-means**, which partitions the data into $k$ clusters by minimizing the within-cluster variance:

$$\min_{C} \sum_{i=1}^{N} \|x_i - \mu_{c_i}\|^2, \tag{2.8}$$

where $\mu_{c_i}$ is the centroid of cluster $c_i$ [37].

### 2.3.2 Dimensionality Reduction

Dimensionality reduction aims to represent high-dimensional data in a more compact form while preserving its most important structure. The most widely used technique is Principal Component Analysis, which projects the data onto orthogonal directions of maximum variance [38, 39]. More recent nonlinear approaches, such as t-SNE and UMAP, are often used for visualization in two or three dimensions, capturing complex manifold structures [40, 41].

Although unsupervised methods are not central to this thesis, they are widely used for data preprocessing, anomaly detection, and exploratory analysis.

## 2.4 Reinforcement Learning

Reinforcement learning studies how an agent interacts with an environment by taking actions and receiving feedback in the form of rewards. The objective is to learn a policy that maximizes long-term cumulative reward. Unlike supervised learning, RL does not rely on labeled examples but instead learns through trial and error. Classical methods include Q-learning, which estimates the value of actions in given states, and policy gradient approaches, which directly optimize the action-selection strategy [30].

## 2.5 Deep Learning

DL is a subfield of ML that uses multi-layer Artificial Neural Network (ANN) to learn hierarchical representations of data. Unlike classical models, which often rely on hand-crafted features, deep networks can automatically extract relevant features from raw inputs, making them particularly powerful in domains such as computer vision, speech recognition, and natural language processing [26, 27, 42].

### 2.5.1 ANN

ANNs are computational models inspired by the functioning of the human brain. They consist of interconnected processing units, called artificial neurons, which collectively learn to approximate complex functions [43–45].

An artificial neuron computes a weighted sum of its inputs, adds a bias, and passes the result through a nonlinear activation function:

$$y = f\left(\sum_{i=1}^{d} w_i x_i + b\right), \tag{2.9}$$

where $x_i$ are the input features, $w_i$ are trainable weights that control the contribution of each input, $b$ is a bias term, and $f(\cdot)$ is a nonlinear activation function, and $d$ denotes the number of input features.

The structure of a single neuron and its training process are illustrated in Fig. 2.4. Inputs $x_i$ are combined linearly with weights $w_i$ and bias $b$ through an *aggregation function*, producing $z = \sum_i w_i x_i + b$. This is then passed through an *activation function* $f(z)$ to generate the output $\hat{y}$. The error, computed as the difference between ground truth $y$ and output $\hat{y}$, is used to update the weights by means of the learning rule $\Delta w = \eta\,(error)\,x$, where $\eta$ is the learning rate. This iterative update mechanism corresponds to *gradient descent*, an optimization algorithm that minimizes a loss function by iteratively updating the model parameters in the direction of the negative gradient. This process underlies the training of neural networks.

Popular activation functions include(see Fig. 2.5):

- **Sigmoid:** $f(z) = \frac{1}{1+e^{-z}}$, squashes input into $(0, 1)$, suitable for probabilities [25].

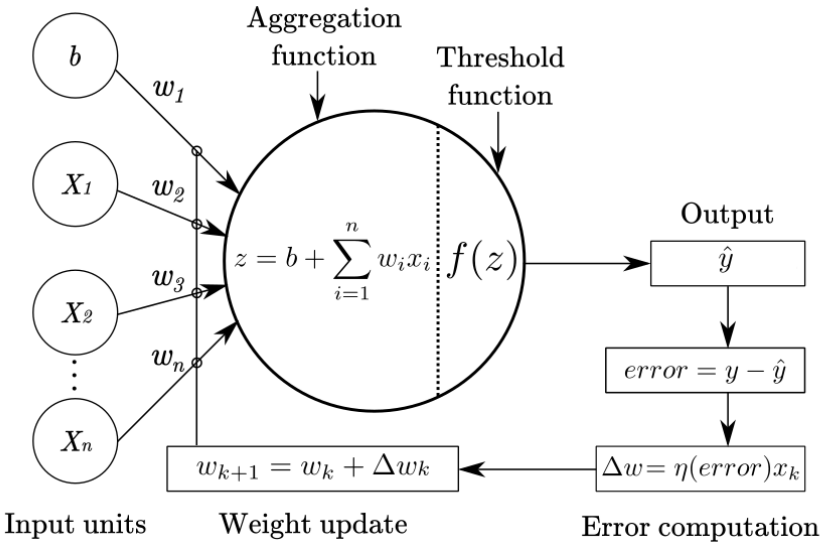



**Figure 2.4:** Schematic representation of an artificial neuron, illustrating input aggregation, activation, error computation, and weight update.

- **Hyperbolic Tangent (tanh):** $f(z) = \tanh(z)$, maps inputs to $(-1, 1)$, often leading to faster convergence than sigmoid [25].
- **Rectified Linear Unit (ReLU):** $f(z) = \max(0, z)$, widely used for deep networks due to efficiency and mitigation of vanishing gradients [46, 47].
- **Leaky ReLU:** $f(z) = \max(\alpha z, z)$, where $\alpha$ is a small constant (e.g., 0.01), allowing a small gradient for negative inputs and addressing the "dying ReLU" problem.

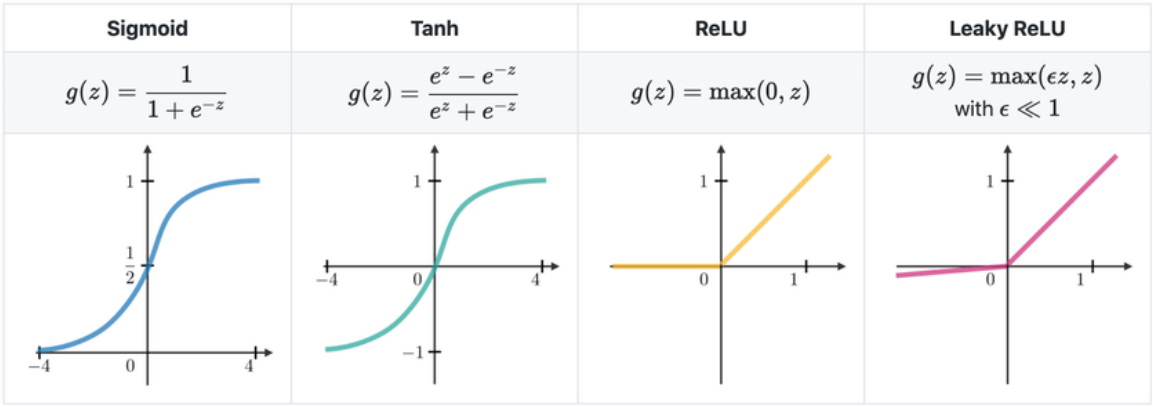



**Figure 2.5:** Activation Functions.

### Multilayer Perceptrons

When neurons are organized into layers, they form a neural network. A typical feedforward network, also called a Multilayer Perceptron (MLP), is structured as shown in 4.25 [44, 48]:

- An **input layer**, which receives raw data.
- One or more **hidden layers**, where nonlinear transformations are applied.
- An **output layer**, which produces predictions (e.g., regression values or classification scores).

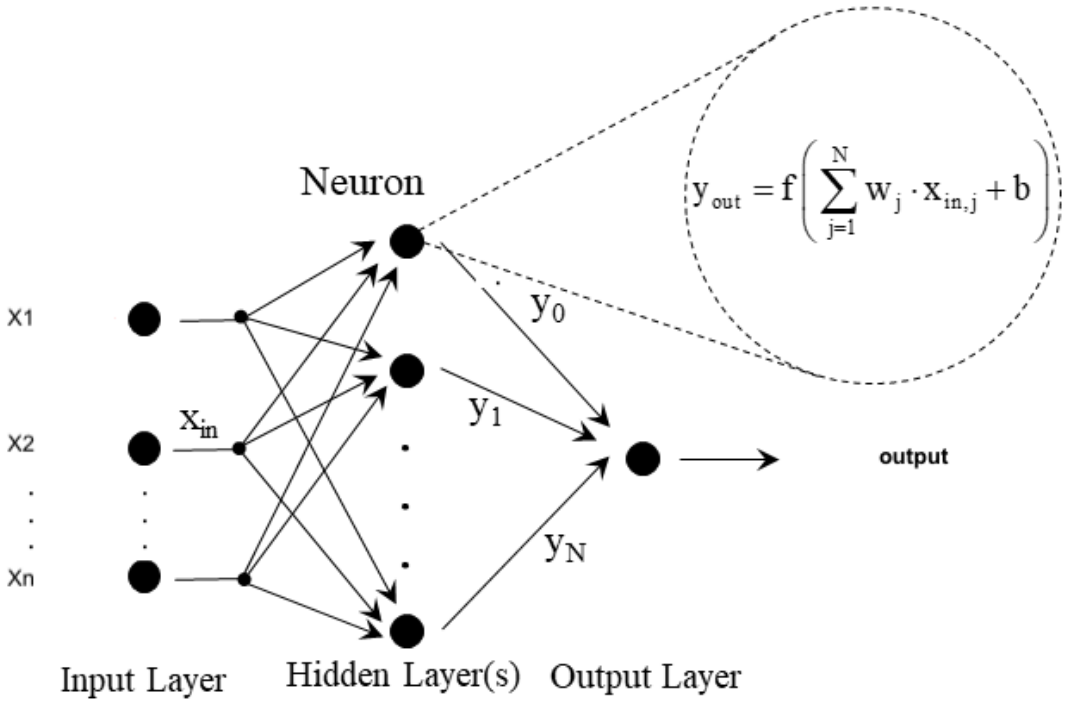



**Figure 2.6:** General structure of a Multi-Layer Perceptron. A single hidden layer is deployed for the sake of simplicity

Each layer $l$ transforms its input vector $\mathbf{x}^{(l)}$ into an output $\mathbf{x}^{(l+1)}$ by:

$$\mathbf{x}^{(l+1)} = f\left(W^{(l)}\mathbf{x}^{(l)} + \mathbf{b}^{(l)}\right), \tag{2.10}$$

where $W^{(l)}$ is the weight matrix, $\mathbf{b}^{(l)}$ the bias vector, and $f(\cdot)$ the activation function.

### Learning Process

Training a neural network involves adjusting weights and biases to minimize a **loss function** $\ell(y, \hat{y})$, which measures the difference between the predicted output $\hat{y}$ and the ground truth $y$. This optimization is performed by:

- **Backpropagation:** efficiently computes gradients of the loss with respect to all network parameters using the chain rule.
- **Optimization algorithms:** Stochastic Gradient Descent (SGD) and its variants such as Adam are commonly used to update parameters [49, 50].

#### Approximation Capability

A fundamental property of MLPs is their ability to approximate any continuous function given sufficient neurons and layers. This theoretical result, known as the *universal approximation theorem* [51], guarantees the existence of such representations. However, it does not address how these networks should be trained in practice, nor how to select suitable architectures to achieve effective learning. Despite these limitations, this property makes ANNs a powerful tool for regression, classification, and more complex learning tasks.

### 2.5.2 Convolutional Neural Networks

While MLPs are powerful universal approximators [51], they become inefficient for inputs such as signals and images. In an MLP, every neuron in a layer is connected to every neuron in the next layer, which leads to a very large number of parameters and poor scalability. Moreover, MLPs do not explicitly exploit the local correlation structures typically present in sequential or spatial data [52].

Convolutional Neural Network (CNN) addresses these limitations by introducing two key concepts:

- **Local receptive fields:** each neuron is connected only to a small, localized region of the input, enabling the network to detect local features such as edges in images or local patterns in signals.
- **Weight sharing:** the same filter (kernel) is applied across different regions of the input, drastically reducing the number of parameters and introducing translation invariance.

These properties make CNNs particularly effective for both sequential data (e.g., time series, electromagnetic signals) and spatial data (e.g., images, coverage maps). By stacking multiple convolutional and pooling layers, CNNs are able to learn hierarchical representations, where early layers capture low-level features (edges, local patterns) and deeper layers capture more abstract structures.

#### 2.5.2.1 One-dimensional CNNs

One-dimensional CNNs (1D CNNs) were originally developed to handle sequential data, such as time series and signals, directly in their raw form [53]. 1D CNNs apply convolutional kernels along a single input dimension, enabling the extraction of local patterns from ordered data. This structure is especially appropriate for signals defined along one axis, such as temporal, spectral, or spatial sequences, and has been successfully applied to speech processing, biomedical signal analysis, vibration monitoring, and electromagnetic signal characterization.

The pioneering framework was introduced by Kiranyaz et al. in 2015 [53], demonstrating their effectiveness in processing electrocardiogram (ECG) signals for arrhythmia

detection. Since then, 1D CNNs have enabled significant advancements in signal classification tasks by autonomously learning features without the need for handcrafted preprocessing. Applications span across domains such as structural health monitoring [54, 55], fault detection in engines and electrical systems [56], and mechanical component diagnostics like bearing damage identification [56, 57].

The general architecture of a 1D CNN consists of a series of interconnected layers (Fig. 2.7), briefly outlined herein:

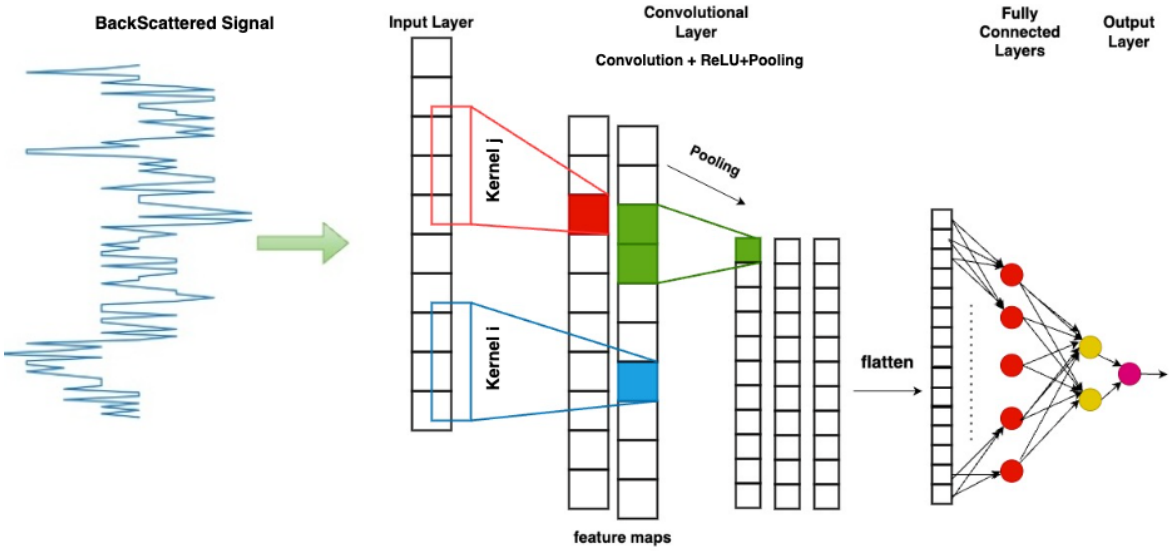



**Figure 2.7:** 1D CNN Architecture

*Convolutional Layer (CL):* Convolutional layers apply filters across the input data to capture local patterns. In a 1D CNN, the filters sweep along the input sequence to extract spatial or temporal dependencies, producing activation vectors (Figs. 2.7, 2.8) [53]. After convolution, a nonlinear activation such as ReLU is applied.

$$f(z) = \max(0, z), \tag{2.11}$$

to enable the network to capture complex patterns [46, 58].

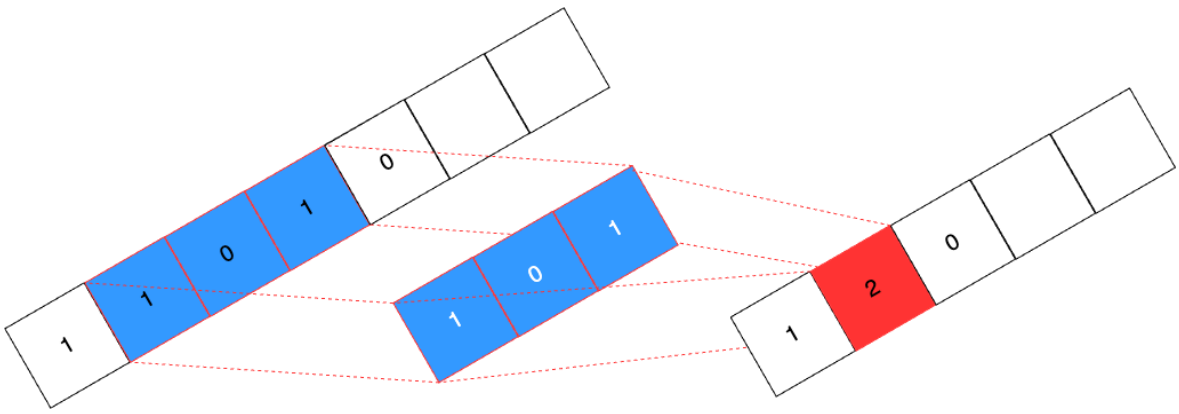



**Figure 2.8:** Convolutional layer in 1D CNN

*Pooling Layer (PLa):* Pooling reduces the size of feature vectors by down-sampling, retaining only the most relevant information while lowering computational complexity [26, 52]. Max pooling is widely used (Fig. 2.9), although average pooling is also an option.

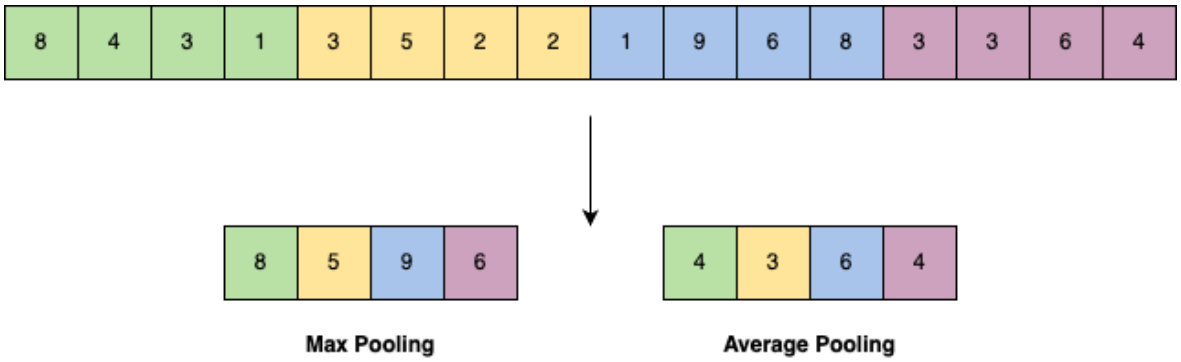



**Figure 2.9:** Max pooling and average pooling in 1D CNNs

*Fully Connected Layer (FCL):* Fully connected layers integrate the high-level features extracted by convolutional and pooling layers and map them to outputs, such as class labels or regression values [45].

#### 2.5.2.2 Two-dimensional CNNs

Two-dimensional CNNs are the standard in computer vision [52]. They operate on two-dimensional data (e.g., images) by extending the same principles introduced for 1D CNNs. Instead of filters sliding along a single axis, in 2D CNNs the kernels slide along both height and width, capturing local spatial features such as edges, corners, and textures.

Formally, given an input image $x \in \mathbb{R}^{H \times W}$ and a filter $w \in \mathbb{R}^{k \times k}$, the convolution at location $(i, j)$ is:

$$(x * w)(i, j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} x(i+m, j+n)\, w(m, n). \tag{2.12}$$

Apart from this dimensional extension, the overall architecture of 2D CNNs is conceptually identical to 1D CNNs: CL extract hierarchical features, PLa reduce dimensionality while retaining essential information, and FCLs perform the final mapping to outputs.

Because of their ability to automatically learn hierarchical spatial features, 2D CNNs form the backbone of modern computer vision tasks such as image classification, object detection, and semantic segmentation.

### 2.5.3 U-Net

U-Net is a CNN architecture specifically designed for image segmentation and other pixel-wise prediction tasks. It was originally proposed for biomedical image analysis,

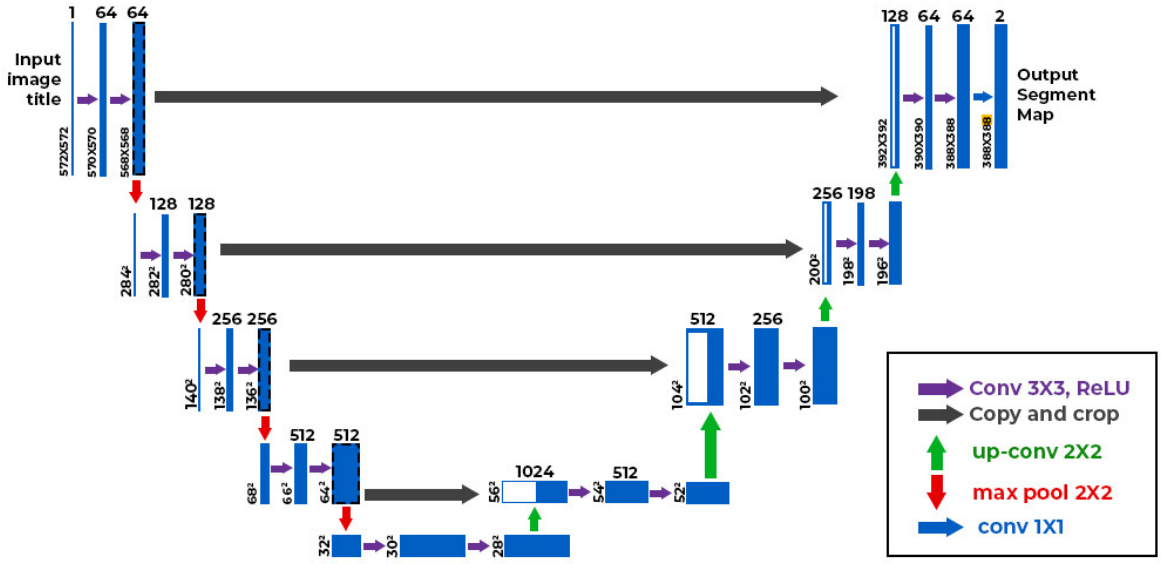



**Figure 2.10:** UNet Architecture.

but its versatility has made it widely adopted in computer vision and scientific domains [59].

#### 2.5.3.1 Architecture

The U-Net architecture follows an encoder–decoder structure with symmetric skip connections, as illustrated in Figure 2.10.

- **Encoder:** consists of successive applications of CL and PLa. Each step halves the spatial resolution while doubling the number of feature channels. This path captures the semantic context of the input.
- **Decoder:** consists of up-convolutions (transposed convolutions) and concatenations with feature maps from the encoder. This path progressively restores spatial resolution.
- **Skip connections:** the distinctive feature of U-Net. At each level, the feature maps from the encoder are concatenated with those of the decoder at the same resolution. This allows the network to combine high-level semantic information with low-level spatial details.

This architecture forms a "U" shape, hence the name U-Net.

#### 2.5.3.2 Mathematical Formulation

Let $E_l$ denote the feature map produced at the $l$-th encoder layer and $D_l$ the feature map at the corresponding decoder layer. The skip connection operation can be expressed as:

$$D_l = \text{UpConv}(D_{l+1}) \parallel E_l, \tag{2.13}$$

where UpConv denotes an up-convolution (transposed convolution), $\|$ is the concatenation operator, and $D_{l+1}$ is the feature map from the deeper layer of the decoder.

#### 2.5.3.3 Advantages

The strengths of U-Net include:

- High accuracy with limited training data, due to the efficient use of contextual information.
- Precise localization enabled by skip connections, which transfer high-resolution spatial features from the encoder to the decoder, allowing fine-grained spatial details to be preserved during reconstruction.
- Flexibility: applicable to both segmentation and general image-to-image regression tasks.

Because of these properties, U-Net and its variants are widely used in domains where spatially resolved predictions are required, including medical imaging, remote sensing, and wireless coverage map prediction.

## 2.6 Hyperparameter Tuning

The performance of ML models depends not only on their architecture but also on the selection of hyperparameters. Unlike trainable parameters (weights and biases), hyperparameters are set before training and control the learning dynamics, model complexity, and generalization ability. Hyperparameter tuning typically involves systematic exploration of candidate values, often using techniques such as grid search, random search, or Bayesian optimization [60, 61].

Beyond these general strategies, the choice of hyperparameters is closely tied to the specific model architecture. Each algorithm introduces its own set of tunable factors that strongly influence performance—for example, the number of trees in ensemble methods, the kernel parameters in SVMs, or the learning rate and batch size in neural networks. In the following, the key hyperparameters for the models employed in this thesis are briefly summarized.

### 2.6.1 XGBoost

XGBoost is a powerful ensemble method based on gradient-boosted decision trees. Key hyperparameters include:

- **Learning rate ($\eta$):** controls the contribution of each tree. Small values improve generalization but require more boosting rounds.
- **Number of trees ($n\_estimators$):** total number of boosting rounds.
- **Maximum tree depth ($max\_depth$):** controls model complexity. Deeper trees

capture more patterns but may overfit.

- **Subsample ratio:** fraction of training samples used to grow each tree, adding randomness for better generalization.
- **Colsample_bytree:** fraction of features sampled for each tree.
- **Regularization terms ($\lambda$, $\alpha$):** L2 and L1 penalties to reduce overfitting.

### 2.6.2 One-dimensional CNNs

In 1D CNNs, hyperparameters influence both architecture design and training dynamics:

- **Number of filters:** determines how many distinct features are extracted per convolutional layer.
- **Kernel size:** length of the convolutional filters; larger kernels capture broader patterns, while smaller kernels focus on fine details.
- **Stride:** step size of the convolution, affecting feature resolution and computational cost.
- **Pooling size:** controls the degree of dimensionality reduction in pooling layers.
- **Number of layers:** deeper networks extract more complex features but risk overfitting.
- **Batch size:** number of samples per training iteration, affecting gradient stability.
- **Learning rate:** governs the step size of parameter updates.

### 2.6.3 U-Net

While many hyperparameters are shared with standard convolutional neural networks, such as kernel size, number of feature maps, learning rate, and batch size, U-Net architectures introduce additional architectural choices related to their encoder–decoder structure and skip connections. The most relevant hyperparameters for U-Net include:

- **Depth of the network:** number of encoder–decoder stages, affecting the level of detail captured.
- **Number of feature maps:** controls the richness of extracted features at each stage and is often doubled after each downsampling step.
- **Kernel size:** convolution filter dimensions, typically $3 \times 3$.
- **Pooling and upsampling strategies:** max pooling vs. average pooling, transposed convolution vs. interpolation.
- **Loss function:** choice depends on the task (e.g., binary cross-entropy, Dice loss, or MSE+SSIM for regression maps).

- **Learning rate and optimizer:** Adam and SGD are commonly used.

### 2.6.4 Tuning Strategies

Hyperparameter tuning can be approached with:

- **Grid search:** exhaustive evaluation of predefined hyperparameter combinations.
- **Random search:** sampling random combinations, often more efficient in large spaces.
- **Bayesian optimization:** adaptive search strategy modeling the performance surface to select promising configurations.

The choice of tuning strategy depends on computational resources and the complexity of the model.

## 2.7 Training Stage

Loss functions play a crucial role in training machine learning and deep learning models by quantifying the discrepancy between the predicted output $\hat{y}$ and the ground truth $y$. The choice of loss function depends on the nature of the task—classification, regression, or image prediction.

- **Binary Cross-Entropy (BCE):** For binary classification tasks, the Binary Cross-Entropy loss is commonly used [62]:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\Big], \tag{2.14}$$

  where $y_i \in \{0,1\}$ denotes the true class label and $\hat{y}_i \in [0,1]$ the predicted probability. BCE penalizes incorrect predictions strongly, encouraging confident classification outcomes.

- **Mean Square Error (MSE):** For regression problems, where the goal is to predict continuous values, the Mean Squared Error is widely adopted:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{2.15}$$

  MSE minimizes the average squared difference between predicted and true values, emphasizing numerical accuracy.

- **Structural Similarity Index (SSIM):** In image prediction or reconstruction tasks, it is often important to preserve the spatial and structural characteristics of the target image. The Structural Similarity Index (SSIM) [63] measures perceptual similarity between $y$ and $\hat{y}$ as:

$$\mathrm{SSIM}(y,\hat{y}) = \frac{(2\mu_y\mu_{\hat{y}} + C_1)(2\sigma_{y\hat{y}} + C_2)}{(\mu_y^2 + \mu_{\hat{y}}^2 + C_1)(\sigma_y^2 + \sigma_{\hat{y}}^2 + C_2)}, \tag{2.16}$$

where $\mu_y$ and $\mu_{\hat{y}}$ are the local means of the ground truth and predicted images, $\sigma_y^2$ and $\sigma_{\hat{y}}^2$ their variances, and $\sigma_{y\hat{y}}$ their covariance. Constants $C_1$ and $C_2$ ensure numerical stability. The corresponding SSIM-based loss is defined as:

$$\mathcal{L}_{SSIM} = 1 - \text{SSIM}(y, \hat{y}). \tag{2.17}$$

- **Hybrid Loss:** For tasks that require both numerical accuracy and perceptual quality (e.g., coverage map or image prediction), a hybrid formulation can be employed:

$$\mathcal{L} = \alpha\mathcal{L}_{MSE} + (1 - \alpha)\mathcal{L}_{SSIM}, \tag{2.18}$$

where $\alpha \in [0, 1]$ controls the trade-off between pixel-wise precision and structural consistency.

## 2.8 Model's Evaluation

The evaluation of ML models requires task-specific metrics. Choosing the appropriate metric is crucial, since it directly affects how model performance is interpreted and optimized. In the following, we distinguish metrics for regression, classification (balanced and imbalanced), and image-based predictions.

### 2.8.1 Regression Metrics

In regression problems, the objective is to predict continuous values. Common metrics include:

- **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{2.19}$$

which penalizes large errors more heavily.

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{2.20}$$

expressed in the same units as the target variable, often more interpretable than MSE.

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{2.21}$$

less sensitive to outliers than MSE/RMSE.

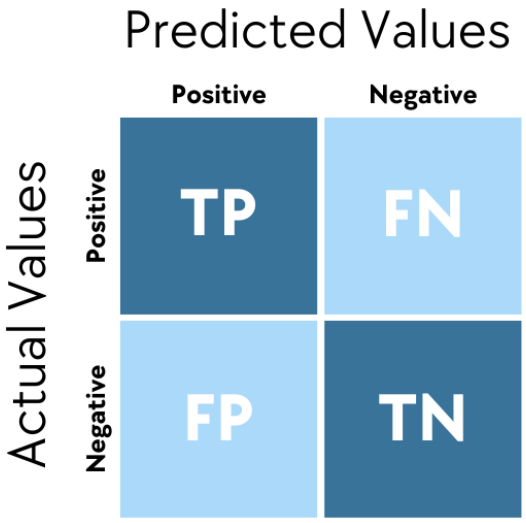



**Figure 2.11:** Confusion Matrix

- **Coefficient of Determination ($R^2$):**

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{2.22}$$

  measuring the proportion of variance explained by the model.

### 2.8.2 Classification Metrics

For binary classification, predictions are typically summarized in a **confusion matrix** (see Fig. 2.11), which counts True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

- TP: correctly predicted positive samples,
- TN: correctly predicted negative samples,
- FP: negative samples misclassified as positive,
- FN: positive samples misclassified as negative.

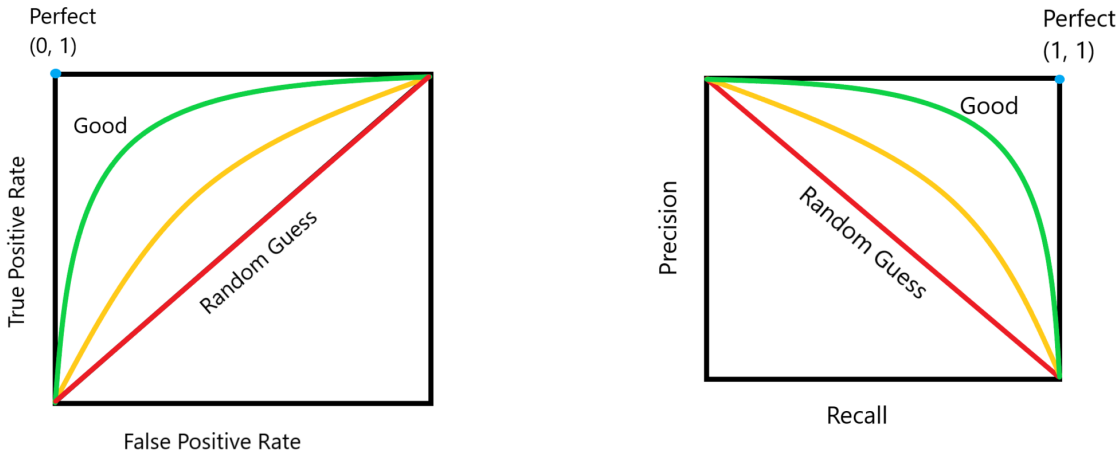
From this, several key metrics are derived:

- **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.23}$$

  Overall proportion of correct predictions.

- **Precision:**
$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2.24}$$
reliability of positive predictions.
- **Recall (Sensitivity):**
$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2.25}$$
ability to detect positive instances.
- **F1 Score:**
$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2.26}$$
harmonic mean of precision and recall, particularly relevant for imbalanced datasets.
- **Receiver Operating Characteristic (ROC)** illustrates the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at different decision thresholds.
- **Area Under Precision Recall Curve (AUPRC)**) provides a scalar summary of the classifier's ability to separate classes, with values closer to 1 indicating stronger discriminative power (Fig. 2.12).




**(a)** ROC Curve **(b)** Precision–Recall Curve

**Figure 2.12:** ROC and Precision–Recall curves as performance evaluation tools.

# 3

# ML for Wireless Propagation

Wireless propagation is driven by multiple contributing factors, including frequency, transmitter and receiver configuration, antenna type, and the geometric and material properties of the surrounding environment. This makes the relationship between system and environment-dependent parameters and the resulting propagation characteristics—such as received power, PL, shadowing, DS, and fast fading — highly complex and nonlinear. The accurate prediction of these propagation characteristics is fundamental for wireless network design, performance evaluation, and optimization.

Traditional wireless channel models have served as the cornerstone of wireless system design for decades. Empirical and semi-deterministic approaches such as the Okumura–Hata model [14], the COST-231 Hata extension [15], and analytical, physics-based models such as Walfisch–Bertoni [16] have been widely used in system planning. More recently, standardized models such as 3GPP TR 38.901 [17] and WINNER II [18] have provided unified frameworks for evaluating wireless systems across multiple propagation scenarios.

Although traditional models are simple to use and computationally efficient, they inherently rely on averaged representations of the propagation environment. These models are typically derived from large-scale measurement campaigns, simplified theoretical assumptions and/or sounded electromagnetic theory applied to some (over-)simplified description of the environment and are usually aimed at the mean values of key propagation markers—such as received power, path loss, or shadowing—as functions of a limited set of general link parameters (e.g., distance, frequency, and environment type). While some models may include environmental or geometrical information, they still offer only statistical, large-scale descriptions of propagation phenomena and cannot accurately predict point-specific variations that depend on the detailed structure and materials of the environment.

This level of abstraction is perfectly suitable for high-level system planning and standardization, where average performance indicators are sufficient. However, it may become inadequate when a more detailed or flexible modeling approach is required — for instance, for coverage prediction over specific maps, or to analyze how propagation

parameters vary with a wider set of input conditions (e.g., transmitter height, building geometry, or material properties). In such cases, more advanced deterministic or data-driven approaches are necessary to capture the site-specific behavior of the wireless channel.

A representative example of deterministic modeling is the RT technique, which explicitly accounts for the fundamental propagation mechanisms—reflection, diffraction, and scattering. RT relies on a detailed three-dimensional representation of the environment and reproduces the contribution of each multipath component, enabling highly accurate, point-specific predictions of received power, delay spread, and fast fading. When supported by precise geometrical and material data, RT results closely match experimental measurements even in complex environments such as industrial or dense urban areas [19, 20]. Owing to its physical rigor, RT serves not only as a reliable simulation tool but also as a ground-truth generator for validating and training data-driven models.

However, this level of accuracy comes at a cost. RT simulations are computationally intensive, as the number of rays grows rapidly with the environment's complexity and the number of interactions considered. Moreover, the need for accurate 3D models and detailed material characterization significantly increases the setup effort and computation time. For large-scale or real-time applications, such as dynamic network optimization or system-level coverage estimation, RT becomes impractical.

In this context, ML has emerged as an effective alternative or complement to traditional and deterministic modeling approaches [64–66]. Unlike empirical or physics-based methods, ML does not require an explicit analytical formulation of propagation phenomena. Instead, it learns the relationship between environmental and system parameters (e.g., geometry, frequency, transmitter configuration) and the resulting propagation markers (e.g., received power, PL, or DS) directly from data. Once trained, ML models provide predictions with accuracy comparable to RT but at a fraction of the computational cost. Furthermore, they can generalize across environments and conditions, offering a flexible framework to integrate measurements, simulations, and prior knowledge.

The potential of ML in wireless propagation modeling extends across multiple levels of abstraction, depending on the target application and the type of data available. ML can be used to support, enhance, or even replace traditional modeling and simulation techniques in a variety of propagation-related tasks, briefly outlined herein.

## 3.1 LoS Prediction

In wireless communications, LoS refers to the direct, unobstructed path between a Tx and a Rx . When LoS exists, the signal typically experiences lower attenuation, reduced multipath distortion, and more predictable propagation. In contrast, in NLoS conditions, obstacles such as buildings, machinery, or foliage block the direct path, and

signals reach the receiver only through reflection, diffraction, or scattering, which leads to fading, DS, and signal degradation.

LoS awareness can be beneficial in several applications, such as:

- **Beamforming:** steering antenna patterns toward the LoS direction often yields an optimal or near-optimal solution.
- **Localization:** methods like Time of Arrival (ToA) achieve much higher accuracy under LoS conditions.
- **5G and beyond:** especially at mmWave frequencies, LoS presence becomes a crucial issue due to the severe blocking effects of obstacles compared to lower frequencies.

For these reasons, accurate LoS detection and modeling have become central problems in modern wireless system design, spectrum sharing, and network optimization [67–72].

## 3.2 Channel Parameter Estimation

Propagation parameters such as the PLE, shadowing standard deviation, DS, and Rice factor are fundamental descriptors of the wireless channel. These parameters jointly define how the signal power decays with distance, how large-scale variations occur due to obstacles and environment irregularities, and how multipath propagation induces both temporal dispersion and rapid amplitude fluctuations.

In narrowband systems, PLE and shadowing primarily determine large-scale attenuation, influencing link budget and coverage performance. The PLE quantifies the average rate of signal decay with distance, while the shadowing deviation measures the possible deviation from the average expectation due to the degree of obstruction affecting specific receiving location compared to the average case. In addition, small-scale multipath interference gives rise to fast fading, resulting in rapid amplitude fluctuations over short spatial or temporal intervals.

In wideband systems, propagation effects also include temporal dispersion, characterized by DS. When the signal bandwidth becomes sufficiently large, different frequency components experience different levels of attenuation and phase shift due to multipath propagation, leading to frequency-selective fading. In the time domain, this phenomenon corresponds to inter-symbol interference, as symbols transmitted over distinct multipath components partially overlap at the receiver. These effects can significantly degrade system reliability, synchronization, and data throughput—particularly in cluttered or metallic environments such as industrial facilities, where strong reflections and obstructions are common.

Understanding and accurately estimating these parameters are crucial for system design and optimization. They directly impact key aspects such as power control, handover management, channel equalization, and error performance. Conventional approaches rely on curve-fitting or analytical models, which often fail to capture the nonlinear and

environment-dependent nature of propagation.

ML offers an effective alternative by learning the underlying relationships between environmental features (e.g., geometry, materials, transmitter-receiver configuration) and channel parameters from data. Once trained, ML models can rapidly predict PLE, shadowing deviation, DS, and fast fading across diverse scenarios without extensive measurements or computationally expensive simulations. This enables adaptive, data-driven channel characterization and supports intelligent wireless system planning and optimization.

## 3.3 Coverage Map Prediction

Radio Environmental Map (REM)s represent the spatial distribution of radio propagation parameters—typically received power ($P_r$) or PL—over a target service area. In this context, PL is not modeled as a function of the Tx–Rx distance only, but as a location-dependent quantity that inherently captures the impact of the surrounding environment, including obstructions, reflections, diffraction, and scattering. Both $P_r$ and PL are key quantities in wireless propagation analysis. $P_r$ is affected by various factors such as free space propagation, and wave interaction with obstacles such as reflections, diffraction, and scattering, resulting in large- and small-scale fading effects [1].

REM can support wireless network optimization and planning, enabling efficient spectrum management and reliable connectivity. In fact, they help identifying wireless coverage holes, and therefore network operators enhancing the quality of service by strategically deploying additional infrastructure [73]. In dense urban environments, REMs are used to optimize resource allocation, ensuring efficient use of bandwidth and power for cellular and IoT networks [74]. They can also play a critical role in localization, where accurate REMs improve positioning accuracy for GPS-denied environments [75].

In the framework of emerging technologies and applications, REMs are beneficial in path planning for unmanned aerial vehicles (UAVs), ensuring continuous connectivity during autonomous flights [76, 77]. Moreover, they are increasingly relevant in physical layer security, where knowledge of RF propagation assists in detecting potential eavesdropping or jamming threats [78].

Finally, autonomous driving can rely on accurate REM prediction to maintain stable vehicle-to-everything (V2X) communication, ensuring real-time data exchange for traffic management and safety-critical applications [79]. In the context of the ongoing evolution of wireless communication systems, the importance of REMs extends beyond traditional mobile networks, making them a valuable asset in next-generation technologies such as 5G, 6G, and smart city deployments.

In order to effectively support such a broad set of wireless services and applications, REMs must be point-specific, i.e., they should track the difference in propagation over

the multitude of spatial locations spread all over the maps. To this aim, propagation data to fill the REMs can be collected either through extensive measurement campaigns or deterministic field prediction models, like RT or Ray Launching (RL).

Unfortunately, field measurements often require expensive equipment, while RT/RL accuracy can be heavily limited by imprecision in the environment database. Moreover, they both can be time-consuming, especially for large measurement/simulation areas. These constraints drive the need for more efficient and flexible prediction methods.

ML offers a powerful alternative to conventional approaches for REM generation. By learning the complex relationship between environmental features (e.g., building geometry, transmitter position, and frequency) and radio propagation characteristics, ML-based models can efficiently predict received power or PL over large areas without the need for exhaustive measurements or computationally expensive RT simulations. CNNs, and in particular encoder–decoder architectures such as U-Net, have shown remarkable success in this task by treating the propagation problem as an image-to-image translation process, where the input (e.g., building and transmitter maps) is mapped to the corresponding coverage image [80, 81].

This data-driven framework enables fast and accurate estimation of site-specific coverage maps while maintaining performance comparable to deterministic methods and significantly reducing computational cost. Consequently, ML-driven REM prediction represents a key enabler for scalable, adaptive, and real-time wireless network design and optimization.

## 3.4 Object Recognition

OR aims at object identification [82]. OR can be carried out at two different levels: *(i)* class level (classification), if the goal is the recognition of the type of item among a set of categories or classes (e.g. a car or a bicycle); *(ii)* item level, if the task is the precise recognition of the object under observation (e.g. a specific car model among different cars). The former issue is addressed in this study. Somehow in the middle between detection and recognition lies object characterization, which is aimed at getting some specific properties of an item (e.g., the material it is made of) [83]. OR has many applications, such as robot navigation, medical diagnosis, security, industrial detection, and information retrieval [82, 84, 85]. Moreover, the potential of OR extends to supporting individuals with visual impairments, by providing real-time information about the surrounding environment, e.g. assisting in locating personal belongings and navigating around obstacles [86].

Nowadays, the most commonly used methods for OR apply pattern recognition algorithms to images or videos, i.e. they rely on a vision-based approach [87–89]: these techniques process visual input from cameras or similar sensors - i.e. images - to detect and recognize objects. There are various drawbacks to vision-based OR systems including: *(i)* they demand substantial computational resources, making them

sometimes unsuitable for real-time applications or deployment on devices with limited processing power and memory [90]; *(ii)* images may be impractical in environments where cameras are forbidden, or their quality might be sometimes compromised by some environmental factors, resulting in noise or unreliable data [91]; *(iii)* relying on video cameras for OR introduces potential security risks, as these systems are often connected to the internet, making them vulnerable to hacking. This could allow unauthorized access, and the fraudulent acquisition of images of sensitive environments or industrial processes [92].

Another potential approach is based on electromagnetic waves. As well proved and explained in spectroscopy-related literature [93], objects have a unique response to incoming electromagnetic waves, depending on their material, size, and shape. When electromagnetic waves impinge on an object, a backscattered wave is radiated all around, which is univocally related to the item, as it depends on its molecular structure and specific size, shape, and orientation [94].

First investigations on OR date back to more than fifty years ago [95, 96], when image / signal processing - e.g. applied to radar signal returned by the target object - was basically relied on to accomplish the recognition task [97, 98]. Not suprisingly, ML techniques have been replacing signal processing over the last decades, and are likely to be today the state of the art for OR.

Each of these tasks addresses a different aspect of the wireless channel, ranging from binary classification (e.g., LoS vs. NLoS) to regression and spatial prediction problems (e.g., path loss or coverage estimation).

The effectiveness of ML-based propagation modeling strongly depends on the quality and diversity of the data used for training and validation. Since ML models learn the underlying relationships between environmental descriptors and channel behavior from examples rather than from analytical equations, the availability of representative datasets becomes a fundamental prerequisite.

The following section discusses two main strategies for dataset generation, highlighting their respective advantages, limitations, and suitability for different applications.

## 3.5 Dataset Generation

The development of reliable ML-based propagation models requires representative datasets that accurately describe the interaction between electromagnetic waves and their surrounding environment. Two primary approaches are commonly adopted for dataset generation:

- **Measurement campaigns**, which provide real-world data acquired through dedicated experimental setups;

- **Electromagnetic simulations**, which synthetically reproduce propagation phenomena using deterministic models such as RT.

### 3.5.1 Measurements

Measurement-based datasets are obtained through field campaigns using specialized equipment such as channel sounders, vector network analyzers, and calibrated antennas. Tx and Rx are placed across the area of interest, and received signals are collected under different conditions. From the recorded data, propagation parameters such as PL, DS, and shadowing standard deviation ($\sigma$) can be extracted through post-processing.

**Advantages:**

- Provide ground-truth data that inherently capture all propagation mechanisms, hardware effects, and environmental peculiarities.
- Enable the analysis of real-world non-idealities (e.g., antenna misalignment, noise, and interference), improving model robustness in practical deployments.
- Essential for validating and calibrating simulation tools or ML-based models against physical reality.

**Disadvantages:**

- Measurement campaigns are resource-intensive, requiring expensive instrumentation, expert personnel, and long acquisition times.
- Data collection is often limited to specific environments or frequency bands, making generalization difficult without additional campaigns.
- Access restrictions (e.g., in industrial or private areas) and environmental variability can hinder repeatability and scalability.

### 3.5.2 Simulations

Electromagnetic simulations offer an efficient alternative to measurements by computing channel characteristics based on physical principles. Among these, Ray Tracing (RT) represents the most popular and effective method, as it models reflection, diffraction, and scattering processes in realistic three-dimensional environments, providing high-fidelity synthetic datasets.

**Advantages:**

- Allow full control over environmental, frequency, and antenna parameters, enabling systematic analysis under reproducible conditions.
- Significantly reduce cost and time compared to large-scale measurement campaigns.
- Facilitate the creation of large, diverse datasets that can cover multiple propaga-

tion scenarios, frequencies, and configurations.

- Can be easily used to generate ground-truth data for ML model training and testing, even in virtual or hard-to-access environments.

**Disadvantages:**

- Accuracy depends heavily on the quality of the 3D environment model and the electromagnetic characterization of materials.
- Computational cost increases rapidly with the number of interactions and environmental complexity.
- Simplifications in modeling (e.g., ignoring diffuse scattering or polarization effects) may lead to discrepancies with real measurements.

## 3.6 Ray Tracing for Dataset Generation

RT is particularly valuable for ML since it produces large-scale, diverse, and fully labeled datasets. It simulates electromagnetic propagation by tracing possible paths (*rays*) between Tx and Rx within a given environment [20, 21].

It is based on the high-frequency approximation of Maxwell's equations, where electromagnetic waves are modeled as rays that interact with objects through reflection, diffraction, and scattering. By combining Geometrical Optics (GO) and the Uniform Theory of Diffraction (UTD), RT provides a physically grounded way to estimate received power, PL, DS, and other channel parameters with high spatial resolution [99–101].

Unlike empirical models, which approximate the average propagation behavior, RT explicitly accounts for the environment's geometry and material composition. This makes it especially valuable for site-specific predictions and for generating realistic synthetic datasets that can be used to train and validate ML-based propagation models.

RT algorithms can be implemented in different forms, mainly:

- **RL:** Rays are uniformly launched from the transmitter in predefined angular directions. Their interactions (reflections, diffractions, scatterings) are tracked until they reach the receiver or fall below a power threshold [20, 102].
- **Image-based RT:** Starting from the actual transmitter, visibility relationships within the propagation environment are managed through the recursive introduction of virtual transmitters. Each virtual transmitter is associated with a specific wall portion—to model reflection and diffuse scattering—or with an edge—to account for diffraction. These relationships are organized into a visibility tree, where each node represents a virtual transmitter and each branch corresponds to a particular interaction type. When a receiver falls within the visibility region of a virtual transmitter, a leaf is added to the tree, which is then traced back to reconstruct the exact propagation path of the corresponding ray [20].

Modern RT tools, implement optimized ray-based algorithms to achieve a balance between accuracy and computational efficiency.

Depending on the propagation scenario, they can adopt either a *RL* or an *Image-based RT* strategy. In both cases, the simulation typically proceeds in two main phases:

1. **Geometrical Phase:** Rays are generated and propagated through the environment, accounting for each possible interaction event up to a maximum order $N_{ev}$.
2. **Field Computation Phase:** The complex field contribution of each valid ray is calculated at the receiver, considering amplitude, phase, polarization, and path delay [20].

Among deterministic approaches, both RT and Ray RL techniques are widely employed to model site-specific propagation. However, the focus in this work is primarily on RT-based simulations, as the adopted tools rely on ray-optical modeling and recursive visibility analysis. When accurately parameterized, RT simulations can provide predictions that closely match real-world measurements [19, 20]. Nevertheless, the reliability of the results depends critically on the correctness and completeness of the input files that describe the scenario.

### 3.6.1 Input Files

Every RT simulator requires a set of input files that describe the geometry, electromagnetic properties, and configuration of the simulation environment. Regardless of the specific software implementation, these inputs generally fall into three categories:

- **Environment description:** defines the geometrical layout of the scenario (e.g., buildings, walls, terrain) and the electromagnetic properties of the involved materials, such as relative permittivity and conductivity.
- **Antenna configuration:** specifies the position, orientation, and radiation pattern of transmitters and receivers, including polarization and gain characteristics.
- **Simulation parameters:** control the propagation modeling process, such as maximum interaction order (number of reflections, diffractions, scatterings), power thresholds, and output data format.

These elements together allow the RT engine to reconstruct the environment, trace rays, and compute field contributions with high spatial resolution. The specific organization of these files may vary among simulators, but the underlying information is conceptually the same.

This structure highlights that while both tools share the same fundamental RT framework, the indoor implementation requires a more detailed geometric and electromagnetic description to accurately capture multipath propagation in confined spaces.

**Table 3.1:** Comparison of Input Structures in ScatRay and 3DScat.

| **Feature** | **Outdoor** | **Indoor** |
|---|---|---|
| Geometry representation | 2D building outlines combined with height data | Full 3D surfaces including walls, floors, ceilings, and objects |
| Material definition | One material per building, uniform electromagnetic properties | Per-surface material definition with specific permittivity and conductivity values |
| Antenna files | Tx and Rx locations and radiation pattern | Tx and Rx locations and radiation pattern |
| Supported mechanisms | Reflection, diffraction, and scattering | Reflection, diffraction, transmission, and scattering |
| Typical use case | Urban or suburban outdoor propagation studies | Industrial or office indoor environments |

### 3.6.2 Output Files

The RT simulator produces several output files that describe the propagation results in different formats, depending on the selected options.

These files contain both intermediate and final data, including ray paths, received power levels, and power delay profiles, i.e., the distribution of the received signal power as a function of propagation delay, which characterizes the temporal dispersion introduced by multipath propagation. These quantities are later processed to generate coverage maps or other channel descriptors.

- **Ray File:** Contains the geometric description of all valid propagation paths connecting the transmitter to the receiver. For each ray, the file stores:
  - The sequence of interaction points (coordinates in 3D space);
  - The type of each interaction (reflection, diffraction, scattering);
  - The total path length, delay, and cumulative attenuation.

  This file is mainly used for visualization or validation of propagation mechanisms.
- **Received Power File:** Provides the received power at each receiver point. Depending on the configuration, the received field can be computed as:
  - *Coherent sum:* phase information of each ray is included before summation;
  - *Incoherent sum:* powers of all rays are added without phase coherence.

  The results are typically expressed in dBm and can be used to generate received power or path loss maps.
- **Power Delay Profile File:** Describes the time-domain multipath structure at

each receiver. Each entry provides the amplitude and delay of the contributing rays, allowing the computation of delay spread (DS) and related time-domain statistics. This file is crucial for wideband channel characterization.

Ray-based propagation models are capable of providing a comprehensive channel characterization that extends beyond large-scale narrowband metrics. When properly configured, they can capture fast fading and temporal, frequency, and spatial dispersion effects, enabling a detailed description of the radio channel's behavior. The combination of the Received Power and Power Delay Profile files provides all the quantitative information needed to reconstruct the channel characteristics across the simulated area. From these outputs, it is possible to generate 2D or 3D coverage maps of received power, path loss, delay spread, or LoS probability. Such datasets are used in this work as input for the development of ML-based predictors that emulate RT accuracy at a fraction of the computational cost.

In the following, the discussion specifically refers to the RT tools employed for dataset generation in this study. In particular, outdoor RT simulations target large-scale modeling of dominant propagation mechanisms (reflection, diffraction, scattering) using simplified building blocks, whereas indoor RT configurations aim at small-scale accuracy by including transmission through walls and detailed material interactions. This distinction affects both the input data structure and the computational complexity of the simulation.

# 4

# Results and Discussions

The propagation channel plays a central role in determining the performance and reliability of wireless communication systems. Accurate modeling of the channel is essential for tasks such as link budget analysis, network planning, interference management, and the design of advanced techniques including MIMO and beamforming. However, the wireless channel is inherently complex and influenced by multiple phenomena occurring at different spatial and temporal scales.

In this chapter, several aspects of channel modeling are investigated through ML and data-driven approaches. The work is structured into four main themes:

- **LoS detection**, which represents one of the fundamental propagation features and is critical for applications such as beamforming, link adaptation, and channel state characterization.
- **Narrowband channel modeling**, focusing on large-scale parameters such as the PLE, shadowing, fast fading and coverage prediction.
- **Wideband channel modeling**, where multipath-induced effects such as DS are analyzed.
- **Object recognition**, which represents an emerging application at the intersection of communication and sensing. By analyzing the electromagnetic response of the environment, ML models can identify or classify objects based on their scattering or reflection characteristics. This capability is particularly relevant for joint communication and sensing systems, enabling environment awareness for applications such as industrial monitoring, autonomous navigation, and intelligent wireless environments.

Each topic is addressed in a dedicated section, combining tabular data analysis, image-based DL approaches, and empirical modeling where appropriate. Together, these results illustrate how ML can complement deterministic and empirical models, enabling scalable, accurate, and flexible wireless propagation modeling across diverse environments.

## 4.1 Line-of-Sight Detection

### 4.1.1 Tabular-Data Modeling

The tabular data approach frames LoS detection as a supervised classification task, where input features are numerical descriptors of the environment geometry. These features are organized in a structured dataset and used to train ML models that output the probability of LoS. This method is efficient, scalable, and independent of direct channel measurements, making it well-suited for large-scale deployment.

#### 4.1.1.1 LoS Assessment in industrial Environments

This study is based on the work presented in "Line of Sight Detection in Industrial Environment: A Machine Learning Approach" [103]. The paper investigates the application of ML techniques for LoS identification in industrial environments using tabular data. The work forms the foundation for subsequent developments on channel characterization and propagation modeling discussed in this thesis.

**Dataset Generation** A large dataset is necessary to train and test ML models in the context of LoS detection in industrial applications. An industrial environment with dimensions of $100\times100\times10$ m$^3$ is investigated to approximate real-world scenarios. This setting encompasses the intricacies of industrial landscapes and provides a framework for the LoS evaluation.

Machines are a key component of this environment. The different sizes and spacings of these machines mirror the variety of machinery found in actual industrial settings. The following is a list of the parameters considered in this study:

- **Machine Size (MS):** lengths of 2, 3, 4, and 5 m;
- **Machine Height (MH):** values of 1.5, 2, 2.5, 3, and 3.5 m;
- **Spacing between machines (SP):** inter-machine spacing values of 2, 2.5, 3, 3.5, and 4 m.

A total of 100 various scenarios are created by combining these parameters, each of which represents a different arrangement of machines in the industrial environment. These case studies offer a thorough perspective on establishing LoS by taking into account how machine dimensions, heights, and configurations affect signal propagation.

To generate a dataset, the transmitter location is equally important. Twelve transmitters, each with different x and y coordinates, simulate different positions throughout the industrial environment. The various height categories found in industrial installations are taken into account by considering:

- **Transmitter Height (TxH):** 1, 2.5, 4, and 5.5 m

First, receivers are placed along the aisles between machines, representing potential positions of mobile robots, which motivates a dense spatial sampling of LoS conditions.

Second, the receivers are located on the sides of the machines themselves, corresponding to fixed installation points and thus a static receiver configuration. In both cases:

- **Receiver Height (RxH):** 2, 3, 4, and 5 m

A new parameter T, which denotes the proportion of randomly eliminated machines, is introduced to increase the realism of situations. The effect of obstacles on the LoS detection for values of T ranging from 0.1 to 0.5 is studied:

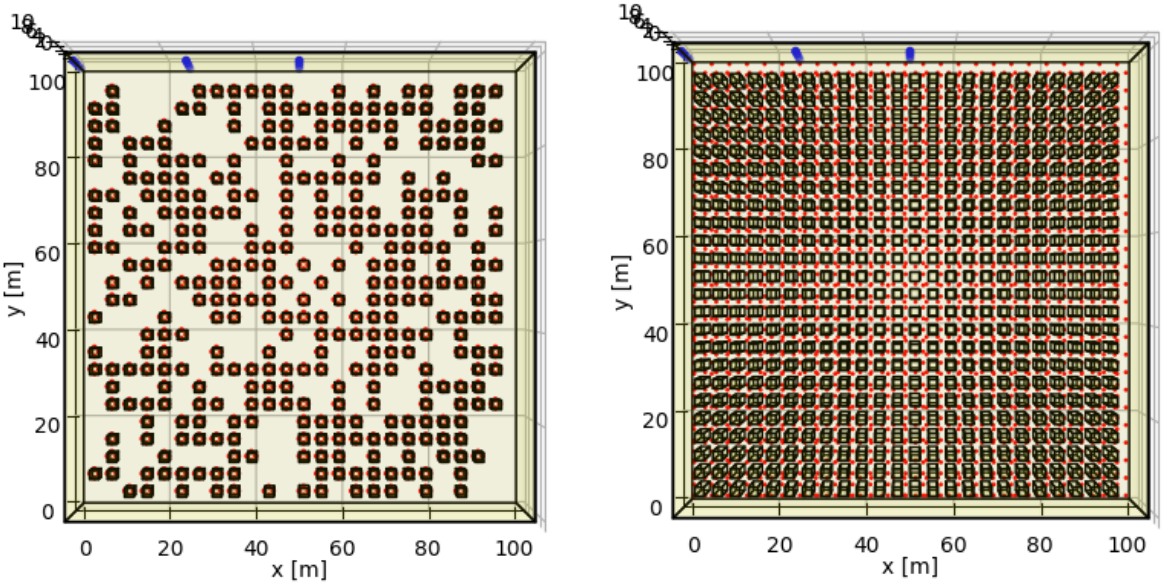
- **Fraction of machine removal ($T$):** 0.1, 0.2, 0.3, 0.4, 0.5

.

Therefore, a new feature called Machine Density can be added to datasets with the use of this new parameter T. In general, the following formula can be used to estimate machine density:

$$MD \approx (1 - T) \cdot \left( \frac{MS}{MS + SP} \right)^2, \tag{4.1}$$

Thanks to the introduction of the T parameter, two different types of scenarios can be explored: regular and irregular. An example of both cases is shown in Fig. 4.1




(a) Irregular machine layout. (b) Regular machine layout.

**Figure 4.1:** Examples of industrial environment layouts used for dataset generation.

Combining these factors—various machine configurations, transmitter locations, and receiver settings—leads to the creation of a rich dataset. For robust LoS detection in challenging industrial environments, this dataset is the starting point for training and assessing ML algorithms.

The use of ray-based simulation emerges as a valuable strategy in the goal of comprehensive LoS assessment inside complicated industrial layouts. The procedure entails

converting the many scenarios created in the previous part into input maps for the simulation program, after which output files are used to create the dataset.

RT simulation tool developed at the University of Bologna [104] allows for wireless channel simulations. In the context of LoS assessment, the RT simulation tool can easily evaluate LoS condition between any Tx and Rx by tracing only the direct path: if the direct path is present the LoS condition occurs, if not present, the path is obstructed. In this way simulations are very fast as RT has to account only for the direct path evaluation and interactions of higher order are not required.

LoS/NLoS evaluations returned by the RT simulations can be arranged in a dataset together with the features describing the industrial environment and the Tx and Rx positions, as outlined in Figure 4.2. It represents a database for training and testing ML tools aimed at LoS detection.

| Tx_X | Tx_Y | Tx_Z | Rx_X | Rx_Y | Rx_Z | MS | MH | SP | link distance | LoSlabel | id | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 25.00 | 99.99 | 1.0 | 1.00 | 1.00 | 1.00 | 2 | 1.5 | 2.0 | 101.86 | 0 | 1 | 0.0 |
| 25.00 | 99.99 | 1.0 | 1.00 | 3.51 | 1.00 | 2 | 1.5 | 2.0 | 99.42 | 0 | 1 | 0.0 |
| 25.00 | 99.99 | 1.0 | 1.00 | 6.03 | 1.00 | 2 | 1.5 | 2.0 | 96.98 | 0 | 1 | 0.0 |
| 25.00 | 99.99 | 1.0 | 1.00 | 8.54 | 1.00 | 2 | 1.5 | 2.0 | 94.55 | 0 | 1 | 0.0 |
| 25.00 | 99.99 | 1.0 | 1.00 | 11.05 | 1.00 | 2 | 1.5 | 2.0 | 92.12 | 0 | 1 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 99.99 | 50.00 | 5.5 | 88.15 | 101.00 | 3.25 | 5 | 3.5 | 4.0 | 52.40 | 0 | 400 | 0.5 |
| 99.99 | 50.00 | 5.5 | 90.62 | 101.00 | 3.25 | 5 | 3.5 | 4.0 | 51.90 | 0 | 400 | 0.5 |
| 99.99 | 50.00 | 5.5 | 93.08 | 101.00 | 3.25 | 5 | 3.5 | 4.0 | 51.52 | 0 | 400 | 0.5 |
| 99.99 | 50.00 | 5.5 | 95.54 | 101.00 | 3.25 | 5 | 3.5 | 4.0 | 51.24 | 0 | 400 | 0.5 |
| 99.99 | 50.00 | 5.5 | 98.00 | 101.00 | 3.25 | 5 | 3.5 | 4.0 | 51.09 | 0 | 400 | 0.5 |

**Figure 4.2:** Example of dataset in the industrial environment. Regular and irregular machine layouts are combined with different Tx/Rx positions and removal fractions $T$ to create diverse scenarios for LoS detection.

**Features and Learning Model** The input features can be divided into two categories, system-dependent parameters and environment-dependent parameters. System-dependent parameters are those independent of the propagation environment, such as TxH and RxH. Environment-dependent parameters are those determined by the geographic environment including the MS, MH, SP, and MD.

The dataset was split into training, validation, and testing sets following a 60% / 20% / 20% partition, respectively. Each scenario comprises on the order of millions of data instances, ensuring statistical representativeness across all splits. Since the LoS detection task results in an imbalanced class distribution, with non-line-of-sight samples occurring more frequently than line-of-sight ones, standard accuracy-based evaluation would be misleading. For learning, Gradient Bossting Decision Tree (GBDT) were selected, specifically XGBoost, due to its high performance on tabular data. The logistic loss function was optimized, and evaluation relied on metrics robust to class

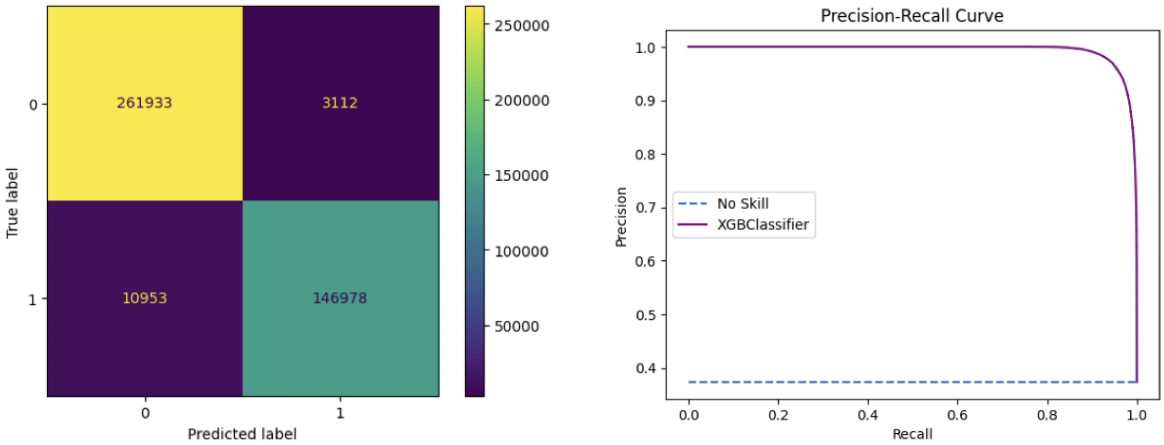
imbalance, including the F1-score, area under the precision–recall curve (AUPRC), and precision–recall metrics.

**LoS Detection for Each Specific Link** The first task addresses the binary classification of LoS/NLoS conditions for each Tx–Rx link in the industrial Manhattan-like environment. Four scenarios were investigated, considering both regular maps and irregular maps. The selected cases are summarized in Table 4.1.

**Table 4.1:** LoS detection scenarios and GBDT classification performance.

| Case | Map and RxType | $T$ | AUPRC | F1-score |
|---|---|---|---|---|
| 1 | Regular - machines | 0.0 | 98% | 98% |
| 2 | Regular - Robo | 0.0 | 98% | 98% |
| 3 | Irregular - Robo | 0.1 | 96% | 94% |
| 4 | Irregular - Robo | 0.5 | 92% | 90% |

According to the evaluation recorded in Table 4.1 and Figs. 4.3a and 4.3b, the results are accurate and trustworthy when it comes to estimating the LoS/NLoS scenario for each link. The confusion matrix in Fig. 4.3a demonstrates how the model can forecast both NLoS conditions and LoS links since TP (green in Fig. 4.3a) and TN (yellow in Fig. 4.3a) are significantly greater than FP (purple) and FN (purple), respectively. The precision-recall curve in Fig. 4.3b demonstrates how well the model performs as a classifier and how much better it performs than a random classifier (a model with no skill).




(a) Confusion matrix for Case 4 (irregular map, $T = 0.5$).

(b) Precision–recall curve for Case 4 (irregular map, $T = 0.5$).

**Figure 4.3:** Examples of industrial environment layouts used for dataset generation.

Training on a regular map and testing on an irregular map, however, can result in an additional evaluation. For this, the regular map is used for training, while T=0.5 (removing half of the machines from the map) which represents the worst-case scenario

for the irregular map is chosen for testing. In this case, The AUPRC is 86%, while the $F_1$ score is 82%. Although the results are less favorable than in prior instances, they can still be trusted to accurately anticipate LoS/NLoS. The advantage of this task is that RT simulations are carried out just on an ideal, regular case (T=0), and the proposed classifier trained on such dataset can be then quite reliably extended to predict LoS/NLoS occurrence in irregular layout even in scenarios where the detailed description of the environment is not presented.

**LoS Probability Evaluation** An essential part of this research is the evaluation of LoS likelihood. In Fig. 4.4 the LoS probability is plotted against the link distance for T=0.2 as an example. In general, the greater the distance, the lower the probability, which drops from about 70% at 10 m to approximately 35 % beyond 60 m. It is worth noting the comparison with the reference LoS probability model proposed by ITU [105]. Different from the proposed ML-based approach, the ITU model is a general indoor model, i.e. not tailored to the industrial case, and consisting of a simple, fixed exponential formula without any possibility of tuning.

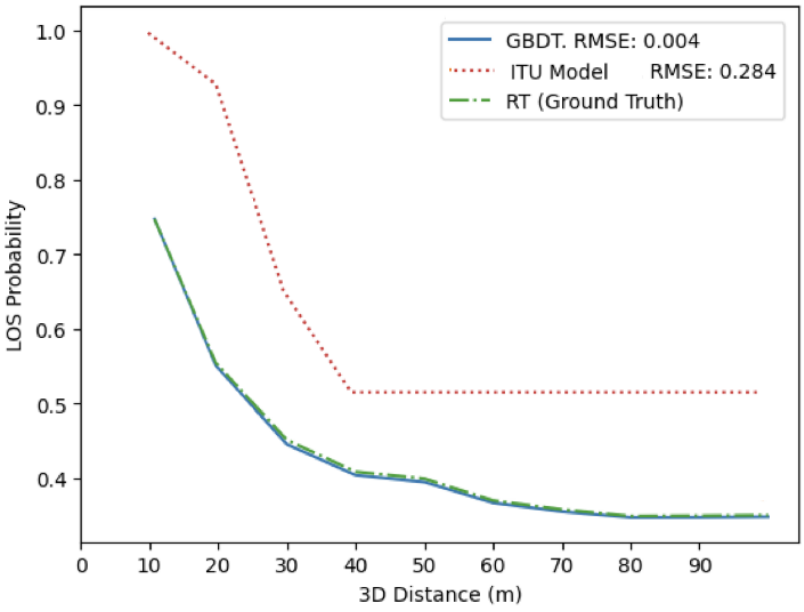



**Figure 4.4:** Comparison of ML-based LoS probability with ITU reference model.

Fig. 4.4 shows how well the model performs in terms of LoS probability evaluation. The RMSE value sheds light on the algorithm's capacity to capture the underlying patterns of signal propagation and offers insights into the precision of the LoS probability forecasts.

**Feature Importance** The sensitivity of the LoS condition to the features has been investigated utilizing some Explainable AI post-processing. Plots in Fig. 4.5 show that the height of the Transmitter, Receiver, and machines has a great impact, that is rather physically sound. More surprisingly, the spatial deployment of the machinery is likely

to be less important, as the overall LoS probability only slightly decreases for increasing the value of the MD, at least in the range considered. The observed lower impact of MD on the final output could be attributed to its comprehensive consideration of both MS and SP, potentially leading to their mutual cancellation.

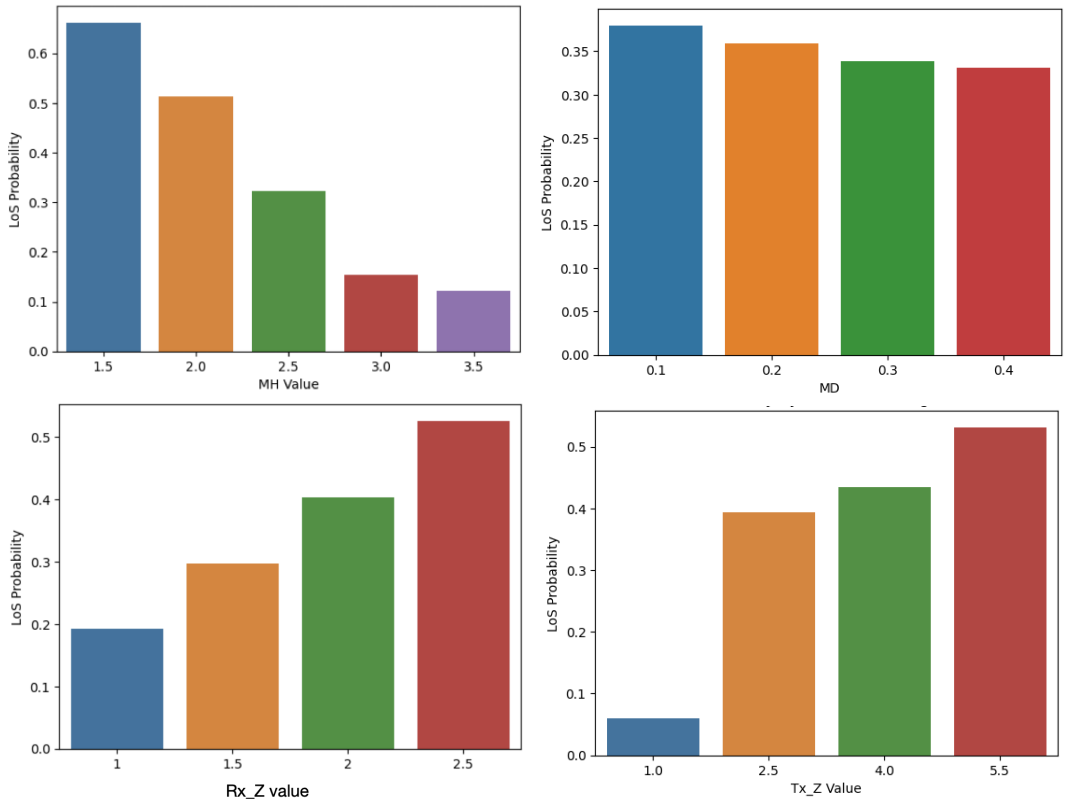



**Figure 4.5:** Feature importance analysis: influence of MH, MD, RxH(Rx_Z), TxH(Tx_Z).

**Extension to Outdoor Urban Scenarios** The same methodological framework can be extended to outdoor, Manhattan-like urban environments by substituting the geometric descriptors. Instead of machine-related features (MS, MH, SP), urban features such as building height, street width, and transmitter elevation can be used as input parameters.

This generalization has been explored in a separate study "Machine Learning-Assisted LoS Detection in Manhattan-like Urban Scenarios" [106], demonstrating that the proposed ML-based LoS detection approach is scalable and adaptable across both industrial and urban propagation contexts.

#### 4.1.1.2 Empirical LoS Formulas for Industrial Environments

This study is based on the work presented in "A Simple Empirical Expression for Line of Sight Probability in Industrial Environment" [107]. In this extension, an analytical expression for the LoS probability was proposed, explicitly considering the relative

heights of Txs, Rxs, and industrial clutter. The resulting model achieved improved prediction accuracy compared to existing ITU [105] and ETSI [17] formulations while maintaining a simple, physically interpretable structure.

The derivation of empirical formulas for LoS probability in industrial environments is motivated by the need for lightweight models that can be directly integrated into system-level simulations. While ML-based methods provide high accuracy, they often require computationally demanding training and inference steps. Analytical expressions, on the other hand, offer simple and fast evaluation, enabling scalability while still capturing the key dependencies of LoS on distance, heights, and machine configurations.

**Dataset Generation** The same industrial environment described in 4.1.1.1 has been considered in this work, maintaining identical geometrical and structural characteristics. The analysis builds upon the previously defined parameters and layout configuration, ensuring consistency across studies. Based on the feature analysis discussed earlier, height-related parameters ($MH$, $h_{AP}$, and $h_{UE}$) were identified as the most influential for LoS probability, while geometric factors ($MS$, $SP$, and $MD$) play a secondary role in determining the visibility conditions within the industrial environment.

**Proposed Formula** Since the relative height of APs and UEs with respect to the industrial clutter level is likely to be more important than the absolute height values for the occurrence of LoS/NLoS conditions, MH, $h_{AP}$ and $h_{UE}$ have been replaced by $\Delta H_1 = (h_{AP} - MH)$ and $\Delta H_2 = (MH - h_{UE})$.

$$\Delta H_1 = h_{\mathrm{AP}} - MH, \qquad \Delta H_2 = MH - h_{\mathrm{UE}},$$

In agreement with common sense, the heatmap in Fig. 4.6 clearly shows that the LoS probability increases at larger $\Delta H_1$ and lower $\Delta H_2$, whereas it reduces otherwise. Furthermore, it is, of course, equal to 1 when the AP and the UE are standing above the machine layer, i.e. $\Delta H_1 \geq 0$ and $\Delta H_2 \leq 0$. The analysis has been then focused on nontrivial cases.

In the end, this study is aimed at looking for a simple, analytical formulation of the LoS probability in an industrial environment (LoSP in the following) as a function of $\Delta H_1$, $\Delta H_2$, and distance $d$.

The empirical evidence in Fig. 4.7 supports an exponential decrease in LoSP with distance, which is also consistent with previous investigations. The decay rate coefficient should be clearly tailored to the specific case. This leads to the following expression:

$$\mathrm{LoSP}(\Delta H_1, \Delta H_2, d) = \begin{cases} \exp\Big( - K(\Delta H_1, \Delta H_2)\, d\Big), & \Delta H_1 \leq 0 \text{ or } \Delta H_2 \geq 0, \\ 1, & \text{otherwise}, \end{cases} \tag{4.2}$$

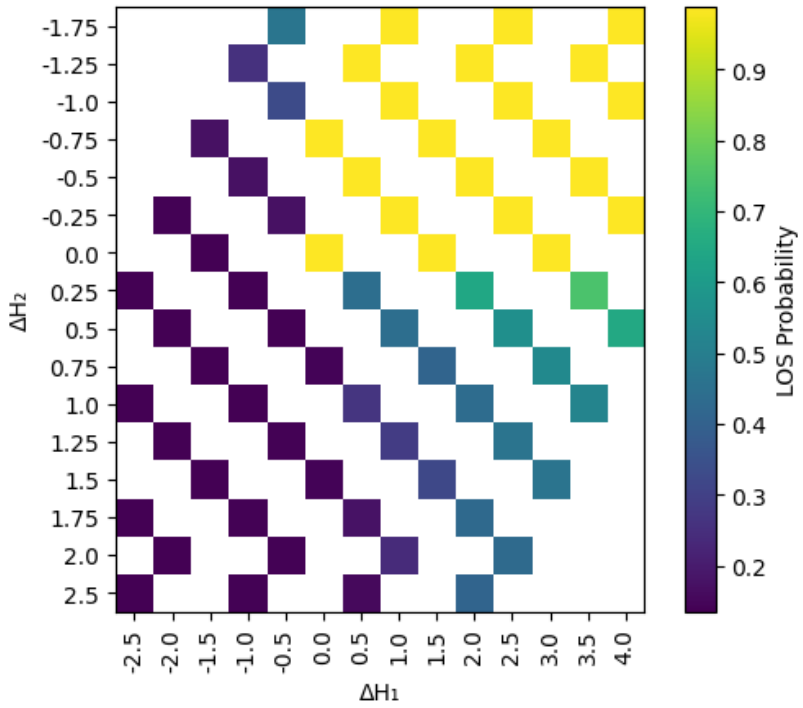



**Figure 4.6:** Heatmap.

where the decay rate $K(\cdot)$ is not constant but depends on $(\Delta H_1, \Delta H_2)$.

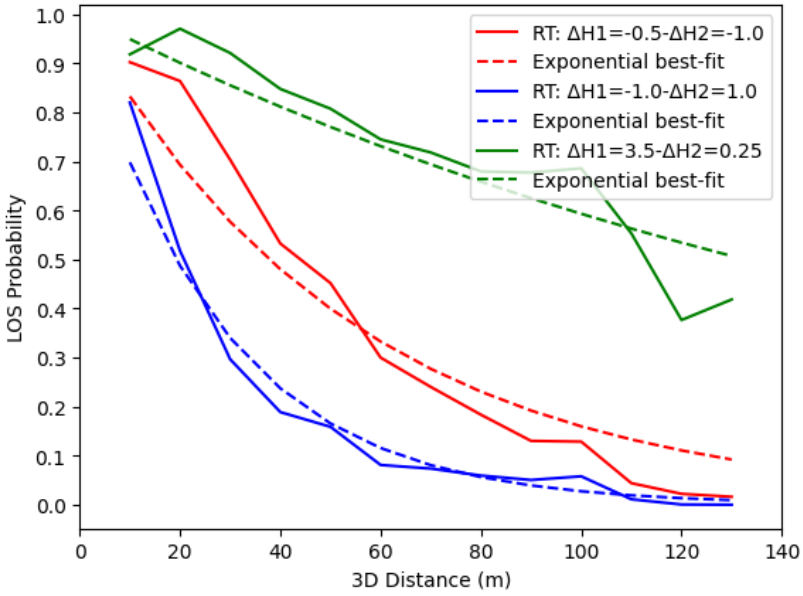



**Figure 4.7:** Los probability as exponential decay.

To discern the relationship between K, $\Delta H_1$, and $\Delta H_2$, Fig. 4.8 shows the best-fit values of the K parameter computed for different $\Delta H_1$ and $\Delta H_2$. The observations suggest a sigmoid-like relationship exists between the optimal K and $\Delta H_1$ values.

In the framework of the sigmoid functions, the following expressions have been in

particular considered:

$$K(\Delta H_1, \Delta H_2) = \frac{0.03}{1 + k_1(\Delta H_2)\, e^{\Delta H_1/2} + 0.007}, \tag{4.3}$$

with $k_1(\Delta H_2)$ modeled as an exponential decay:

$$k_1(\Delta H_2) = e^{-1.38\,\Delta H_2}. \tag{4.4}$$

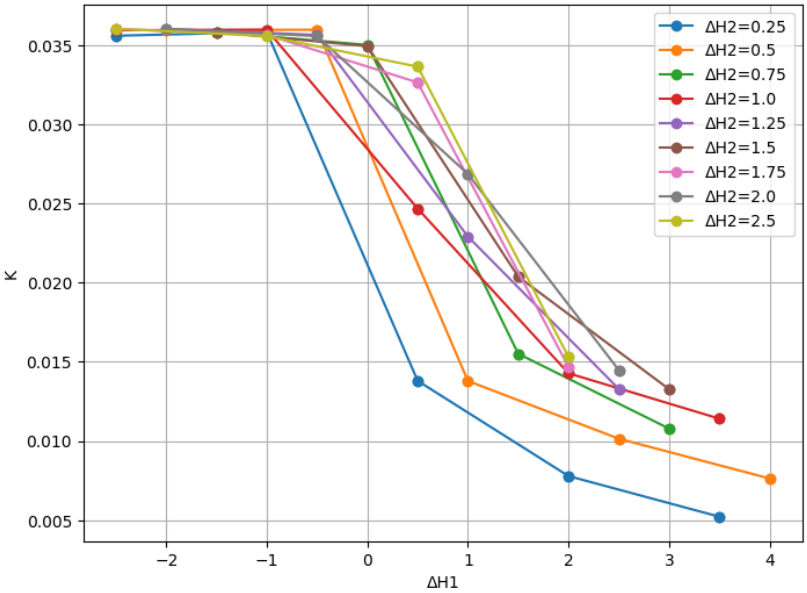



**Figure 4.8:** K vs. $\Delta H_1$ for different $\Delta H_2$ values

To model the dependence of the coefficient $k_1$ on $\Delta H_2$, the optimal values of $k_1$, i.e. corresponding to the optimal K values in Eq. 4.9, have been first computed for different $\Delta H_2$, and finally plotted against $\Delta H_2$ (Fig. 4.9, continuous line). Overall, $k_1$ exhibits a clearly decreasing trend with $\Delta H_2$, which was then modeled as a further exponential decay function, i.e. $k_1 = e^{-\gamma \cdot \Delta H_2}$. The best value of the decay rate coefficient was found through the least square method and found to be $\gamma = 1.38$ (Fig. 4.9, dashed line).

$$K(\Delta H_1, \Delta H_2) = \frac{0.03}{1 + e^{-1.38\,\Delta H_2}\, e^{\Delta H_1/2} + 0.007}. \tag{4.5}$$

The ultimate formulation of the K coefficient established as follows:

$$K = \frac{0.03}{1 + e^{-1.38\Delta H_2} \cdot e^{\Delta H_1{}^2}} + 0.007 \tag{4.6}$$

**Validation** RT simulations have been repeated for $(\Delta H_1, \Delta H_2) \in \{(-0.75, -2), (4, 0.5)\}$, i.e. in conditions not considered for deriving the LoSP function. The LoSP values extracted from the new simulations have been then compared with the new proposed

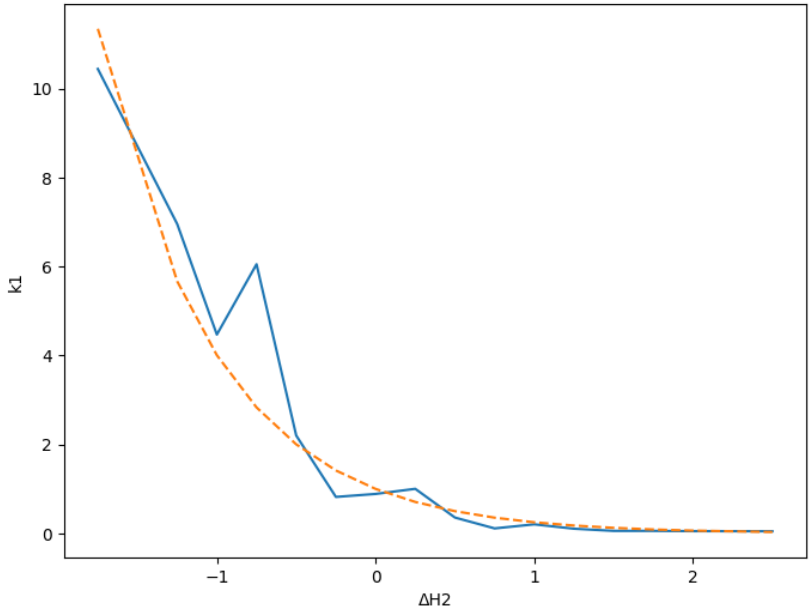



**Figure 4.9:** $k_1$ Vs. $\Delta H_2$

formula. The comparison has also been extended to the models included in the ITU and ETSI reports [17, 105]. The accuracy of each model is reported in Table 4.2 through its Root Mean Square Error (RMSE) with respect to the ground truth represented by the raw LoS probability provided by RT simulations.

| Model | RMSE |
|---|---|
| Proposed formula | 0.076 |
| ETSI | 0.254 |
| ITU | 0.304 |

**Table 4.2:** Comparison of model accuracy against RT ground truth.

It is worth pointing out that the LoS probability formula suggested by ETSI takes into consideration also machines' size and density, besides the heights of devices and machinery. Despite this greater flexibility, the RMSE of the model is greater compared to the new formula proposed in this work.

The reason is explained in Figs. 4.10 and 4.11. Although the ETSI model turns out to be basically as reliable as the model described by eq. 4.2 when limited machine density and larger machine size are considered (Fig. 4.10), its performance clearly drops when it comes to a greater density of the industrial clutter with a smaller dimension of machines (Fig. 4.11). Conversely, the new model exhibits an overall fair accuracy in both cases. The legend in Figs. 4.10 and 4.11 include the RMSE of the models concerning the RT ground truth for the sake of comparison. Figs 4.10 and 4.11 also explain why the ITU model has the largest RMSE. In fact, it consists of a very simple expression that can't be tuned to the different $\Delta H_1$ and $\Delta H_2$ values as it includes the link distance only.

On the whole, the inclusion in the LoSP model of extra parameters (like MD and MS), which are not of primary importance in LoS evaluation, may result in an overall worsening of the prediction accuracy.

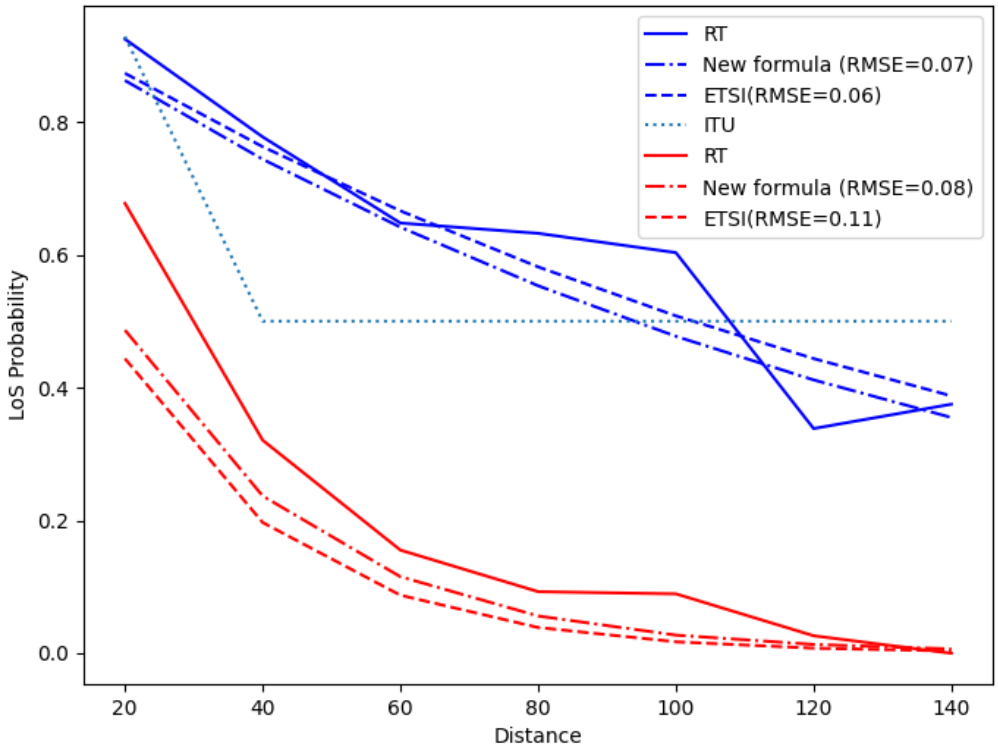



**Figure 4.10:** Models comparison for $MD = 0.15$, $MS = 4$. Blue and red curves correspond to ($\Delta H_1 = 2.5$ m, $\Delta H_2 = 0.5$ m) and ($\Delta H_1 = -1$ m, $\Delta H_2 = 1$ m), respectively.

#### 4.1.1.3 Site-Specific LoS Detection in Urban Environment

This study builds upon the work presented in "Site-Specific Machine Learning Approach for Line-of-Sight Detection" [108], where a GBDT classifier was trained on RT datasets from multiple urban environments, including Bologna, Munich, and San Francisco. The study demonstrated that tabular, geometry-based features—such as Tx and Rx coordinates, height differences, and building dimensions—can effectively describe the propagation environment without requiring full map data. Results showed strong classification performance across different cities, confirming that GBDT models can learn physically meaningful decision boundaries for LoS detection in complex urban scenarios.

**Dataset Generation** The study focused on predicting LoS conditions in real urban layouts. RT simulations were carried out on three representative cities: Bologna, Mu-

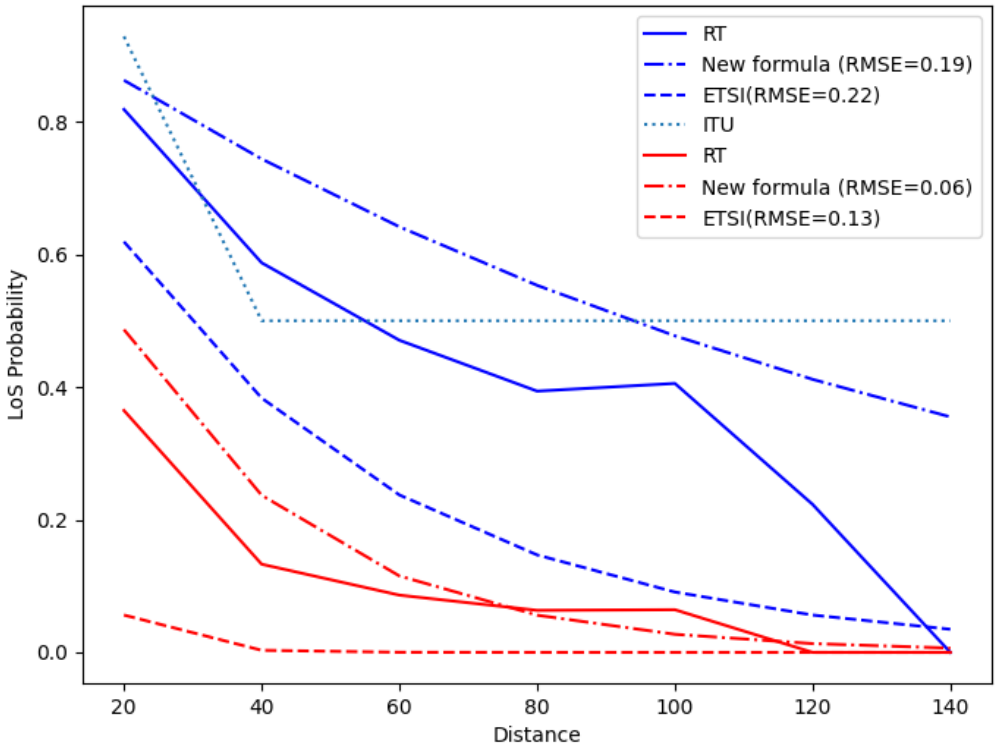



**Figure 4.11:** Models comparison for $MD = 0.25$, $MS = 2$. Blue and red curves correspond to ($\Delta H_1 = 2.5$ m, $\Delta H_2 = 0.5$ m) and ($\Delta H_1 = -1$ m, $\Delta H_2 = 1$ m), respectively.

nich, and San Francisco. For each city, a detailed digital map containing building footprints and heights was used. Txs were placed on the rooftops of the tallest buildings, while Rxs were distributed along the streets.

The characteristics of the three city maps used in this study are summarized in Table 4.3.

**Table 4.3:** Geometrical characteristics of the three considered urban maps.

| Info | Bologna | Munich | San Francisco |
|---|---|---|---|
| Number of buildings | 9377 | 2000 | 18031 |
| Average building height [m] | 12.98 | 16.95 | 19.75 |
| STD of building height [m] | 6.74 | 8.22 | 21.48 |
| Average building length [m] | 11.10 | 32.35 | 12.23 |
| Total area [$km^2$] | 6.63 | 8.13 | 10.20 |
| Buildings area [$km^2$] | 1.32 | 3.07 | 4.03 |
| Percentage of area [%] | 20 | 38 | 40 |

Each Tx–Rx link was labeled as LoS or NLoS depending on the RT output. This

resulted in a large site-specific dataset for each city. The datasets were split into training (80%), validation (10%), and test (10%) sets.

**Features and Learning Model** The ML model relied on a small but informative set of geometrical features:

- 3D Tx–Rx distance,
- Tx coordinates $(x_{Tx}, y_{Tx}, z_{Tx})$,
- Rx coordinates $(x_{Rx}, y_{Rx})$, The receiver height is fixed and equal to 1.5 m for all Rx locations.
- Building size below the Tx (length and height),
- Tx elevation above the rooftop ($\Delta$).

GBDT were selected as the baseline classifier, optimized with logistic loss. Evaluation metrics included the Area Under the AUPRC and F1-score, due to the strong class imbalance between LoS and NLoS.

**Results and Discussion** The datasets were highly imbalanced, with the LoS/NLoS ratio as low as 0.02–0.03 in Bologna and San Francisco. Despite this, the GBDT model demonstrated strong classification capabilities. Table 4.4 summarizes the performance across the three cities.

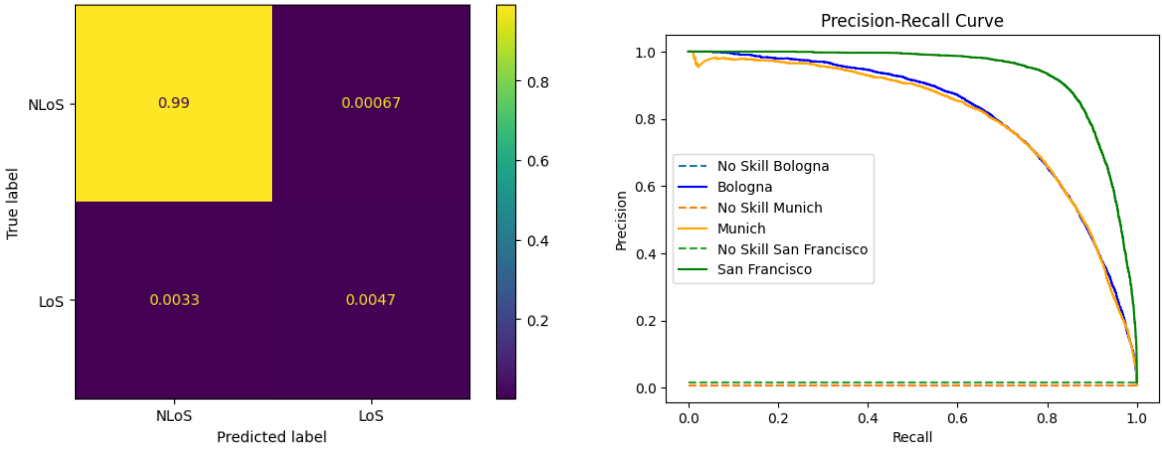
**Table 4.4:** Classification performance of GBDT model in urban scenarios.

| City | AUPRC | F1-score |
|---|---|---|
| Bologna | 81% | 74% |
| Munich | 80% | 74% |
| San Francisco | 93% | 87% |

Fig. 4.12a shows the confusion matrix for Bologna, where the imbalance is evident: NLoS links dominate, but the model still manages to detect LoS cases with reasonable accuracy. Precision–recall curves (Fig. 4.12b) further illustrate the superiority of the proposed ML method compared to a random classifier.

**LoS Probability Evaluation** In addition to binary classification, the model was used to estimate LoS probability as a function of Tx–Rx distance. Figs. 4.13a, b, c compare ML-based predictions with the WINNER model [18]. Unlike the generalized WINNER approach, the ML model adapts to each specific city, closely matching the ground truth from RT simulations.

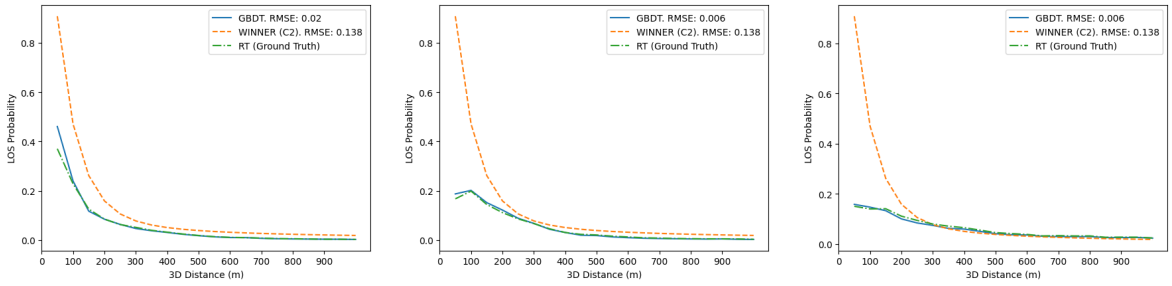
**Feature Importance** Explainable AI analysis revealed that Tx–Rx distance and Tx elevation above rooftop ($\Delta$) were the most influential features, followed by building dimensions. Figures 4.14a and 4.14b illustrate their effect on LoS probability: higher transmitter elevation and greater TxH significantly increase LoS likelihood.




(a) Normalized confusion matrix for Bologna city dataset.

(b) Precision–recall curves for Bologna, Munich, and San Francisco.

**Figure 4.12:** Performance evaluation of ML-based LoS detection in urban environments.



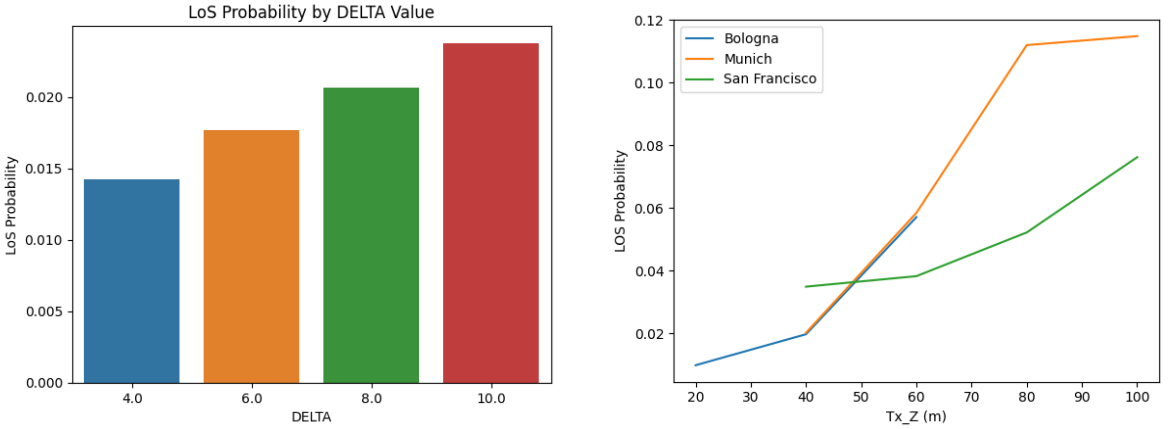

(a) Bologna (b) Munich (c) San Francisco

**Figure 4.13:** LoS probability vs distance: comparison between ML model, RT ground truth, and WINNER in three cities.

### 4.1.2 Image-based LoS Detection in Urban Environment

This section is based on the work presented in "Deep Learning Approach to Line of Sight Detection in Urban Environments" [109], where a U-Net-based DL framework was developed to predict LoS coverage maps directly from urban city maps. The study leveraged RT simulations for ground-truth generation and demonstrated that convolutional architectures can effectively capture spatial dependencies, providing accurate and scalable solutions for LoS prediction in complex urban scenarios.

**Dataset Generation** The first step in the proposed workflow is the creation of input maps that accurately represent the physical structure of urban environments, including buildings, streets, and open spaces. For this study, a high-resolution map of the urban center of Bologna city (Italy) (Fig. 4.15) was randomly divided into 1000 smaller tiles, each measuring 150m × 150m, to generate manageable input samples for




(a) LoS probability vs Δ for Bologna. (b) LoS probability vs transmitter height.

**Figure 4.14:** Feature importance analysis: impact of (a) Δ and (b) transmitter height on LoS probability.

the ML model. A threshold of 25 buildings per map was set to ensure a diverse and representative dataset.

**Figure 4.15:** Bologna City Center.

To establish the ground truth for model training, RT simulations were performed to generate LoS coverage maps. The RT tool processes each urban tile and produces output maps detailing LoS availability, effectively capturing realistic RF propagation conditions.

The resulting dataset comprises paired images: one set representing the input city maps and the other depicting the corresponding LoS coverage maps derived from RT simulations. These image pairs form the foundation for supervised DL, enabling the model to learn spatial patterns and predict LoS coverage in complex urban scenarios with high accuracy.

**Learning Model** Unlike tabular approaches, which require handcrafted features explicitly extracted from the environment (e.g., distances, Tx and Rx heights, material parameters, or geometric descriptors), the image-based method directly leverages spatial structures. A U-Net architecture (shown in Fig. 4.16) was employed, consisting of four downsampling and four upsampling blocks with skip connections, enabling pixel-level LoS segmentation. The input was a two-channel image (buildings + Tx), while the output was a single-channel probability map.

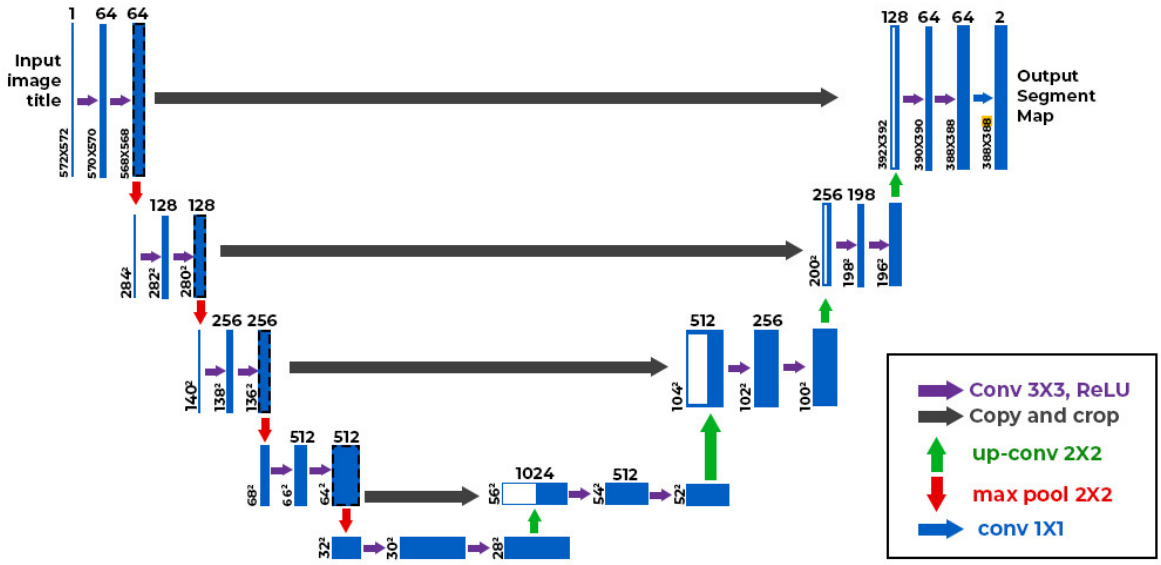



**Figure 4.16:** U-Net architecture used for LoS prediction. Encoder–decoder structure with skip connections ensures both global context and fine localization.

The implemented U-Net consists of four downsampling and four upsampling blocks, with each block composed of two convolutional layers using $3 \times 3$ kernels followed by ReLU activations. Downsampling is performed via $2 \times 2$ max-pooling, while upsampling is achieved through transposed convolutions with the same kernel size. Skip connections are used to preserve spatial details between encoder and decoder stages. The network accepts a two-channel input (representing the building map and transmitter location) and outputs a single-channel LoS probability map. The final layer is a $1 \times 1$ convolution without activation, producing a pixel-wise prediction. Table 4.5 summarizes the architectural specifications.

Since the task involves binary classification at the pixel level, an appropriate loss function is essential for robust learning. BCE is widely used for segmentation tasks, but in cases of class imbalance, where LoS pixels may be less frequent than NLoS pixels, a Dice Loss or a combination of Dice Loss and BCE can be more effective. Dice Loss emphasizes overlap between predicted and ground-truth LoS regions [110], ensuring better segmentation performance in unbalanced datasets.

$$L_{BCE}(y, \hat{y}) = -[y \cdot \log \hat{y} + (1 - y) \cdot \log (1 - \hat{y})] \tag{4.7}$$

**Table 4.5:** U-Net architecture details used in this study.

| Component | Specification |
|---|---|
| Input Channels | 2 (Tx + Buildings) |
| Output Channels | 1 (LoS map) |
| # Downsampling Blocks | 4 |
| # Upsampling Blocks | 4 |
| Conv. Layers per Block | 2 |
| Total Conv. Layers | 18 |
| Conv. Filter Size | $3 \times 3$ |
| Pooling Layers | 4 (MaxPool, $2 \times 2$) |
| Upsampling Layers | 4 (Transposed Conv, $2 \times 2$) |
| Activation Function | ReLU |
| Final Layer | $1 \times 1$ Conv (No Activation) |
| Skip Connections | Yes (Encoder–Decoder) |

where y is the true binary label (0 or 1), and $\hat{y}$ is the predicted probability of the positive class. For a dataset with N samples, the average BCE loss is:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i)] \tag{4.8}$$

$$DICE(y, \hat{y}) = \frac{2\sum_{i=1}^{N} y_i \hat{y}_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i} \tag{4.9}$$

The loss function used in this work is defined as follows:

$$L_{tot} = \alpha \cdot L_{BCE} + (1 - \alpha) \cdot L_{DICE} \tag{4.10}$$

where $\alpha$ balances the contribution of each term, ensuring stable optimization and improved segmentation quality.

**Results and Discussion** The training process was performed over 50 epochs using the combined BCE and Dice loss. Fig. 4.17 shows the training loss and test F1-score over epochs. As seen in the figure, the loss consistently decreases while the F1-score steadily increases and plateaus, indicating good convergence and generalization of the model.

Following the confusion matrix analysis (Fig. 4.18), a quantitative evaluation of the model's performance is provided using standard classification metrics. The model achieves an accuracy of 90%, indicating the overall correctness of its predictions. Precision and recall are reported as 95% and 85% , respectively, highlighting the model's effectiveness in identifying true LoS pixels while minimizing false positives and false

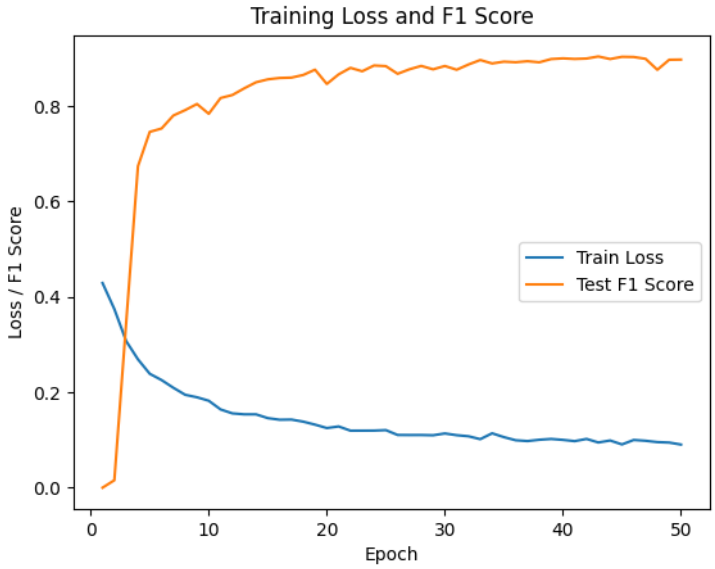



**Figure 4.17:** Training process: convergence of loss and stabilization of test F1-score over epochs.

negatives. The F1-score, which balances precision and recall, is 90%, confirming the robustness of the model in pixel-wise classification tasks.

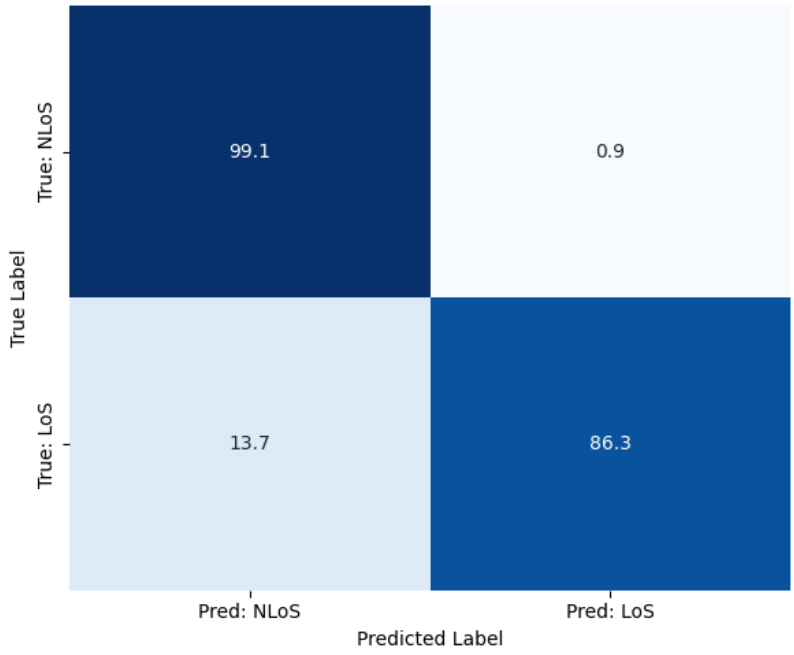



**Figure 4.18:** Normalized confusion matrix for Bologna dataset.

These metrics validate the reliability of the DL approach in detecting LoS regions from urban map inputs. Furthermore, the visual comparison in Fig. 4.19 between the input map and the corresponding predicted LoS coverage showcases the model's ability to generate spatially consistent and realistic predictions. These findings demonstrate

that image-based DL techniques can successfully model complex propagation scenarios in dense urban areas and potentially serve as real-time tools for LoS estimation in next-generation communication systems.

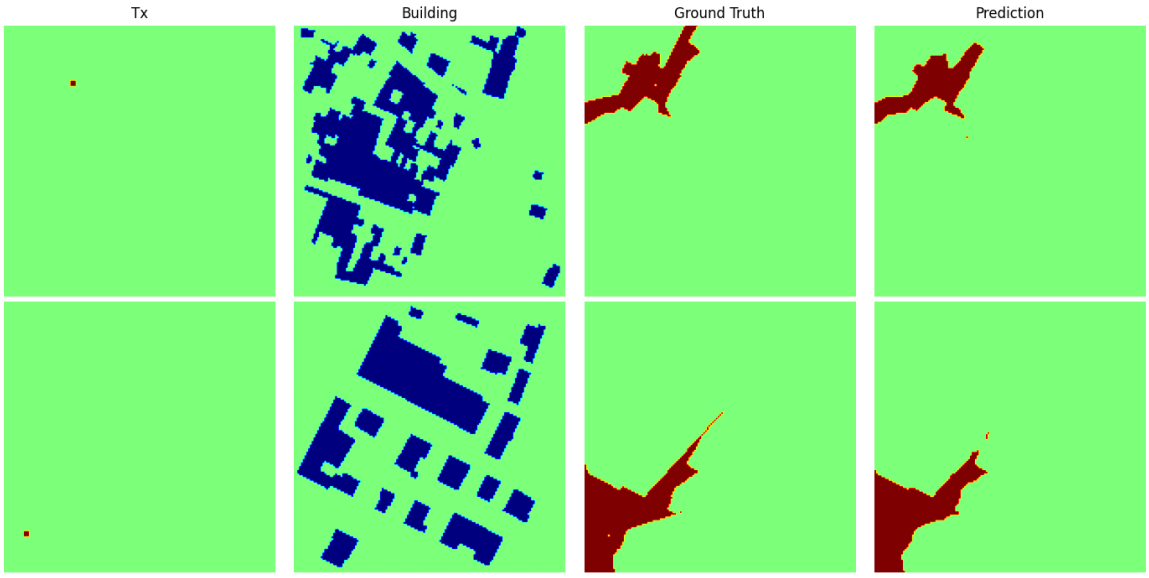



**Figure 4.19:** Qualitative comparison: input building map with Tx, RT-generated LoS coverage (ground truth), and U-Net prediction.

### 4.1.3 Comparison Between Tabular and Image-Based LoS Detection in Urban Environments

A direct comparison between the tabular-data and image-based approaches is meaningful for the site-specific LoS detection task in urban environments, where both methods were evaluated on the Bologna city scenario.

The tabular approach relies on explicitly extracted geometric and link-related features (e.g., distances, heights, and relative positions) and employs GBDT for classification. While this method achieves satisfactory performance with relatively low computational complexity and good interpretability, its accuracy—measured in terms of F1-score, precision–recall metrics, and AUPRC—depends strongly on the quality and completeness of the handcrafted features. In contrast, the image-based approach operates directly on spatial maps encoding the urban layout and transmitter location, allowing CNN model(e.g. U-Net) to implicitly learn spatial obstructions and propagation patterns. This results in consistently higher classification performance, with improved F1-scores and precision–recall characteristics compared to the tabular approach, particularly in challenging urban regions where spatial context plays a dominant role.

Although the image-based method requires higher computational resources and larger training datasets, it provides a more holistic and robust representation of the environment, effectively capturing spatial correlations that are difficult to encode explicitly in tabular features.

Overall, the comparison highlights a trade-off between model simplicity and feature engineering in tabular methods, and superior predictive accuracy and spatial awareness in image-based approaches, making the latter particularly suitable for large-scale, site-specific LoS prediction in complex urban environments.

## 4.2 Narrowband Channel Modeling

### 4.2.1 Channel Parameter Estimation (PLE, Shadowing, and Fast Fading)

Narrowband propagation modeling typically involves the estimation of three fundamental parameters that describe the main components of the received signal: the PLE, accounting for the average power decay with distance; the shadowing standard deviation ($\sigma$), capturing large-scale variations due to obstacles and environmental clutter; and the Rice factor ($K$), representing the small-scale fading effects caused by multipath propagation.

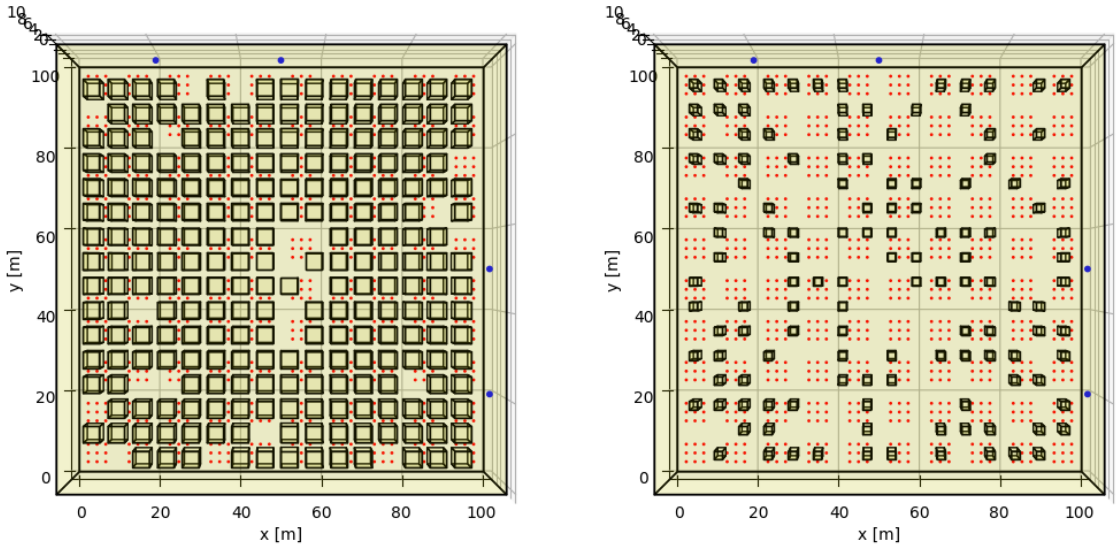
This section combines two complementary studies. The first, presented in "A Machine Learning Approach to Wireless Propagation Modeling in Industrial Environment" [111], focuses on the estimation of PLE and $\sigma$ through a hybrid RT–ML approach. The second, detailed in "Fast-Fading Modeling in Wireless Industrial Communications" [112] extends the investigation to the characterization and prediction of the Rice factor under the same industrial conditions. Together, they provide a unified framework for narrowband channel parameter estimation in industrial environments.

#### 4.2.1.1 Industrial Environment Representation

The same industrial environment is considered for the estimation of the PLE, shadowing standard deviation ($\sigma$), and Rice factor ($K$), ensuring consistent modeling conditions across all channel parameters. The simulated environment reproduces a typical industrial shed with dimensions of 100 m $\times$ 100 m and a uniform height of 10 m, populated by metallic machinery represented as rectangular boxes of equal height and variable size. Machines are arranged in a grid-like structure with adjustable spacing between them to emulate different density conditions, as commonly found in manufacturing halls and automated facilities.

Initially, a perfectly regular machine layout with constant spacing ($SP$) is defined. To enhance realism, a fraction $T$ of machines is then randomly removed, introducing irregularity and local variations in the layout. This approach allows the representation of both highly cluttered and more open industrial scenarios. Examples of two extreme cases are illustrated in Fig. 4.20, corresponding respectively to high and low machine densities.

Unlike the LoS analysis, which relies solely on geometric visibility and can be computed very quickly, the estimation of channel parameters such as PL, shadowing, and




(a) High Density (MS = 4, SP = 2, T = 0.1, MD = 0.4).

(b) Low Density (MS = 2, SP = 4, T = 0.5, MD = 0.05).

**Figure 4.20:** Two extreme cases of industrial maps. Blue and red points represent the Txs and Rxs, respectively.

fast fading statistics requires full EM simulations. These simulations account for multiple propagation mechanisms (e.g., reflections, diffractions, and scattering) and are therefore considerably more time-consuming. For this reason, the number of parameter values and combinations has been reduced here to keep the computational effort within reasonable limits while still ensuring a representative coverage of the industrial scenario.

Multipath propagation in this industrial layout is investigated for different values of machine size ($MS$ = 2, 3, 4, 8 m), spacing between machines ($SP$ = 2, 3, 4 m), and fraction of removed machines ($T$ = 0.1, 0.2, 0.35, 0.5). A total of 48 distinct configurations is obtained from all parameter combinations, and for each case five realizations are generated by randomizing the removed machines, resulting in 240 total maps.

Txs are placed along one wall of the shed at a height of 3 m (slightly higher than $MH$), while Rxs are distributed along the aisles at 1 m height, representing automated guided vehicles (AGVs) or mobile robots. Four Tx positions and multiple Rx points per layout are considered, allowing extensive coverage of link distances and geometric configurations.

Four operating frequencies are analyzed through RT simulations—700 MHz, 3.5 GHz, 28 GHz, and 60 GHz—covering a wide range of industrial wireless applications. Metal dominates the environment due to the machinery, beams, shelves, and pipes, leading to rich multipath propagation characterized by strong reflections and occasional diffractions on metallic edges [113–118].

Because of multipath interference, the envelope of the received signal—i.e., the slowly varying amplitude obtained after removing the carrier oscillations—exhibits rapid spatial fluctuations (small-scale fading), whereas blockage and shadowing effects cause slower variations (large-scale fading) [1]. As a result, the received signal strength decreases with distance only on average, according to:

$$PL(d) = PL(d_0) + 10 \cdot n \cdot \log_{10}\left(\frac{d}{d_0}\right) + \chi, \tag{4.11}$$

where $n$ is the propagation PLE, $PL(d_0)$ the PL at the reference distance $d_0$ (commonly 1 m), and $\chi$ a random variable describing large-scale variations. The reference loss is computed as:

$$PL(d_0) = 20 \cdot \log_{10}\left(\frac{4\pi}{\lambda}\right), \tag{4.12}$$

and the PLE can be estimated from a set of measured or simulated links as:

$$PLE = \frac{\sum_{i=1}^{N}(PL_i - PL(d_0)) \cdot \log_{10}(d_i)}{10\sum_{i=1}^{N}(\log_{10}(d_i))^2}, \tag{4.13}$$

with $N$ being the number of Tx–Rx links, $PL_i$ denotes the PL measured or simulated for the $i$-th Tx-Rx link at distance $d_i$. To remove the influence of small-scale fading, path loss values are spatially averaged over $3 \times 3$ grids of RX points, ensuring that $\chi$ represents only large-scale fluctuations and follows a normal distribution, $\chi \sim \mathcal{N}(0, \sigma)$ [1, 114].

Eq. 4.13 is used to compute the PLE values reported in Fig. 4.22, where it is applied to the RT-based PL samples forming the dataset.

#### 4.2.1.2 Dataset Generation

RT simulations are performed using the tool developed at the University of Bologna [104], which computes electromagnetic interactions between transmitters, receivers, and obstacles within the 3D industrial model. Table 4.6 summarizes the main simulation parameters.

Two main materials are defined: machinery is modeled as a Perfect Electric Conductor (PEC), while walls and floor are concrete with $\varepsilon_r = 5$, $\sigma = 0.01$ S/m at 0.7–3.5 GHz and $\sigma = 0.1$ S/m at 28–60 GHz.

A total of 960 RT simulations (4 frequencies $\times$ 240 maps) are performed. Each simulation includes four Txs and several hundred Rxs, producing thousands of Tx–Rx links. Rxs are grouped in $3 \times 3$ grids spaced 10 m apart, with spacing of $10\lambda$ to ensure statistically independent fading. Fast-fading effects are filtered out for large-scale modeling by averaging over each grid.

**Table 4.6:** Simulation Configuration

| Parameter Name | Values | |
|---|---|---|
| Frequency | 0.7, 3.5, 28, 60 GHz | |
| Antenna Type | Omnidirectional | |
| Propagation Mechanisms | Direct Ray | On |
| | Interactions | Up to 4 |
| | Reflections | Up to 3 |
| | Diffractions | Up to 1 |
| | Scattering | Off |
| | Transmission | Off |

The resulting PL values are plotted versus link distance (Fig. 4.21) and used to extract $n$ and $\sigma_{dB}$ for each feature combination, yielding 192 unique configurations.

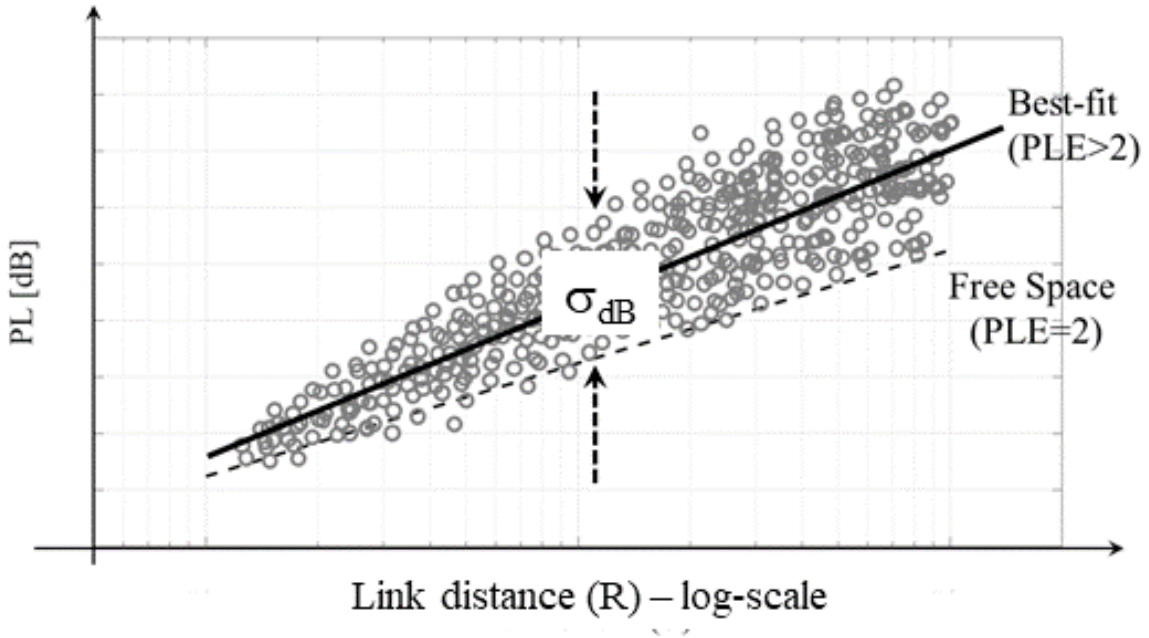



**Figure 4.21:** Typical dependence of path loss on link distance.

The corresponding dataset structure is sketched in Fig. 4.22, where each row contains the input features (MS, SP, MD, and $f$) and the target outputs ($n$ and $\sigma_{dB}$).

The Rice factor $K$, instead, is derived from a separate fast-fading analysis based on the same set of RT simulations but processed independently.

Because small-scale fading is irregular, fast-fading statistical properties can be investigated by collecting values of $\rho$ (see Eq. 1.5) over the service area. In this study, this data collection is based on the outcomes of RT simulations. Based on the samples gathered for each industrial scenario at each different frequency, a PDF can be easily achieved (Fig. 4.23, continuous line). Many previous investigations have proved that the Rice distribution is often the best-fit choice for the statistical representation of the

| | MS[m] | SP[m] | MD | freq[GHz] | $PLE_{nlos}$ | $PLE_{los}$ | PLE | $\sigma[dB]_{nlos}$ | $\sigma[dB]_{los}$ | σ[dB] | $DS_{mean}$[ns] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 2 | 0.23 | 0.7 | 2.46 | 1.82 | 2.30 | 5.51 | 3.58 | 7.01 | 33.52 |
| 1 | 2 | 2 | 0.20 | 0.7 | 2.43 | 1.85 | 2.29 | 5.31 | 3.97 | 6.68 | 36.48 |
| 2 | 2 | 2 | 0.16 | 0.7 | 2.33 | 1.88 | 2.23 | 4.39 | 3.63 | 5.40 | 42.88 |
| 3 | 2 | 2 | 0.12 | 0.7 | 2.27 | 1.92 | 2.19 | 4.11 | 3.63 | 4.75 | 50.90 |
| 4 | 2 | 3 | 0.14 | 0.7 | 2.30 | 1.91 | 2.18 | 4.22 | 3.36 | 5.03 | 42.59 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 187 | 8 | 3 | 0.26 | 60.0 | 2.81 | 2.03 | 2.68 | 10.06 | 3.82 | 10.64 | 36.61 |
| 188 | 8 | 4 | 0.40 | 60.0 | 2.81 | 2.09 | 2.72 | 11.27 | 3.69 | 11.32 | 25.41 |
| 189 | 8 | 4 | 0.36 | 60.0 | 2.80 | 2.03 | 2.67 | 10.57 | 3.18 | 10.89 | 29.73 |
| 190 | 8 | 4 | 0.29 | 60.0 | 2.73 | 2.05 | 2.60 | 10.35 | 4.22 | 10.55 | 31.73 |
| 191 | 8 | 4 | 0.22 | 60.0 | 2.68 | 2.01 | 2.47 | 9.46 | 4.12 | 9.87 | 39.85 |

**Figure 4.22:** Example of dataset with features (MS, SP, MD, frequency) and outputs (PLE, $\sigma$).

fast-fading fluctuations (Fig. 4.23, dashed line).

The procedure to compute the $K$ of the best-fit Rice distribution for each scenario (i.e. each combination of $MS$, $SP$, $T$, and $f$ is detailed as follows:

- *Fast-fading collection:* In each scenario, the received signal amplitude $V(P)$ is computed in every Rx location with respect to each Tx position. The corresponding small-area average $V_{SA}(P)$ is then achieved through spatial averaging over the corresponding 3x3 grid. Finally, the fast-fading contribution is computed according to eq. 1.5. In this way, approximately 13,000 samples of $\rho$ are collected for each scenario.
- *Empirical PDF Construction:* The PDF of the $\rho$ samples is extracted from the empirical data.
- *Rice Distribution Fitting:* The Rice distribution is fitted to the empirical PDF and the corresponding K value is recorded to fill the database necessary to train and test the ML model.
- *Dataset Compilation:* The final dataset is compiled, containing columns for the features of each scenario (MS, SP, MD, and f) and the corresponding output K.

Fig. 4.23 clearly shows the empirical PDF is usually really close to a Rice distribution, that also proves the outcomes of RT simulations are physically sound.

#### 4.2.1.3 Machine Learning Model

The dataset achieved from RT simulations was finally used to train and test an MLP network aimed at catching the relationship between the output labels (PLE, $\sigma_{dB}$, and $K$) and the corresponding features. The learning process was organized in three different steps:

1. Since the labels included in the final dataset look somehow linearly dependent

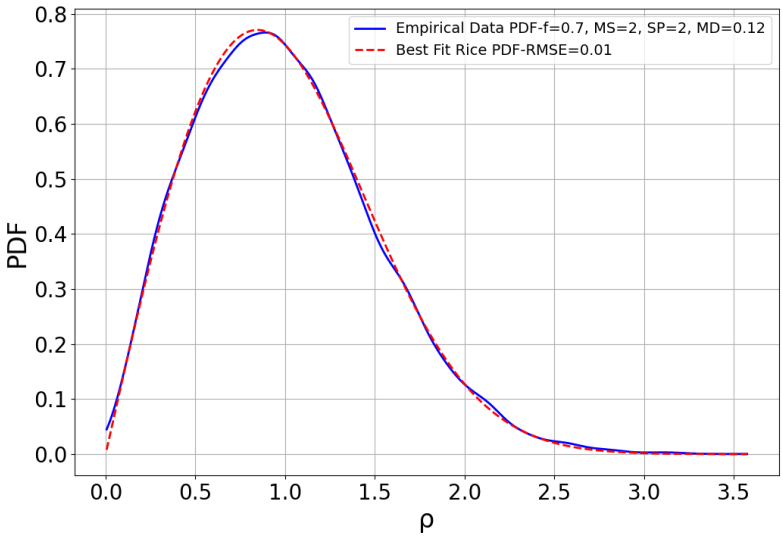



**Figure 4.23:** An example of best-fit Rice distribution.

on $log_{10}(f_{GHz})$ (Fig. 4.24), a simple linear regression was first carried out to compute the coefficients $\beta$ and $\gamma$ describing the average frequency dependence of the label as:

$$< y >= \beta \cdot log_{10}(f_{GHz}) + \gamma \tag{4.14}$$

where $y$ indifferently stands for PLE, $\sigma_{dB}$, and $K$, and $\langle\cdot\rangle$ represents the mean value.

2. For each $y_i$, $i = 1, 2, .., 192$, the residual $r_i$ concerning the regression line has been computed, i.e.:

$$r_i = y_i - \beta \cdot log_{10}(f_{GHz}) - \gamma, \text{ i=1, 2, ..., 192} \tag{4.15}$$

3. MLP has been leveraged to seize the dependence of the residuals on the geometrical features (MS, SP, and MD).

It is worth noting that, although the linear trend with frequency is shown in Fig. 4.24 for PLE and $\sigma_{dB}$, a similar dependency was also verified for the Rice factor $K$, and hence the same regression approach was applied for its modeling as well.

Two different MLP networks, respectively tailored to PLE and $\sigma_{dB}$, have been in the end arranged in agreement with the outlined procedure, and an additional one has been trained for the Rice factor $K$ regression.

The primary objective during the training stage of every ML model is to optimize the parameters, specifically the weights ($w$) and biases ($b$) of each layer in MLP, to achieve optimal learning. Following the training stage, the validation phase focuses on fine-tuning hyperparameters such as the number of hidden layers, the neurons within each hidden layer, and the activation function (represented as 'f' in Fig. 4.25).

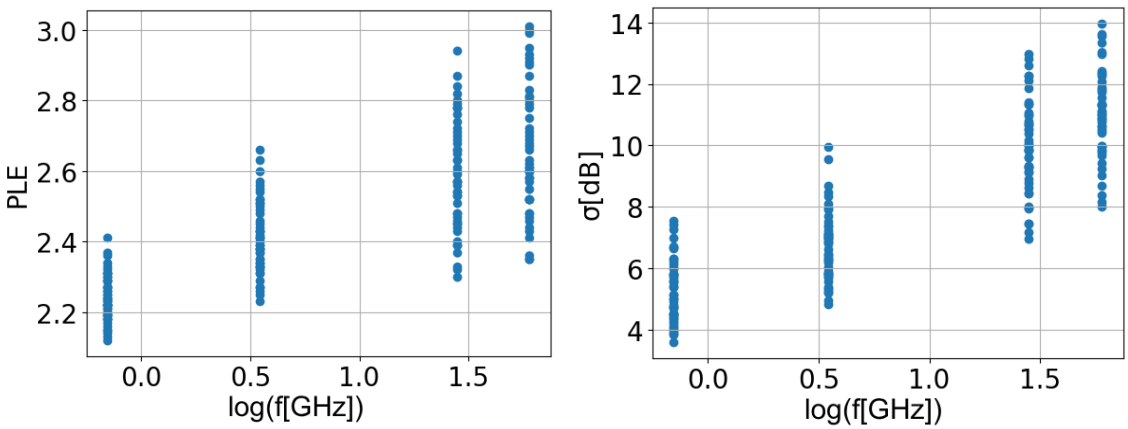



**Figure 4.24:** Linear Relationship between PLE and $\sigma_{dB}$ with log(f).

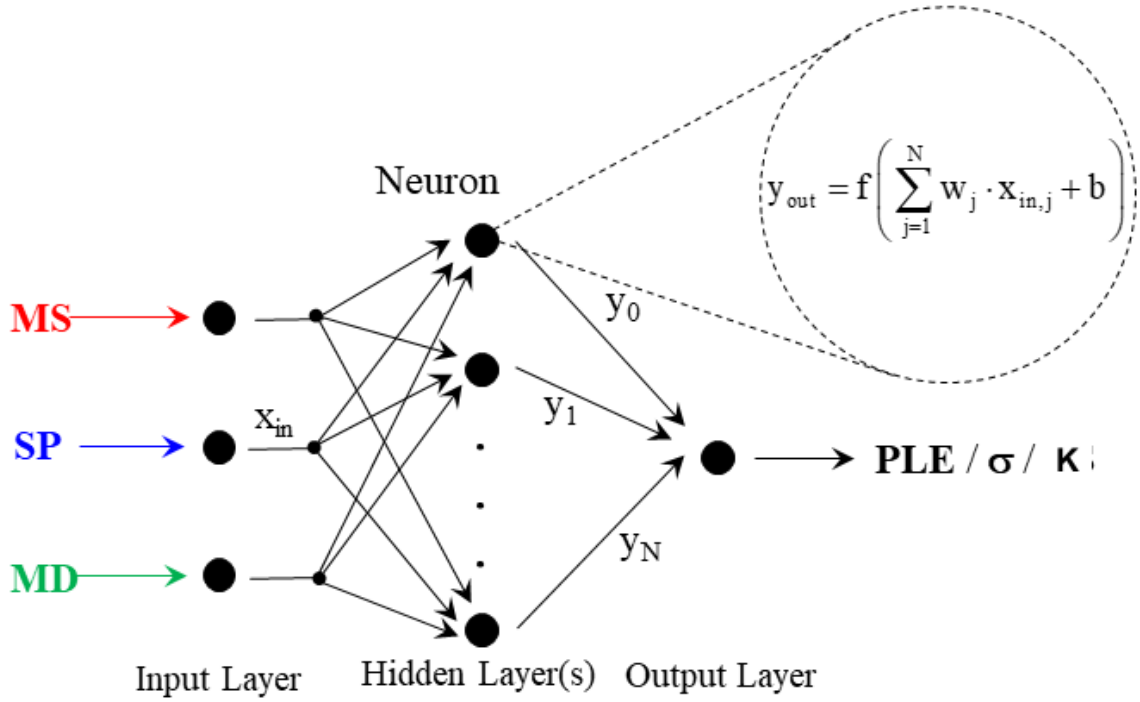



**Figure 4.25:** General structure of a Multi-Layer Perceptron. A single hidden layer is deployed for the sake of simplicity.

Table 4.7 provides a unified summary of the tuned hyperparameters for the three models designed for predicting PLE, $\sigma_{dB}$, and $K$. It is noteworthy that 'ReLU' stands for the Rectified Linear Unit function, 'tanh' indicates the hyperbolic tangent activation, 'lbfgs' is a quasi-Newton optimizer suitable for small datasets [119, 120], and 'Adam' is a stochastic gradient-based optimizer effective for slightly larger datasets.

#### 4.2.1.4 Results and Discussion

**PLE and $\sigma_{dB}$** The accuracy of the ML approach to PLE and $\sigma_{dB}$ prediction is displayed in Table 4.8 in terms of overall loss over the test dataset, together with the

**Table 4.7:** Major hyperparameters of the MLP networks for PLE, $\sigma_{dB}$, and $K$ regression.

| **Hyperparameter** | **PLE** | $\sigma_{dB}$ | $K$ |
|---|---|---|---|
| Hidden Layer Sizes | (3,) | (4,) | (8, 4) |
| Activation Function | tanh | ReLU | ReLU |
| Solver for Weight Optimization | lbfgs | lbfgs | Adam |
| Learning Rate | — | — | 0.001 |
| Maximum Iterations | 3000 | 3000 | 3000 |
| Early Stopping | False | False | True |

range of variability of each target label. As the loss turns out quite small compared to the range of variability for both parameters, the learning task has been fairly accomplished. Furthermore, a comparison between different models shows satisfactory overall effectiveness, with the MLP and XGBRegressor slightly outperforming the others.

**Table 4.8:** Model evaluation for PLE and $\sigma$ with minimum and maximum values.

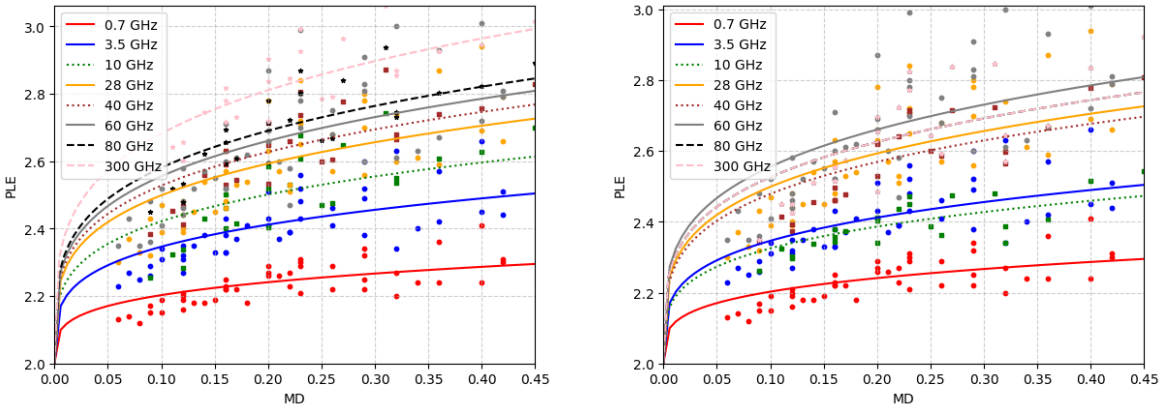
| | **RMSE of different models** | | | | **Min** | **Max** |
|---|---|---|---|---|---|---|
| | MLP | XGBR | SVR | RF | Value | Value |
| PLE | 0.049 | 0.051 | 0.105 | 0.056 | 2.12 | 3.01 |
| $\sigma$ [dB] | 0.630 | 0.699 | 1.242 | 0.751 | 3.61 | 13.96 |

To further investigate and evaluate the models' performances, a generalization perspective is adopted. This entails the evaluation of the models' consistency when challenged on new values of the features never seen during the training stage. To this aim, a fresh dataset with entirely new feature values was considered, as reported in Table 4.9.

**Table 4.9:** Example of new dataset configuration.

| **Feature** | **Values** |
|---|---|
| MS | 2.5, 3.5, 6 m |
| SP | 2.5, 3.5 m |
| Frequency | 10, 40, 80, 300 GHz |

Fig. 4.26 shows PLE against MD for different frequency values for both the MLP-based model (left) and XGBRegressor (right). Besides the data returned by RT simulations (dots), new data generated utilizing the trained ML tools related to the new dataset (squares and stars) are added to the figures, which also include the corresponding best-fit line. Results are in clear agreement with the RT outcome as far as the MLP-based model is concerned (Fig. 4.26(a)), thus further corroborating the effectiveness of the




(a) MLP-based model. (b) XGBRegressor.

**Figure 4.26:** PLE sensitivity to MD and frequency for MLP-based model (left) and XGBRegressor (right).

training but also highlighting a fair robustness in terms of generalization skill. Conversely, the XGBRegressor cannot consistently track the PLE sensitivity to completely new frequency values (Fig. 4.26(b)). The reason lies in the discrete representation of frequency in the training set, which limits the ability of tree-based structures to extrapolate beyond known split thresholds. In conclusion, the MLP-based model turns out as the most reliable and flexible overall and will be therefore referred to in the following.

Fig. 4.27 shows $\sigma$ against MD for different frequency values. Similar to the PLE model, they are in clear agreement with the RT outcome as far as the MLP-based model is concerned.

Prior research suggests that conventional ML models excel in interpolating within the known data range but often exhibit poor performance when extrapolating beyond this range [121]. However, the MLP-based model employed in this study is not in agreement with this trend, as it effectively learned the correlation between the target output and frequency, demonstrating a remarkable ability to predict accurately even for frequency values outside the frequency range explored inside the training dataset.

The trained MLP can therefore be exploited to quickly and reliably complement the limited amount of information that has been painstakingly gathered through electromagnetic simulations (or could be provided by channel experimental sounding). The availability of a large set of information helps to clearly understand the existing relationship between the output propagation labels and the considered input features. For instance, Figs. 4.26(a) and 4.27 clearly show that both PLE and $\sigma$ increase with machine density and frequency. Simple best-fit lines have been computed to formally describe the highlighted trends:

$$PLE = 2 + k_1 \cdot \sqrt[4]{MD \cdot f_{GHz}^{\frac{1}{k_2}}}, \tag{4.16}$$

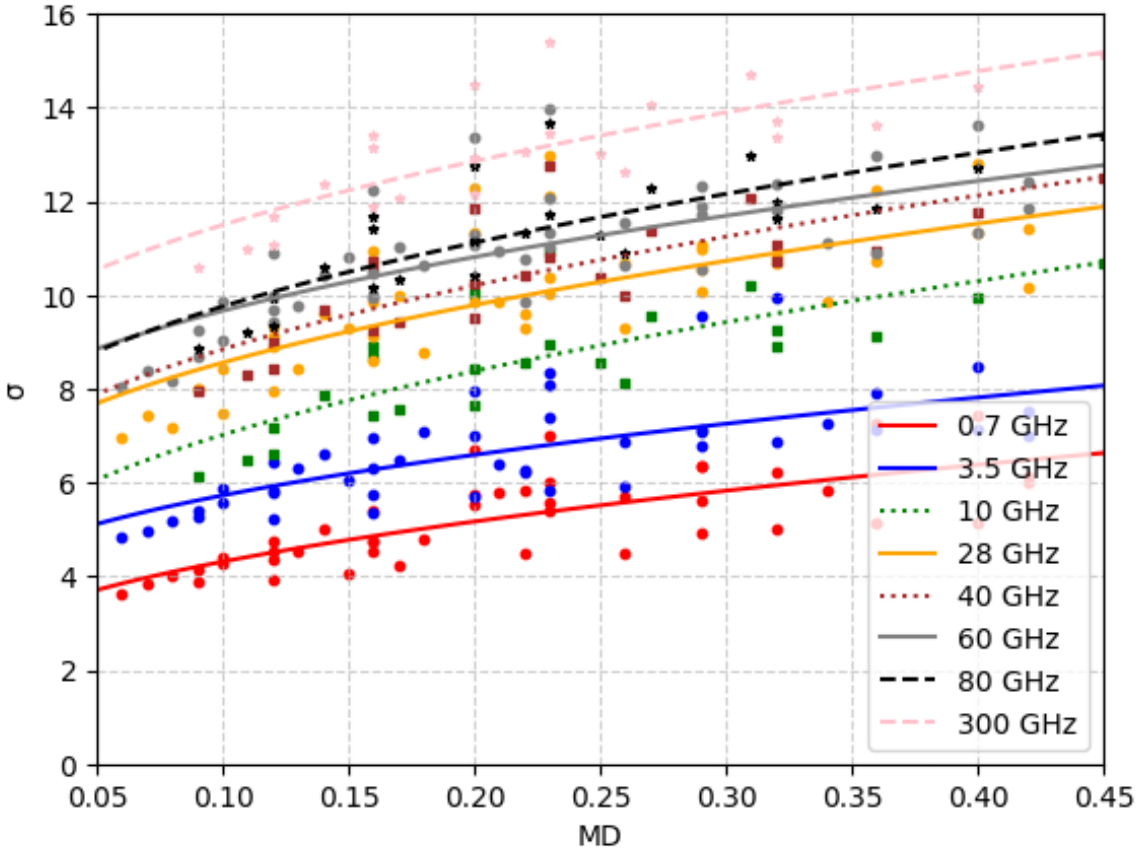



**Figure 4.27:** $\sigma$ sensitivity to MD and frequency for MLP-based model.

$$\sigma = k_3 \cdot \sqrt{MD} + k_4 \cdot \log_{10}(f_{GHz}) \tag{4.17}$$

The coefficients $k_1$–$k_4$ can be computed for each frequency according to the least square method, leading to the continuous and dashed lines in Figs. 4.26(a) and 4.27. To obtain a single, simple analytical model, the same coefficients were optimized across all frequencies, giving: $k_1$=0.48, $k_2$=1.5, $k_3$=10.27, and $k_4$=3.45.

**Comparison with Previous Studies** This sub-section discusses the reliability of the proposed ML-based approach to wireless propagation modeling in factories in comparison with the results reported in previous studies [72, 116–118, 122–127] and in the technical report [17]. The comparison takes into account measurements and simulations referring to different propagation conditions, including LoS, NLoS with light clutter, and NLoS with heavy clutter, as typically encountered in industrial environments characterized by varying machine density and obstruction levels.

Experimental and simulation assessments in literature show that in the presence of LoS, PLE is approximately equal to 2 regardless of the frequency (green crosses in Fig. 4.28), whereas in NLoS conditions it is increasingly greater as the NLoS level gets heavier (red dots and purple stars in Fig. 4.28). Also, PLE significantly increases with

frequency in the case of dense industrial clutter (purple stars in Fig. 4.28). These trends are actually in contrast with the path-loss models for indoor factories included in [17], where a frequency-independent PLE is assumed and the value for sparse clutter is greater than that for dense clutter. PLE values in [17] for industrial cases are only slightly greater than 2 even in NLoS, which seems unlikely.

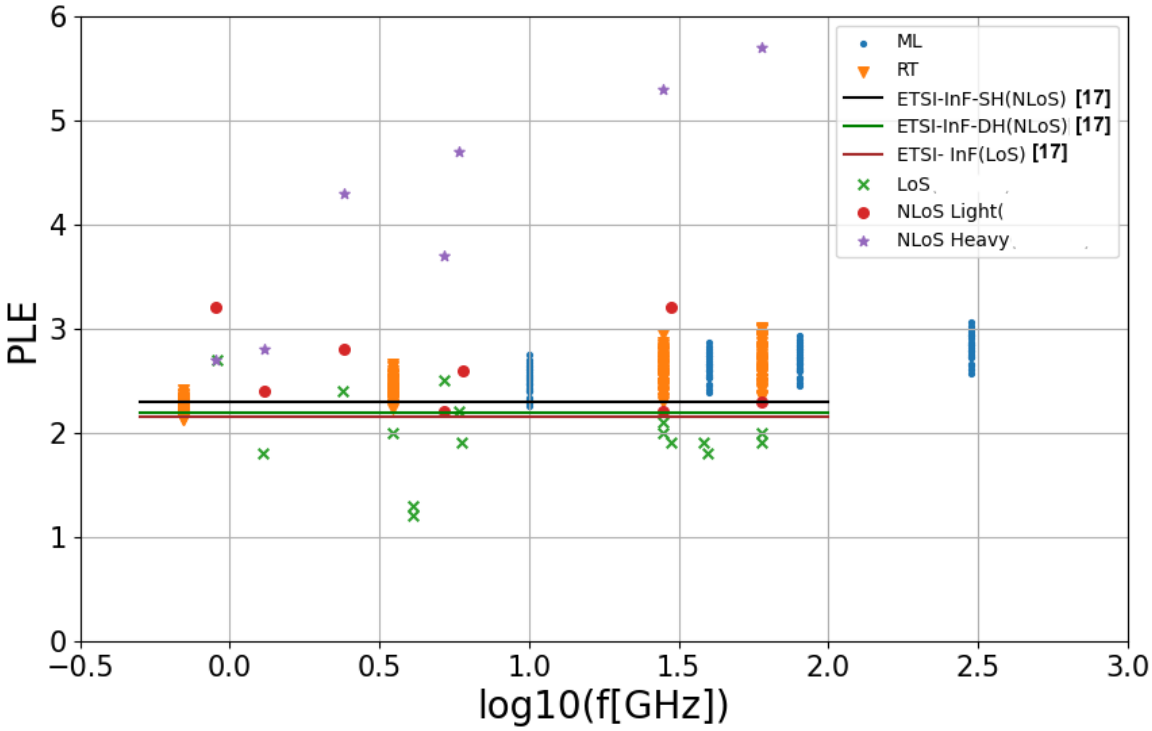



**Figure 4.28:** Model–literature comparison for path loss exponent.

The results returned by the proposed MLP-based model appear as a trade-off between previous evaluations, showing a PLE that slightly increases with both frequency (Fig. 4.28) and machine density (Fig. 4.26(a)). Since the PLE values referring to RT and ML correspond to mixed LoS/NLoS cases, it is not surprising that they fall between typical LoS and heavy NLoS values.

Similar considerations hold for $\sigma_{dB}$ (Fig. 4.29). The values of $\sigma_{dB}$ corresponding to RT and ML again refer to both LoS and NLoS, explaining why they can be higher than those reported separately for each condition.

**Rice Factor ($K$)** The performance of the model is outlined in Fig. 4.30, where the $K$ values predicted by the ML model for the industrial layouts belonging to the test dataset are compared with the corresponding ground-truth values from RT simulations. A close agreement is achieved in most cases, as also confirmed by the RMSE reported in Table 4.10.

A preliminary analysis based on the Chi-square test was conducted to investigate which features most affect $K$ in industrial environments. As shown in Fig. 4.31, MD is the

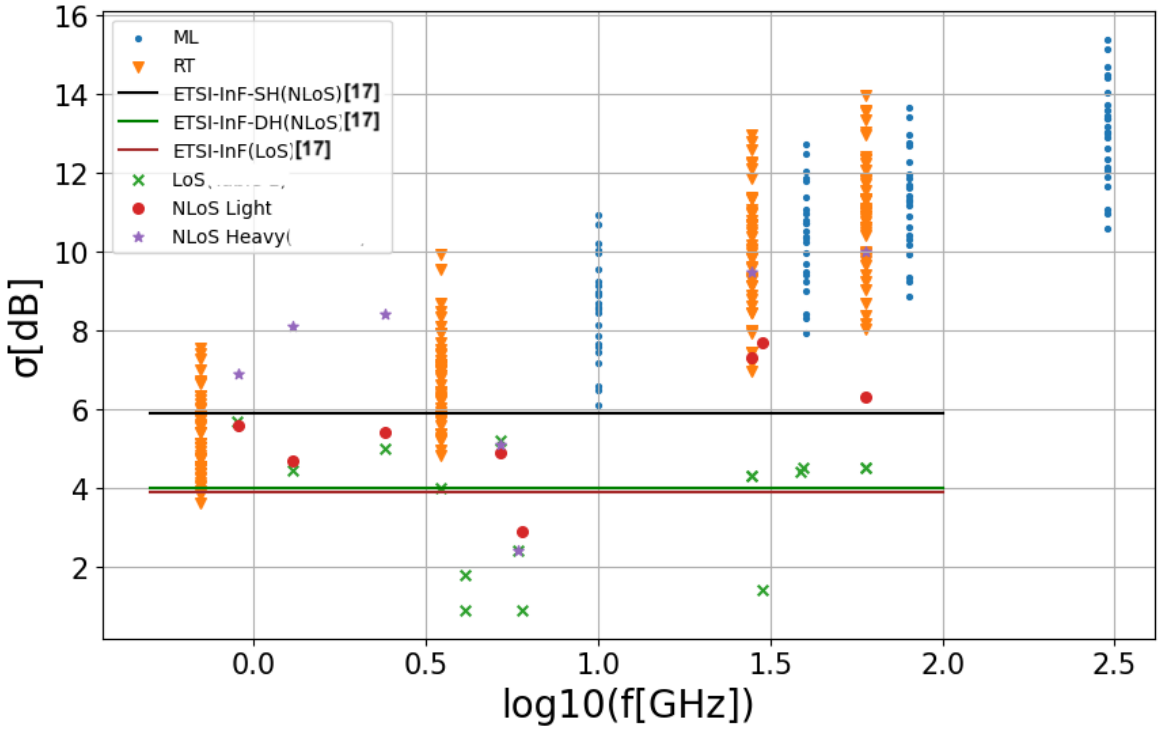



**Figure 4.29:** Model–literature comparison for shadowing level.

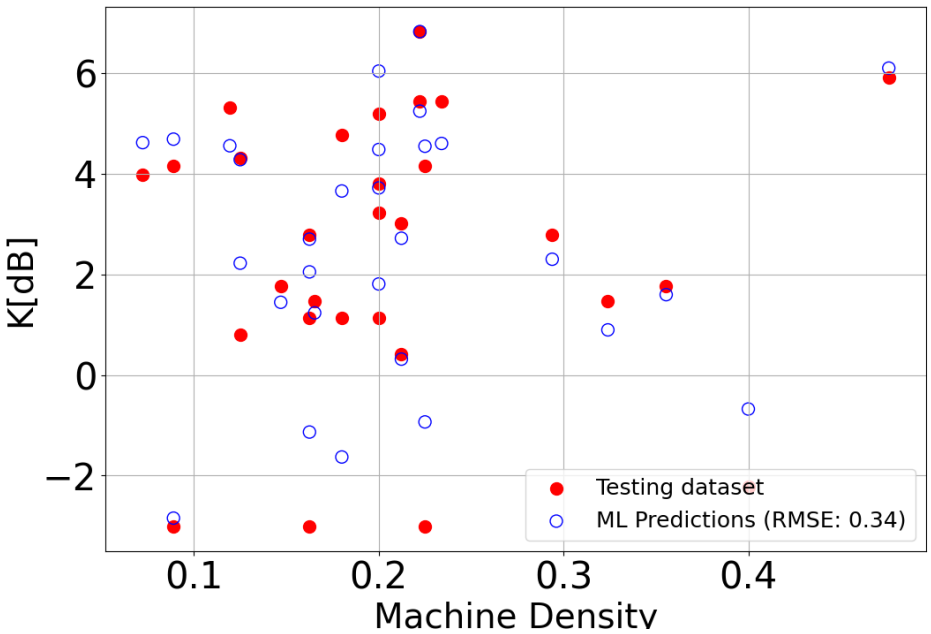



**Figure 4.30:** Comparison between ML prediction and RT ground truth (test set).

most significant predictor of $K$, followed by frequency and machine size, while spacing shows negligible impact.

Figure 4.31 and additional analyses show that $K$ tends to increase with $\log_{10}(f_{GHz})$,

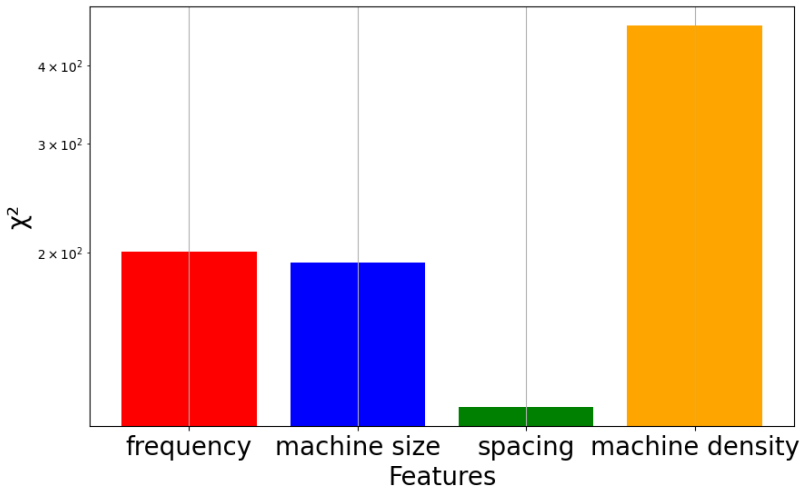



**Figure 4.31:** Feature importance for $K$ estimation.

machine size, and spacing—indicating a positive correlation—while it decreases with increasing machine density, as expected due to stronger obstruction. For illustration, in the 28 GHz case, $K$ plotted against MD for different MS values confirms the negative trend. These observations justify the use of ML models like MLP to capture residual nonlinear dependencies beyond simple regression.

A ready-to-use linear empirical model for predicting $K$ in industrial environments is also proposed:

$$K = A + B \cdot \log_{10}(f_{\text{GHz}}) + C \cdot MS + D \cdot MD, \tag{4.18}$$

where the least-squares optimized coefficients are $A = 0.221$, $B = 0.619$, $C = 0.359$, and $D = -2.0609$. Although this linear formulation is not optimal—since the true dependence is more complex—it remains accurate enough for practical use, as confirmed by the RMSE values in Table 4.10.

**Table 4.10:** Performance of ML model and empirical formula for $K$ estimation.

| Model | RMSE | Min K | Max K |
|---|---|---|---|
| ML Model | 0.34 | 0 | 6.7 |
| Empirical Formula | 0.62 | 0 | 6.7 |

**Extension of the work** An extension of this study was presented in "Wireless Industrial Channel Characterization Through Machine Learning and Ray Tracing Simulations" [128], where the same methodology was applied to assess the effect of transmitter placement on key propagation parameters. In this extended setup, the transmitter was moved from the wall to the center of the ceiling to investigate how elevation influences channel behavior. The results demonstrated that this configuration significantly

improved signal propagation, reducing the average PLE from 2.47 to 2.11 and the shadowing standard deviation from 8.24 dB to 4.37 dB, indicating smoother received power distribution and reduced multipath variability.

### 4.2.2 Coverage Map Prediction

This section is based on the work presented in the paper "On the Effectiveness of Radio Environmental Map Prediction Through Image-Based Deep Learning" [129], which has been submitted for publication.

In this work, a modified U-Net architecture is considered for REM prediciton. As discussed in the ML framework, U-Net is particularly suited for pixel-level classification and regression tasks thanks to its encoder–decoder structure with skip connections, which enables both the extraction of high-level propagation features and the preservation of spatial details from the input building maps. This makes it an effective choice for learning the mapping between city layouts and the corresponding LoS coverage maps.

Some recent works have competitively addressed REM prediction in urban environments using deep learning, usually relying on image-based inputs (e.g., city maps) and UNet-style architectures [80, 81, 130–133].

In order to achieve reliable prediction accuracy, they all simultaneously work/act on the multiple factors affecting the final performance with the highest likelihood, like the UNet structure and its (hyper-)parameters, the size of the training dataset, the resort to data augmentation techniques, and to auxiliary input data. In spite of the effort, the final, overall setting of the UNet sometimes resulted in a marginal performance improvement compared to other, similar studies.

Rather than contributing to this competitive challenge, a different analysis is proposed in this work, where the goal is not a further (marginal) reduction of the prediction error, but most of all a comparative evaluation of the performance dependence on the following key factors: UNet complexity, auxiliary input types, augmentation strategies, training dataset size and characteristics, and target label fidelity. Instead of presenting yet another accuracy benchmark, this contribution investigates the sensitivity of REM prediction accuracy to various architectural and input data choices. By systematically evaluating these elements, the aim is to derive general insights into what matters most to achieve practical, scalable, and robust REM prediction, especially under realistic constraints where full-scale RT simulations or large paired datasets may not be feasible. In doing so, this work fills a current gap in the literature by providing a roadmap for optimizing such systems beyond pure accuracy performance, offering both theoretical value and practical guidelines.

#### 4.2.2.1 Dataset Generation

Input data typically consists of environmental features, such as building maps and transmitter locations, while the output label describes the corresponding RF coverage characteristics (i.e., the REM). In addition to these primary input features and output labels, auxiliary data can be introduced to further enhance the model's predictive capabilities. Auxiliary data, collected in a faster way with respect to input data, refer to supplementary input information, such as LoS maps [81, 130] or (interpolated) sparse coverage maps [80, 132]. They provide context to improve generalization and accuracy. Previous studies have shown that incorporating auxiliary data can actually increase DL-based REM prediction accuracy by reducing uncertainty and improving spatial consistency [134].

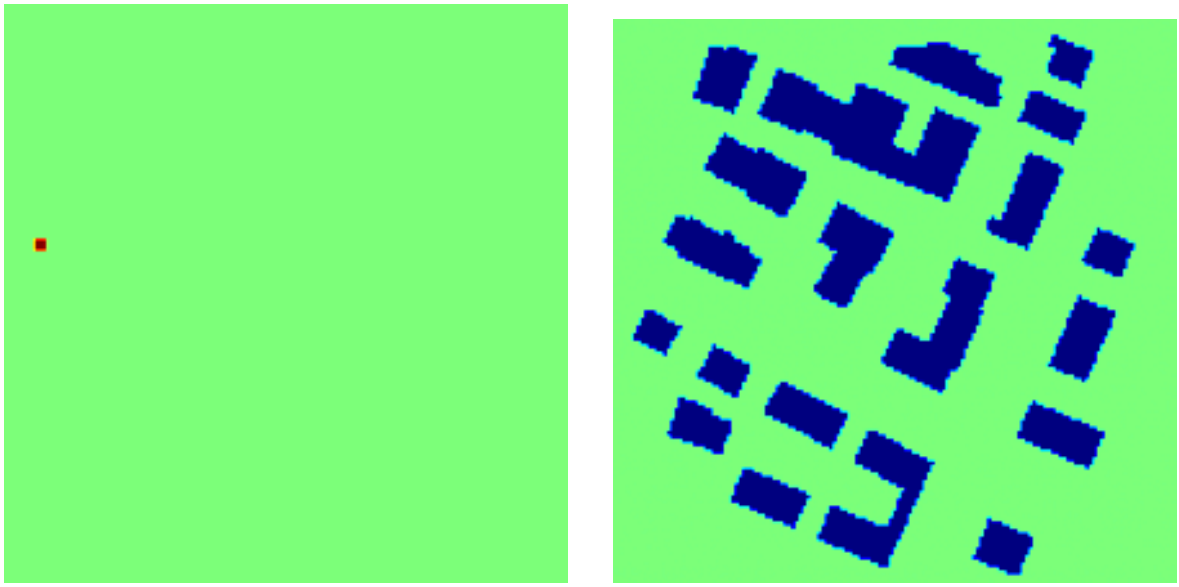
**Input Map Generation** In order to cope with the computational effort required by RT simulations to generate the propagation dataset to train and test the DL model, some expedients have been enforced in this study to properly restrict the REMs size. In particular, the analysis has been limited to microcellular urban layouts, i.e., with the TX and the RXs placed at a height respectively equal to 5m and 1.5m, i.e., well below the average height of the buildings included inside the urban area, equal to 14m. With this measure, given that propagation primarily occurs in the horizontal plane between the transmitter and receiver, we can simplify the simulations by using a 2D urban layout, significantly reducing their complexity.

A 5km $\times$ 5km high-resolution map of the city of Bologna (Italy) was randomly divided into 1000 smaller maps with a size of 150m x 150m, corresponding to suitable input samples for RT simulations. To ensure a meaningful urban environment, each tile contains a minimum of 25 buildings, serving as a threshold to filter out uninformative regions.

In particular, two different maps have been considered as input features for the DL model:

- **Transmitter Location Map:** A binary matrix accounting for the TX position in the map. The TX is randomly placed over the map region not covered by buildings, and its location corresponds to the only active pixel in the input matrix (Fig. 4.32a).
- **Building Map:** A binary matrix describing the position, size, and shape of buildings within the urban map. Buildings are marked as active pixels in the matrix, while the others are left inactive (Fig. 4.32b).

As clearly shown in Fig. 4.32, the information conveyed by the two input maps is limited to 2D data, i.e. height values are not included. Although electromagnetic propagation is inherently a 3D process, it is likely to occur below the rooftop level in microcellular scenarios, i.e., in the streets surrounding the buildings rather than above them. In this regard, 2D information is assumed to be sufficient for a reliable REM prediction.



(a) TX location (b) Buildings map

**Figure 4.32:** An example of input maps: transmitter map and buildings map.

**RT Simulation** In agreement with previous investigations, the propagation data to fill the database for training and testing the DL model has been collected by means of RT simulations carried out over the 1000 available urban maps. While a single TX has been placed in each map, a multitude of RXs have been spread along the streets between buildings, at a height of 1.5m and with a spatial resolution of 1 m. Simulations have been run at the frequency $f_0$ = 3.5GHz, and assuming an isotropic radiation pattern for both the transmitting and the receiving antennas. In order for the simulation to better mimic real-world propagation, diffuse scattering (S) can be enabled in the considered RT tool [104], in addition to reflection (R) and diffraction (D) considered in previous studies. The maximum number of bounces experienced by each ray, as well as the type of electromagnetic interactions have been set as summed up in Table 4.11, which represents an effective trade-off between the overall simulation time and the need for realistic results. The minimum power threshold was set to $P_{r,min}$ = -110 dBm, corresponding to a maximum path-loss equal to 110 dB, in agreement with previous works [80, 81]. Still in compliance with previous investigations, the field intensities carried by the different rays have been commonly summed up neglecting any phase relationship, i.e. multipath interference was not taken into account. That means, the total received power at any RX location has been computed as follows:

$$P_r = 10 \log_{10} \left( \sum_i P_{r,i} \right) \tag{4.19}$$

$$P_{r,i} = \frac{\lambda^2}{8\pi\eta} g_r(\theta_i, \varphi_i) \cdot \left| \vec{p}_r(\theta_i, \varphi_i) \cdot \vec{E}_i(Rx) \right|^2 \tag{4.20}$$

where N is the number of rays tracked by the RT tool, $(\theta_i, \phi_i)$ the direction of arrival

of the $i-th$ ray, $g_{rx}$ the directional gain of the receiving antenna, $\vec{p}_{rx}$ the polarization vector of the receiving antenna and $\vec{E}_i$ the electric field arriving at the RX through the $i-th$ ray. $\lambda$ is the wavelength and $\eta = 120\pi$ the free space impedance.

Just for the RT3 case in Table 4.11, an additional dataset was also generated, enabling coherent simulation mode, i.e., a phase-aware RT configuration in which the complex electric-field contributions associated with different propagation paths are coherently summed. In this case, the interference among the fields conveyed to the same RX along the corresponding rays has been taken into account in the computation of the overall received power according to the following expression:

$$P_{r,\mathrm{coh}} = 10\log_{10}\left(\frac{\lambda^2}{8\pi\eta}\left|\sum_{i=1}^{N_r}\sqrt{g_r(\theta_i,\varphi_i)}\cdot\vec{p}_r(\theta_i,\varphi_i)\cdot\vec{E}_i(Rx)\right|^2\right) \tag{4.21}$$

**Table 4.11:** RT settings and corresponding computation time for 1000 simulations.

| Type | Max. no. of int. | R | D | R and D | S | Time |
|---|---|---|---|---|---|---|
| LoS | 0 | 0 | 0 | 0 | Off | Fast |
| RT1 | 1 | 1 | 1 | 1 | Off | 12h |
| RT2 | 2 | 2 | 1 | 2 | Off | 1 week |
| RT3 | 3 | 3 | 1 | 2 | On | 2 weeks |

To further speed up the simulation process, parallel computing was also enforced. The simulations were executed on a 12-core CPU, enabling concurrent processing of different maps. The computation time $T_c$ reported in Table 4.11 refers to the parallel computation on the 12 cores. The values can then be easily scaled to a different number of cores $n_c$ as $T_c \cdot 12/n_c$. The major parameters related to the simulated microcellular scenarios are reported in Table 4.12.

**Auxiliary Input Data** In addition to the primary input maps (Tx position and building layout), the dataset can include auxiliary data to enhance the learning process. Auxiliary data provides additional input information that further helps the model improving its insight into the complex relationship between the primary input files and the corresponding output label (i.e., the REM). In this study, different types of auxiliary data are considered, as briefly described in the following list:

- **Simplified Coverage Map:** it simply consists of a REM achieved through RT simulations carried out with a reduced number of interactions compared to the REMs generated and used for training and test (Fig. 4.33a). Although these simulations are run enabling a lower number of reflections/diffractions/scattering, they provide additional propagation insights while significantly saving computation time compared to full-scale simulations. In the following, REMs

**Table 4.12:** Major parameters of the microcellular scenarios.

| Parameter | Value |
|---|---|
| Map Size | 150 m × 150 m |
| Maps Number | 1000 |
| Antenna Type | Isotropic |
| Transmit Power | 0 dBm |
| Transmitter Height | 5 m |
| Receiver Height | 1.5 m |
| (Average) Building Height | 14 m |
| Carrier Frequency | 3.5 GHz |
| Image Resolution | 1 RX/m |
| Minimum Threshold Power | -110 dBm |

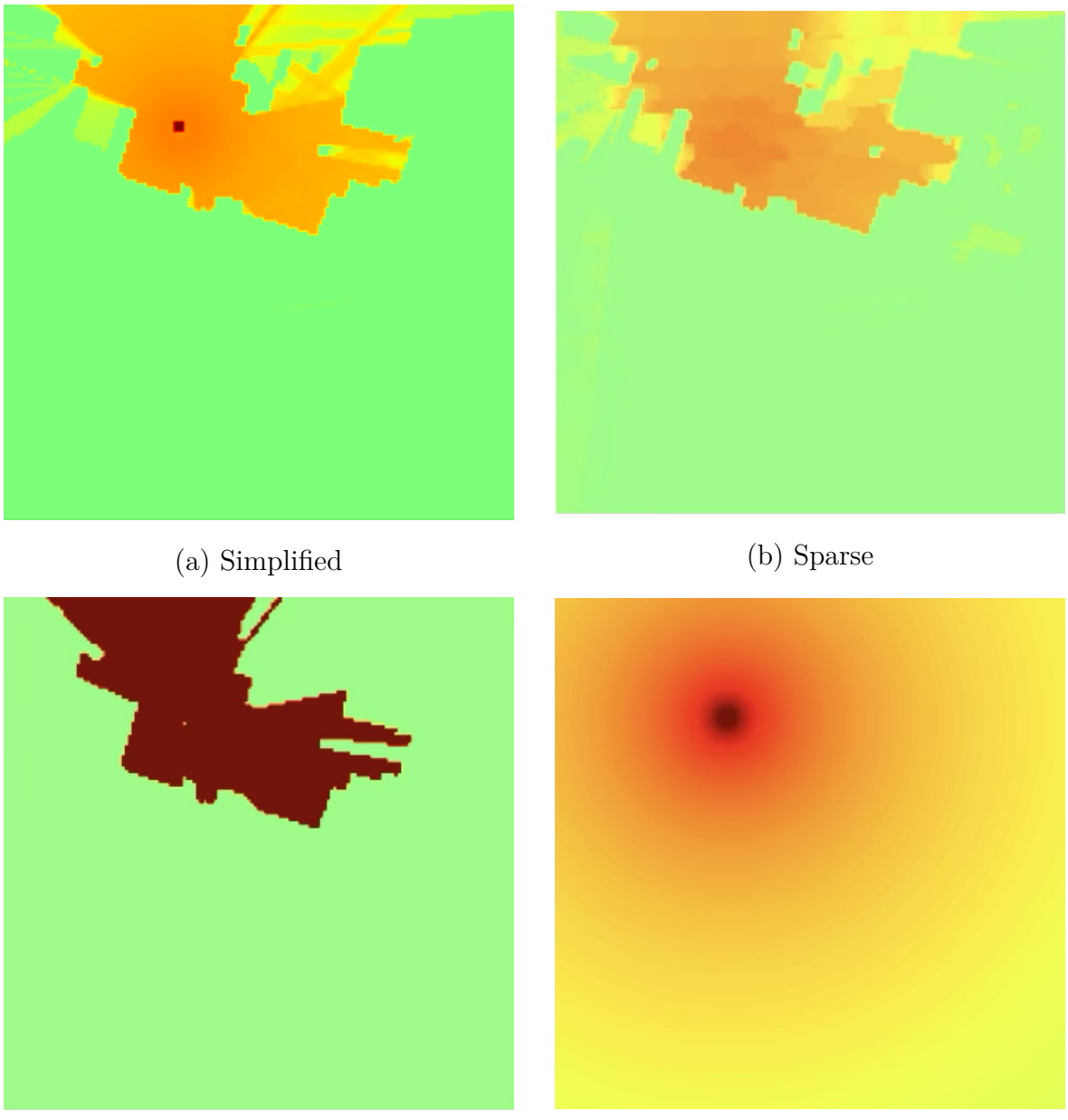
corresponding to the RT setting marked as RT1 in Table 4.11 are considered as possible auxiliary input data.

- **Interpolated Sparse Map:** it is a coarse coverage map based on propagation data collected from a lower number of RXs, i.e., corresponding to a reduced spatial density of RXs compared to the REMs generated and used for training and testing. Then, the sparse coverage map can be interpolated to get a full resolution image (Fig. 4.33b). In the following, sparse maps achieved with 20m spacing between the RXs are considered.
- **LoS Map:** it is a binary map providing information about the existence of indicated LoS between the TX and each RX location spread on the map (Fig. 4.33c).
- **Hata-like Map:** it is generated by means of the following simple analytical formula:

$$\begin{aligned} P_r[\text{dB}] &= P_t[\text{dB}] - PL[\text{dB}] \\ PL[\text{dB}] &= 10 \cdot \alpha \cdot \log_{10}\left(\frac{d}{d_0}\right) + PL_0(d_0) \end{aligned} \tag{4.22}$$

where $\alpha$ is the *path-loss exponent* (PLE, or *propagation factor*) and $PL_0(d_0)$ is the free space attenuation at a reference distance $d_0$, here set equal to 1 m. Eq. 4.22 describes the average range dependence of received power, and its slope is related to the value of $\alpha$. In an urban environment, $\alpha$ usually belongs to the range [3-3.5] [1]. The result is a smooth, linear decay of received signal strength with distance (in a log-log scale) (Fig. 4.33d). Although this map does not account for specific building obstructions, it nevertheless provides additional information about the expected (average) rate of power reduction with distance, which can

anyway support the prediction of the output target REM to some extent. In the following $\alpha = 3.5$ is considered.



(a) Simplified

(b) Sparse

(c) LoS

(d) Hata-like

**Figure 4.33:** Different types of auxiliary input maps.

The different auxiliary input maps are expected to improve the accuracy of the REM prediction to a different extent. At the same time, they pose different requirements on the final user of the trained model for REM prediction. In order to generate simplified coverage maps, the final user must have access to both the urban maps and an RT tool, that might be not always obvious. Interpolated sparse maps can also be achieved by means of RT simulations, but they might also consist of real propagation data collected from working end-user equipment (e.g., sensor nodes) distributed over the urban area [132]. With reference to the LoS maps, they certainly require the availability of the

urban layout, but they can be computed without a RT model, as the presence of LoS is a geometrical condition that can be checked in some different ways [81]. Finally, the computation of Hata-like maps just requires the value of the PLE, which can be fixed by means of simulations, but also resorting to ad hoc experimental data, experience, or suggestions from existing literature.

**Data Augmentation** Data augmentation is widely used in DL and refers to any technique aimed at artificially increasing the size of the training dataset by creating a modified representation of existing data. This approach helps preventing overfitting and improves the model's robustness by exposing it to diverse scenarios. In the context of RF coverage prediction, data augmentation can be particularly beneficial as it mitigates the need for extensive data collection or computationally expensive RT simulations to generate additional labeled samples.

To maximize the benefits of data augmentation while preserving the original spatial resolution of 150m x 150m tiles, a comprehensive transformation strategy was employed. First, all possible rotations (90°, 180°, and 270°) were applied to each image, corresponding to three additional images. Then, horizontal, vertical, and diagonal flips were also enforced, thus introducing four new representations. Since some combinations lead to the same final result, the size of the dataset extracted from RT is finally expanded by a factor of eight. This dataset extension exposes the model to a broader range of spatial patterns and propagation scenarios, improving its ability to generalize across different urban layouts and transmitter placements.

**Out Map Generation** The final output maps, returned by RT simulations performed with high resolution, represent the ground truth for training and testing the DL model. These maps provide the spatial distribution of received power levels across the outdoor urban environment, accounting for the RF signal strength spatial fluctuations due to building obstruction and/or multipath propagation. To make these maps effective input for NNs, they are usually converted into grayscale images with values ranging from 0 to 1.

#### 4.2.2.2 DeepUNet for REM Predictions

The proposed DeepUNet architecture is designed to effectively capture both low-level and high-level spatial features necessary for accurate RF coverage prediction. As illustrated in Fig. 4.34, the model follows a symmetric encoder-decoder structure with five levels of depth, enabling the extraction of hierarchical representations from the input maps. each followed by a non-linear activation function known as the Rectified Linear Unit (ReLU), and a max-pooling operation for spatial downsampling (red arrows). The use of a non-linear activation function is essential in deep neural networks to allow the model to learn and represent complex, non-linear relationships in the data. In particular, ReLU is one of the most commonly used activation functions due to its computational efficiency and ability to mitigate the vanishing gradient problem.

ReLU operates by outputting zero for any negative input and passing positive inputs unchanged, thereby introducing non-linearity without saturating the output. The max-pooling operation, on the other hand, is employed to progressively reduce the spatial dimensions of the feature maps. By retaining only the maximum value within small local regions (typically $2 \times 2$), max pooling helps in reducing computational complexity and achieving spatial invariance, while also preserving the most salient features in the data. Together, convolution, activation, and pooling layers enable the encoder to extract meaningful hierarchical features that are crucial for accurate RF coverage prediction.

The decoder mirrors the encoder using transpose convolutions for upsampling. Blue boxes indicate the number of feature maps (filters) at each stage, growing deeper in the encoder and symmetrically reducing in the decoder. Skip connections between corresponding encoder and decoder layers (gray arrows) help preserve spatial details, which are crucial in high-resolution prediction tasks. The network outputs the final prediction map via a $1 \times 1$ convolutional layer. In addition to the main output, auxiliary outputs are extracted (yellow arrows) at each decoder stage to implement deep supervision, which strengthens multi-scale feature learning and stabilizes training. The considered architecture is summarized in Table 4.13:

**Table 4.13:** DeepUNet architecture details.

| **Layer Type** | **# of Layers** | **Filter Size** | **Activation Function** |
|---|---|---|---|
| Convolutional Layer | 22 | $2 \times 2$ | ReLU |
| Pooling Layer | 5 | $2 \times 2$ (MaxPool) | – |
| Upsampling Layer | 5 | $2 \times 2$ (Transpose) | – |
| Skip Connections | 5 | – | – |
| Final Output Layer | 1 | $1 \times 1$ | – |

To guide the training process, a hybrid loss function is employed, combining MSE and SSIM. In the context o ML, a loss function is a quantitative measure of the difference between the model's predicted output and the actual result (known as the ground truth — in this case, the RT-generated coverage map). During training, the model parameters are iteratively adjusted to minimize this loss function, thereby improving the accuracy of predictions. The two components of the hybrid loss serve complementary purposes: MSE focuses on pixel-wise accuracy, while SSIM emphasizes structural similarity and perceptual fidelity. The hybrid formulation aims to ensure both accurate value prediction and preservation of spatial patterns, which are important in radio signal mapping.

Deep supervision is applied by introducing auxiliary losses at intermediate decoder stages, each using the same hybrid loss. The total training loss is computed as:

$$L_{total} = L_{main} + \alpha_1 L_{aux1} + \alpha_2 L_{aux2} + ... + \alpha_n L_{auxn} \tag{4.23}$$

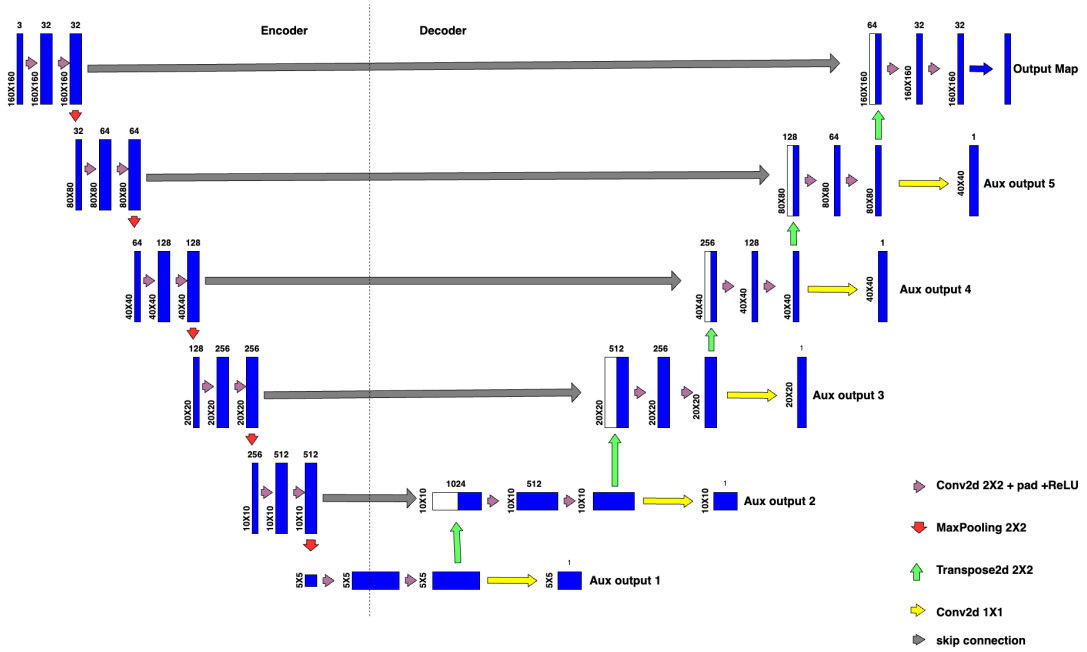



**Figure 4.34:** DeepUNet Architecture.

where $L_{main}$ is the loss at the final output, $L_{auxi}$ are auxiliary losses, and $\alpha_i$ are corresponding weights.

It is important to clearly distinguish between the training and evaluation metrics. The training process relies on the hybrid loss function (MSE + SSIM), while the model's performance is evaluated using the RMSE on the test dataset. This separation aligns with established practice and avoids confusion between optimization and evaluation objectives.

Since the output coverage maps are normalized to grayscale values between 0 and 1, RMSE is computed as:

$$RMSE_{GS} = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (P_{r,norm,true} - P_{r,norm,pred})^2} \tag{4.24}$$

Since the grayscale values are derived from the received power levels using min-max normalization, the RMSE in grayscale can be approximately converted to RMSE in dB using the following relationship:

$$RMSE_{dB} \approx RMSE_{GS} \cdot (P_{r,max} - P_{r,min}) \tag{4.25}$$

In agreement with [130], the RMSE is calculated only over the map area where valid RT based ground truth values exist, e.g., excluding the area covered with buildings. This ensures that the evaluation is limited to areas relevant to RF propagation, providing a more reliable assessment of model performance. Although this may sound obvious, it's worth pointing out that RMSE is instead computed over the whole map (i.e.,

buildings included) in some other works [81, 132, 133], which is expected to sweeten the performance evaluation.

#### 4.2.2.3 Results and Discussions

This section analyzes the prediction results by organizing the discussion into three logical parts: sensitivity analysis, fast fading tracking capability, and generalization potential towards real-world propagation.

**Sensitivity Analysis: Effect of Model and Data Enhancements** Standard UNet, trained on RT3 targets, without auxiliary data or without data augmentation, is here considered as the Baseline Model (BM). As reported in Table 4.14, it yields a normalized RMSE of 0.25 (13.25 dB), which is actually not particularly good. Enabling data augmentation alone reduces the RMSE to 0.16 (8.48 dB), proving its regularization effect and contribution to improved generalization. Replacing the Standard UNet with the DeepUNet architecture, still without auxiliary inputs and data augmentation, results in an RMSE of 0.14 (7.42 dB), highlighting the benefit of deeper feature extraction and hybrid loss supervision.

Finally, Table 4.14 also shows that auxiliary input information turns out to be beneficial to the greatest extent, with RMSE dropped to less than 7dB regardless of the type of auxiliary data map. This result confirms that low-complexity simulated REMs can serve as effective training support for predicting higher-fidelity coverage maps. Interestingly, the RMSE reduction brought by auxiliary input maps is directly related to the complexity of their generation process: Hata-like maps can be quite easily achieved, but also correspond to the lowest gain, whereas simplified coverage maps computation requires heavier effort but also produces the most remarkable advantage. Not surprisingly, the most accurate REMs prediction, corresponding to an RMSE equal to 0.03 (1.59 dB), is achieved when the DeepUNet architecture, data augmentation, and auxiliary input are simultaneously enforced, as also shown in Fig. 4.37. The RMSE improvement triggered by the different cases considered in Table 4.14 is also represented in Fig. 4.35.

Fig. 4.36 shows the training loss and test RMSE over 100 epochs for the best-performing model configuration (DeepUNet + RT1 auxiliary input). The model exhibits stable and monotonic convergence, with the training loss decreasing consistently and the test RMSE stabilizing at a low value. The minimal gap between training and test curves confirms strong generalization and the absence of overfitting. This behavior reinforces the reliability of the selected configuration and suggests that the learning process is both stable and efficient.

The qualitative results for the best-performing configuration—DeepUNet with RT1 map as auxiliary input—are shown in Fig. 4.37, which includes the ground truth and predicted coverage maps for both the test samples with minimum and maximum RMSE. While the minimum RMSE case illustrates excellent agreement between the prediction

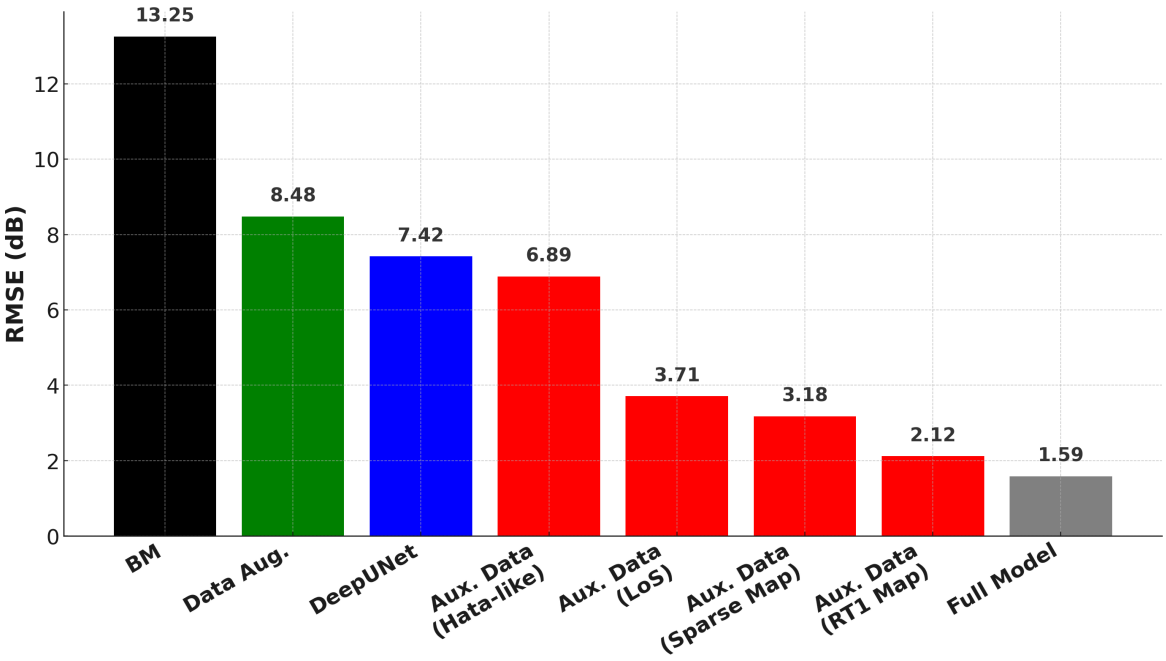



**Figure 4.35:** Bar Plot of RMSE for different model

**Table 4.14:** Step-by-step performance analysis.

| Model Type | Aux. Data | Data Aug. | RMSE | RMSE (dB) |
|---|---|---|---|---|
| Standard U-Net | None | No | 0.25 | 13.25 |
| | | Yes | 0.16 | 8.48 |
| DeepUNet | None | No | 0.14 | 7.42 |
| Standard U-Net | Hata-like | No | 0.13 | 6.89 |
| | LoS | | 0.07 | 3.71 |
| | Sparse Map | | 0.06 | 3.18 |
| | RT1 Map | | 0.04 | 2.12 |
| DeepUNet | RT1 Map | Yes | 0.03 | 1.59 |

(4.37b) and ground truth (4.37a), the maximum RMSE sample (4.37d and 4.37c) also shows a strong qualitative match. Although some deviation is expected, especially in complex propagation zones, the model still captures the dominant structures and gradients accurately. This consistency across both extremes reflects the model's overall robustness and stability, as confirmed by the very low standard deviation of RMSE reported for this configuration. It highlights that even in relatively harder scenarios, the model maintains reliable predictive quality.

**Fast Fading Tracking Capability under Coherent Propagation** To assess the model's ability to handle more complex propagation effects, simulations were repeated carried out in the coherent case, where multipath components arriving at the receiver

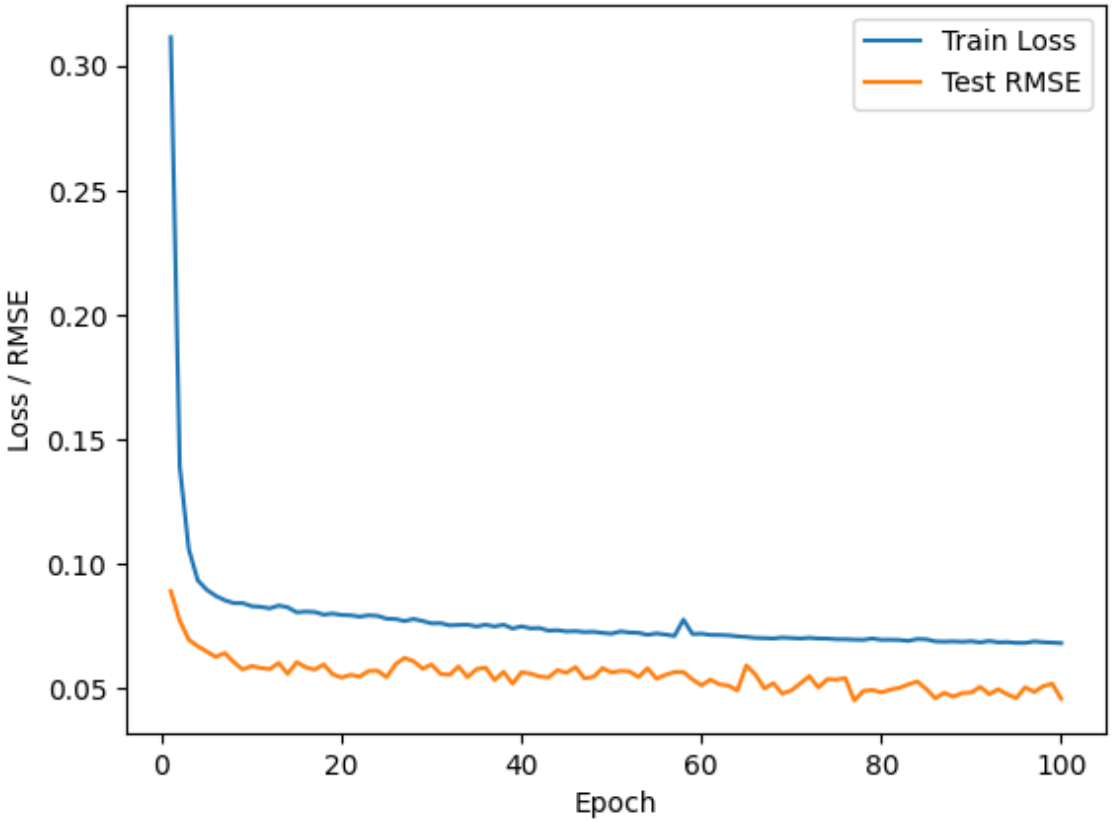
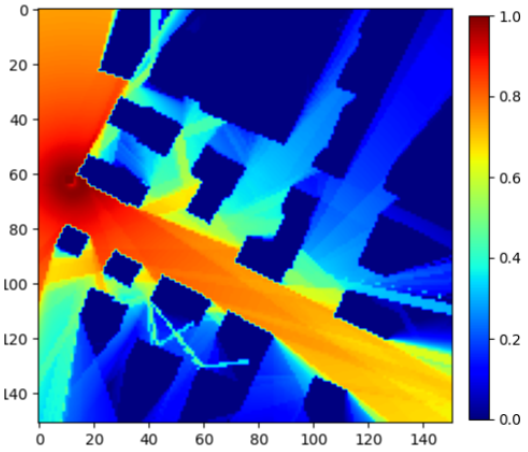



**Figure 4.36:** Train and Test loss

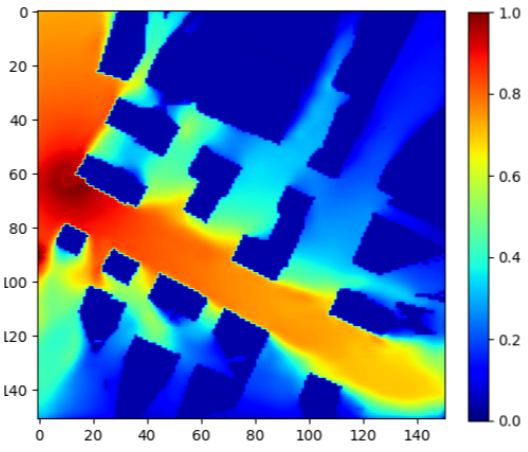
are combined, taking into account phase information, therefore capturing the interference patterns related to fast fading. Unlike the non-coherent scenario, where power contributions from different rays are simply summed up, neglecting any phase relationship, the coherent model introduces stronger local fluctuations and fine-grained spatial variations that might represent a further challenge for the model learning process. In this context, the received power should ideally be modeled according to Eq. (4.21), which accounts for the vectorial (i.e., phase-aware) summation of the field components, as opposed to Eq. (4.19), which assumes incoherent power addition.

Surprisingly, this added complexity does not seem to be affecting the model's performance to a significant extent. When tested on coherent REMs using LoS maps as auxiliary input, the final RMSE was found equal to 0.07 (3.7 dB), which is nearly identical to the value obtained under non-coherent conditions. This result suggests that the model generalizes fairly well to more complex propagation phenomena like multipath spatial interference.

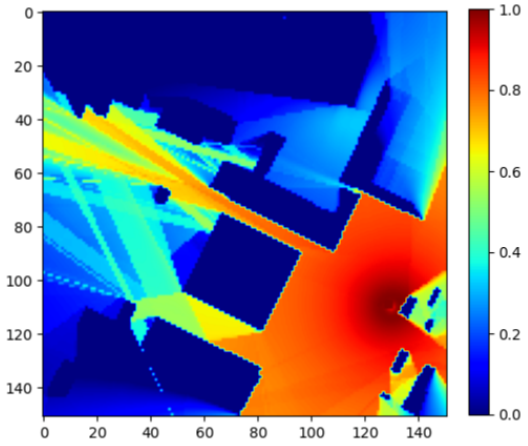
Figure 4.38 provides a qualitative comparison between the ground-truth coherent REM and the predicted one. Owing to the enforced coherent simulation mode, the ground truth REM coming from RT now clearly includes interference-induced multipath ripples. Although the DeepUNet cannot actually track them at the point-specific level, it anyway can, nonetheless, fairly well predict the macro structure of the interference pattern (especially present in the LoS region in Fig. 4.38).



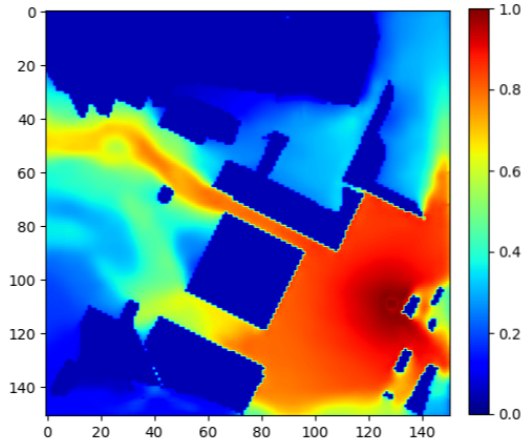
(a) Ground truth - Min RMSE case



(b) Prediction - Min RMSE case



(c) Ground truth - Max RMSE case



(d) Prediction - Max RMSE case

**Figure 4.37:** Examples of ground truth vs. prediction in the best case (a,b) and worst case (c,d).

**A Glance to Real World Propagation** In practical scenarios, the ultimate goal is to predict REMs as they actually are in the real world. However, collecting comprehensive real-world measurement data for training DL models is infeasible due to time, cost, and logistic limitations [135]. To simulate such conditions to some fair extent, a realistic evaluation strategy involves training the DeepUNet model on synthetic data generated by a simplified RT simulation (RT1) and testing it on data from a more accurate and detailed simulation (RT3), treated as a proxy for real-world measurements.

This approach allows for assessing the model generalization ability when faced with increased physical complexity and previously unseen propagation behaviors. When the model is both trained and tested on RT1 data, it achieves an RMSE of 0.05 (2.65 dB).

After training on RT1 and testing on RT3, the RMSE rises to 0.8 (4.24 dB). As clearly shown in Fig. 4.39, the main reason for this increase lies in the poor agreement corresponding to the furthest NLoS RX locations. This is actually not surprising, as

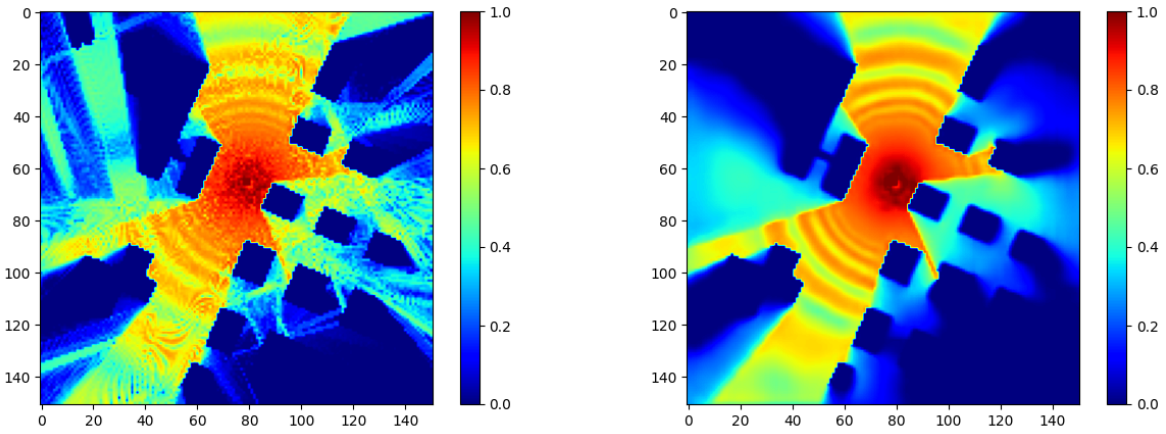
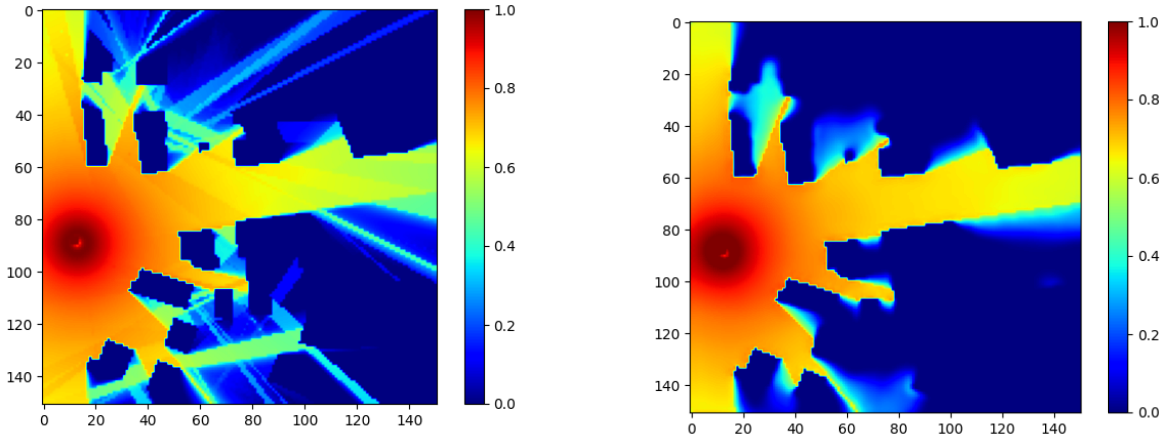


**Figure 4.38:** Prediction results under coherent propagation conditions. Left: Ground Truth REM. Right: Predicted REM.

the single bounce enabled in RT1 can spread rays only to the RX positions closer to the TX. Therefore, the training stage could just feed the model with information about short-range propagation, unless LoS conditions occur. Although image-based U-Net can quite reliably grasp information on the propagation process from the training dataset, any limitations or flaws in the database, making it not well representative of the real world target process to be modeled, automatically reduce the final prediction skill, regardless of the degree of refinement of the model. Although this should sound quite clear, it is worth pointing out.



(a) Ground truth (b) Prediction

**Figure 4.39:** Prediction of RT3 map based on training on the RT1 map.

## 4.3 Wideband Channel Modeling

This study is based on the work presented in "A Machine Learning Approach to Wireless Propagation Modeling in Industrial Environment" [111], where the same methodology described in subsection 4.2.1.1 was extended to wideband analysis for the

evaluation of DS. Although attenuation will always represent a crucial issue in wireless communication systems, distortion can also further impair the correct symbol sequence detection at the receiver side. In particular, time dispersion still due to multipath propagation can contribute to the overall degree of distortion to an extent that depends on the DS experienced over the environment and the symbol time length [1]. If M different propagation paths can be identified between a TX and RX pair, and $P_i$ and $\tau_i$ represent the intensity and the propagation delay of the i-th path, then:

$$DS = \sqrt{\sum_{i=1}^{M} \frac{P_i}{P} \cdot (\tau_i - \langle\tau\rangle)^2} \tag{4.26}$$

where $P = \sum_{i=1}^{M} P_i$ and $\langle\tau\rangle = \sum_{i=1}^{M} \frac{P_i}{P} \cdot \tau_i$. Averaging the DS values over many, different TX, and RX pairs leads to a final, mean DS somehow characteristic of the whole propagation environment.

The dataset, RT configuration, feature set (MS, SP, MD, frequency), and the learning pipeline (linear trend vs. $\log_{10}(f)$ (Fig. 4.40)+ MLP on residuals) follow exactly the same procedure described in Sec. 4.2.

The training and validation of the ML models for DS prediction follow the same procedure adopted for PLE and shadowing in Sec. 4.2. Several regression models have been investigated (MLP, XGBRegressor, SVR, RF) to capture the residual dependence of DS on the geometrical features (MS, SP, MD), once the average frequency trend has been removed.

The best–performing configuration of the MLP was selected after a careful hyperparameter search, considering the number of hidden layers, neurons per layer, activation function, and solver. Table 4.15 summarizes the final set of hyperparameters for the MLP tailored to DS regression.

**Table 4.15:** Major parameters of the neural networks for PLE and $\sigma_{dB}$ regression

| **Hyperparameter** | $\mathbf{DS_{mean}}$ |
|---|---|
| Number of hidden layers | 1 |
| Neurons distribution across the layers | 3/4/1 |
| Activation function | tanh |
| Solver for weight optimization | lbfgs |

The results confirm that the MLP outperforms the other models in terms of RMSE while also providing the most consistent generalization capability, especially when tested against feature values not included in the training set.

**Results and Discussions** Table 4.16 shows that the learning process can be accomplished also for DS prediction. Although the four considered ML methods turn out quite accurate, the MLP-based model outperforms the others. The model flexibility

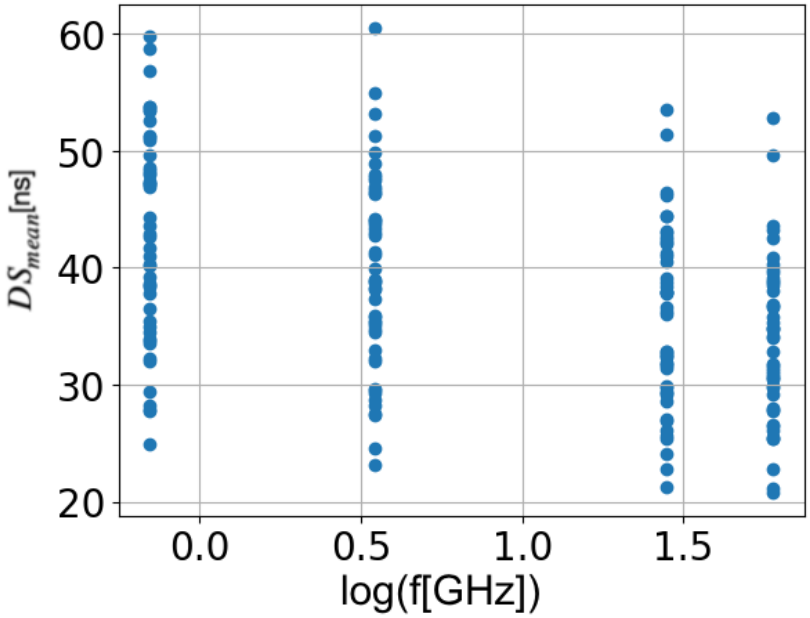



**Figure 4.40:** Model - literature comparison for shadowing level

and consistency are again investigated in Fig. 4.41, where the $DS_{mean}$ values returned by RT simulations are reported for different MD and frequency together with the values achieved from the trained MLP-based model corresponding to the same new set of features previously referred to (Table 4.9).

**Table 4.16:** Model Evaluation for $\mathbf{DS_{mean}}$ and min and max values

| | RMSE of different models | | | | Min value | Max value |
|---|---|---|---|---|---|---|
| | MLP | XGBR | SVR | RF | | |
| $DS_{mean}$[ns] | 2.2 | 2.8 | 4.9 | 4.6 | 20.79 | 52.84 |

As shown in Fig. 4.41, the outcomes from the MLP for the values of the fresh features are physically sound and consistent with the RT results. Best-fit lines are also traced in Fig. 4.41, aimed at highlighting the average dependence of $DS_{mean}$ on MD and frequency. The corresponding analytical formulation is reported in the following eq. 4.27:

$$DS_{mean} = \mathrm{k}_5 \cdot \frac{1}{\sqrt{MD \cdot f_{GHz}^{\frac{1}{\mathrm{k}_6}}}} \tag{4.27}$$

The least square method can be leveraged to compute the best $\mathrm{k}_5$ and $\mathrm{k}_6$ values at each different frequency (continuous and dashed lines in Fig. 4.41. A simple, average DS prediction formula describing its sensitivity to MD and frequency can be instead achieved by extending the least square computation all over the whole dataset. In this case, the values of the coefficients were found to equal $\mathrm{k}_5$=8.35 and $\mathrm{k}_6$=20.35.

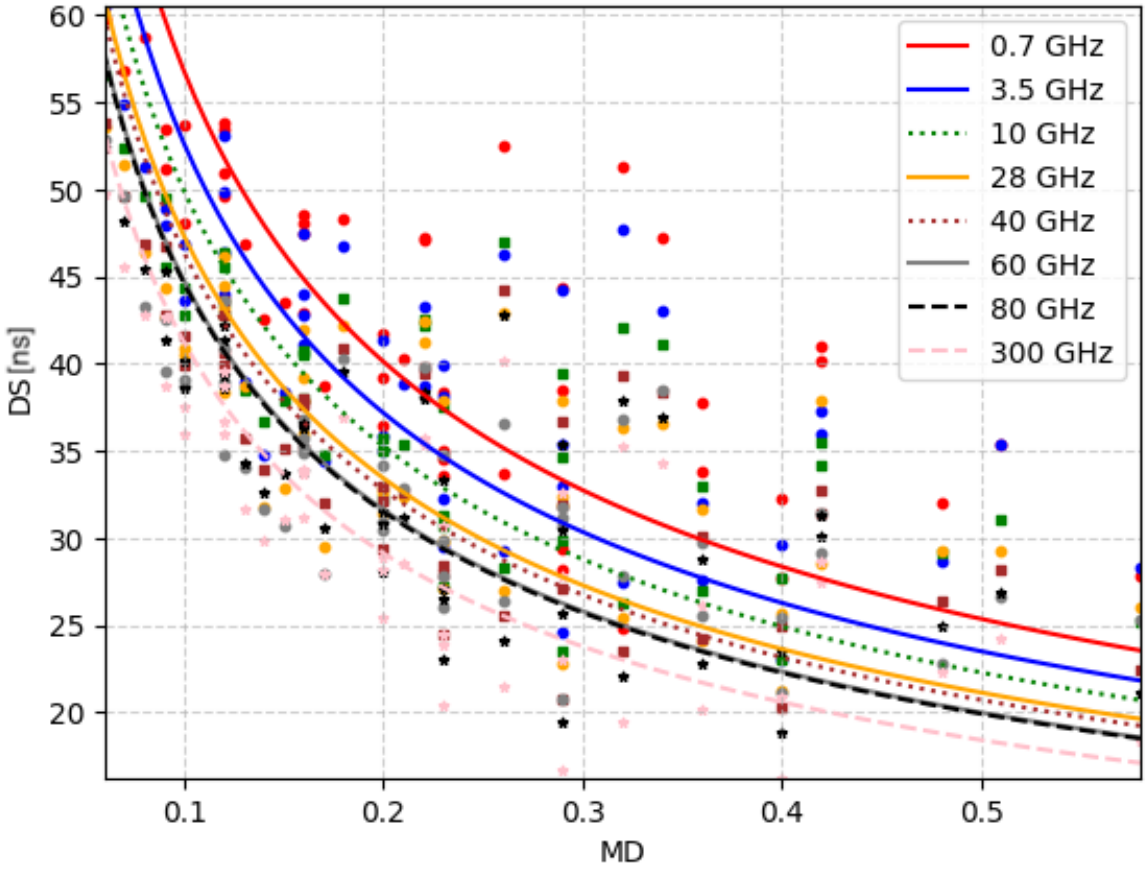



**Figure 4.41:** DS sensitivity to MD and frequency for MLP-based model

**Comparison with Previous Studies** A comparison in terms of DS is presented in Fig.4.42. Since the DS values reported in previous studies [72, 116–118, 122–127] and in the technical report [17] for NLoS conditions are rarely categorized according to the clutter level, Fig.4.42 distinguishes only between LoS and NLoS scenarios. The results obtained from the RT and ML assessments show fair agreement with the experimental findings (green crosses and red dots in the figure), indicating a slight decrease in DS with increasing frequency. This behavior is consistent with the trend already visible in Fig.4.27, where the difference between the mean DS values at 0.7GHz and 300GHz is approximately 7ns. The comparison with the simplified DS formulas provided in [17] appears slightly less accurate but remains acceptable. In particular, the standard deviation of DS ($\sigma_{DS}$) reported in [17] is about 18 ns for both LoS and NLoS conditions, meaning that most of the values predicted by the combined RT and ML approach fall within the $\pm\sigma_{DS}$ interval around the average values suggested in [17].

It is finally worth noting that although the learning process was based on data collected from simulations carried out for the same size of the industrial shed, results discussed in this section have turned out to be in general agreement with the previous model and experimental investigations clearly referred to different industrial environments and machinery layout. This confirms the model is flexible and versatile, and suggests

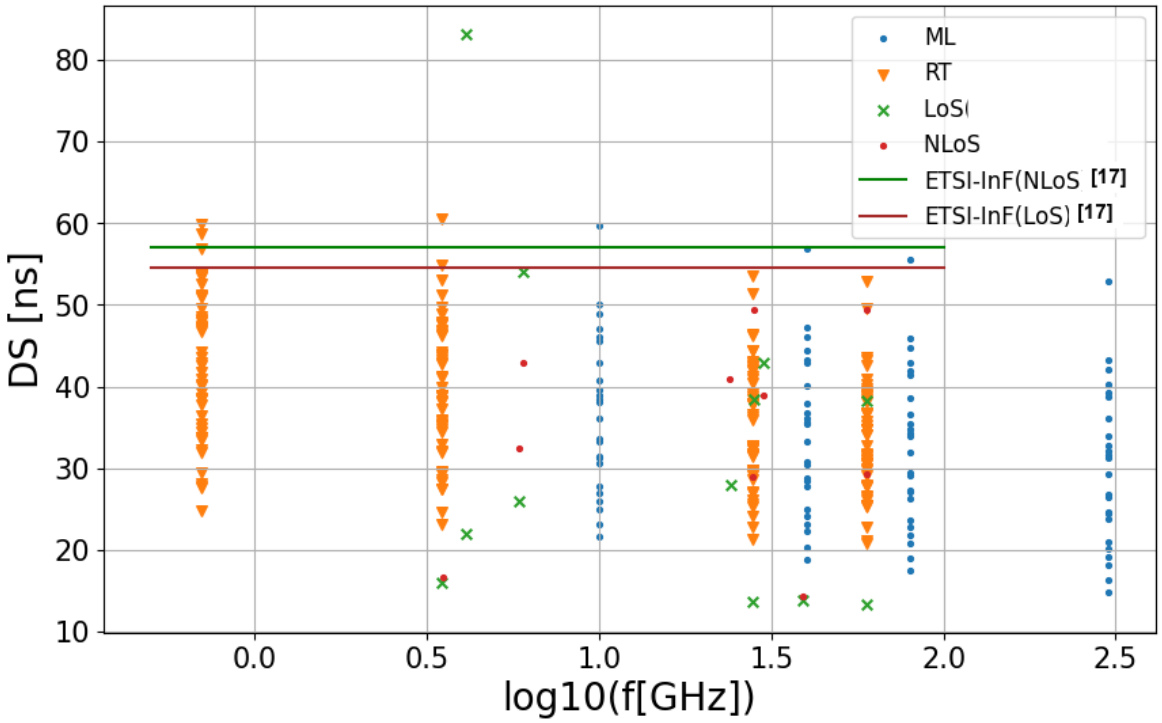



**Figure 4.42:** Model - literature comparison for DS

it has fair generalization skills also in the environment domain.

**Extension of the work** An extension of this work has been conducted to further analyze the impact of transmitter configuration on wideband channel characteristics, specifically focusing on DS. In this extended study, the transmitter was repositioned from the wall to the center of the ceiling, allowing assessment of the influence of transmitter height on temporal dispersion. The results revealed a substantial improvement in channel performance, as the mean DS decreased from 37.83 ns to 5.01 ns, indicating shorter multipath delays and enhanced temporal coherence. These findings confirm that the ceiling-mounted configuration provides a more stable and reliable wireless channel in industrial environments [128].

## 4.4 Object Recognition

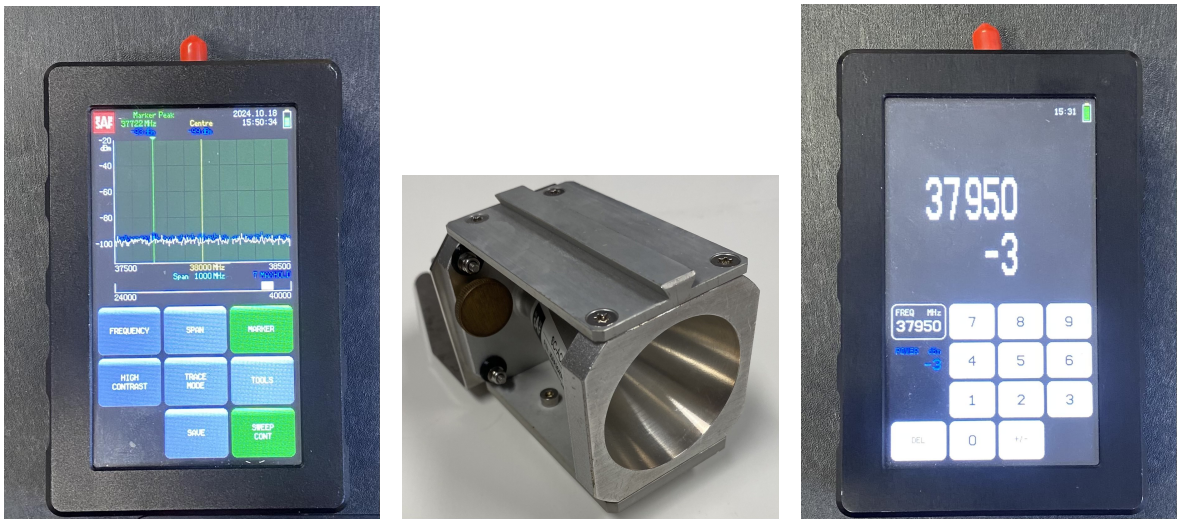
This section is based on the work presented in "1D Convolutional Neural Network for Object Recognition Through Electromagnetic Backscattering in the Frequency Domain" [136], which has been submitted for publication.

OR based on EM backscattering offers a privacy-preserving and environment-robust alternative (or complement) to camera-based vision systems. When illuminated by EM waves, an object re-radiates a backscattered response whose spectral signature depends on its material, size, and shape. Learning these signatures with data-driven models enables detection and class-level recognition even when imaging is impractical

(e.g., restricted areas, poor visibility) or undesirable (privacy/security). In this work, we investigate OR using backscattered measurements in the mmWave band and a 1D CNN classifier trained on frequency-domain responses.

### 4.4.1 Experimental Framework

**Measurement Equipment** The setup as shown in Fig. 4.43 uses a portable Signal Generator (SG) and a Spectrum Analyzer (SA) [137] with conical horn antennas, all covering 26–40 GHz: (i) SG (SAF Tehnika J0SSAG14), up to 5 dBm radiated power; (ii) SA (SAF Tehnika J0SSAP14), sensitivity $\approx -100$ dBm and 0.5 s sweep time at 100 MHz span; (iii) horns with 20.5–21.5 dBi gain, $\sim$13.5° HPBW, vertical polarization.




(a) Spectrum Analyzer. (b) Horn Antenna (c) Signal Generator

**Figure 4.43:** Measurement Equipment

**Object Classes** Two everyday object classes were considered, each comprising multiple distinct physical instances to make the recognition task realistic (see Fig. 4.44):

- **Ceramic Mug (CM):** 18 different mugs, each measured 3 times $\Rightarrow$ 54 frequency responses.
- **Screw Box (SB):** a cardboard box filled with screws, re-deployed into 18 distinct realizations, each measured 3 times $\Rightarrow$ 54 frequency responses.
- **No Object (NO):** 27 measurements with no target present.

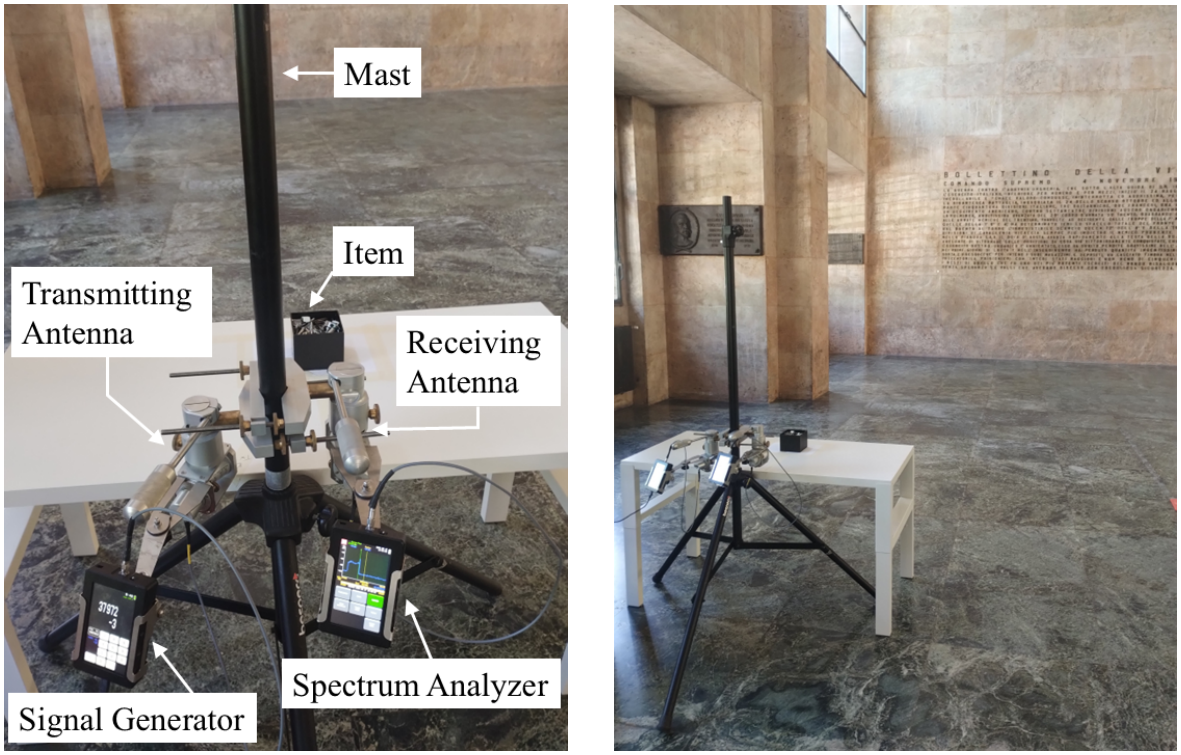
These choices ensure multiple exemplars per class and include a background (NO) condition for robust detection/recognition.

**Measurement Procedure and Data Collection** The measurement setup is shown in Fig. 4.45a, where the transmitting and the receiving antennas, held up on the same mast together with the SG and the SA, are pointed toward the item placed on a


**Figure 4.44:** Considered Items

wooden table in front of them at a distance of about 40 cm. The signal from the SG feeding the transmitting antenna had a power equal to -3dBm and a frequency sweeping the 1 GHz band from 37.5 GHz to 38.5 GHz, which sounds fairly consistent with the items size according to previous considerations, with 40 MHz increments. The radiated signal impinges on the object and triggers a backscattered wave that is then detected by the receiving antenna. The SA finally records the corresponding received signal strength (RSS) for each frequency sample. Since the transmitter and the receiver are not synchronized, the time the SG dwells on each frequency was set equal to the sweeping time of the SA over the 1 GHz bandwidth, that means a new RSS value was achieved at every sweep round. Therefore, the time required to collect the whole frequency response from a single item amounted to 125 sec. Synchronization between the transmitting and the receiving stage (e.g. by using a vectorial network analyzer) would greater reduce the frequency scanning time. In order to limit possible multipath interference from the environment, the measurement activities have been carried out in a large, basically empty hall, where possible sources of additional scattering (besides the target item) were fairly faraway from the equipment (Fig. 4.45b).

Fig. 4.46 shows an example of received frequency response over the target 1 GHz band for the three considered use cases (CM, SB and NO). RSS values collected in presence of an item in front of the antennas (either a CM or a SB) are nearly 20




(a) Measurement setup. (b) Measurement Environment.

**Figure 4.45:** Experimental setup and Environment.

dB greater than the received signal strength without any objects, that means possible (multipath) interference and/or noise coming from the environment were not affecting the measurement to a significant extent. Furthermore, this result also points out that any direct cross-talk between the antennas is either negligible or sufficiently suppressed, further corroborating the physical soundness of the collected measured data.

In order to reduce the sensitivity of the recognition procedure to the distance between the antennas and the item under test, the RSS values collected by the SA have been normalized each time to the maximum value before they have been stored in the final dataset.

### 4.4.2 Learning Model and Performance Metrics

For the classification task, a one-dimensional convolutional neural network (1D CNN) was employed to capture possible frequency-domain dependencies. CNN-based architectures are well suited for extracting robust features from spectral data. The models were trained using the dataset described above, with the normalized frequency responses as input and object class labels (CM, SB, NO) as output.

Performance was evaluated using standard metrics previously introduced, namely accuracy, precision, recall, and F1-score, to ensure consistency in comparing results across different tasks in this thesis.

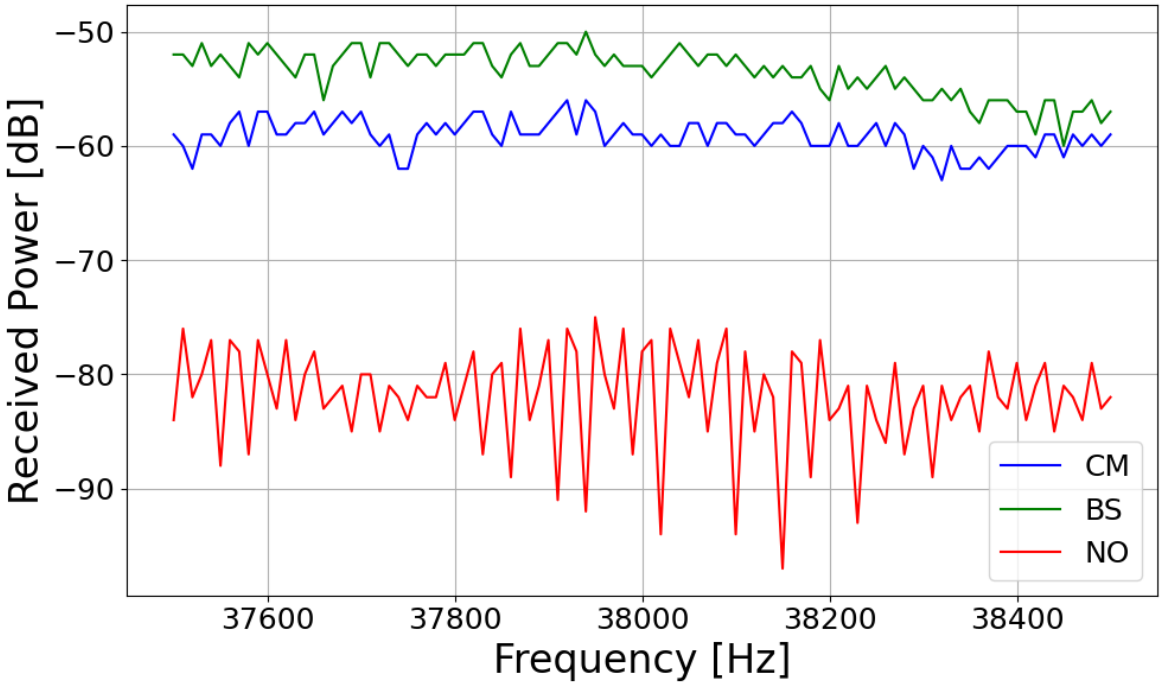



**Figure 4.46:** Example of Measurement Results.

The chosen hyperparameters, reported in Table 4.17, were selected after testing multiple candidate values to balance convergence speed and generalization capability. The network structure included a convolutional layer with ReLU activation, followed by max pooling and two fully connected layers for classification. Early stopping was employed to avoid overfitting, with a patience of 20 epochs.

The dataset of normalized frequency responses was split into training (80%) and testing (20%), ensuring class balance across the splits. The trained 1D CNN achieved reliable discrimination among the three object classes (CM, SB, NO), demonstrating the feasibility of learning representative features directly from frequency-domain backscattering patterns.

### 4.4.3 Results and Discussion

Auto- and cross-correlation analysis was first carried out on the raw dataset to investigate the similarity between the frequency responses associated with different classes. The Pearson correlation coefficient $\rho\,(s(f), v(f))$ was computed for each pair of responses $s(f), v(f)$, and then averaged within the same class (auto-correlation) or across different classes (cross-correlation).

The results, shown in Fig. 4.47, confirm that auto-correlation values are consistently higher than cross-correlation ones. This indicates that backscattered frequency responses exhibit greater similarity within the same class compared to different classes, which is consistent with expectations based on the electromagnetic properties of the objects. Nonetheless, correlation levels vary depending on the specific pair of classes

**Table 4.17:** 1D CNN Hyperparameters.

| Description | Value | Note |
|---|---|---|
| # Kernels | 32 | Convolutional layer |
| Kernel size | 16 | Convolutional layer |
| Activation function | ReLU | Convolutional layer |
| Type of pooling | Max | Pooling layer |
| Kernel size | 2 | Pooling layer |
| # Neurons | 50 | 1st fully connected layer |
| # Neurons | 2 | 2nd fully connected layer |
| # Epochs | 100 | Candidate: 50, 100, 200 |
| Learning Rate | 0.001 | Candidate: 0.0001, 0.001, 0.01, 0.1 |
| Weight Decay | 0.001 | Candidate: 0.001, 0.01, 0.1 |
| Batch Size | 2 | Candidate: 2, 4, 8 |
| Early Stopping | 20 | Patience (epochs) |

considered, highlighting that some objects (e.g., CM vs. SB) are more easily distinguishable than others.

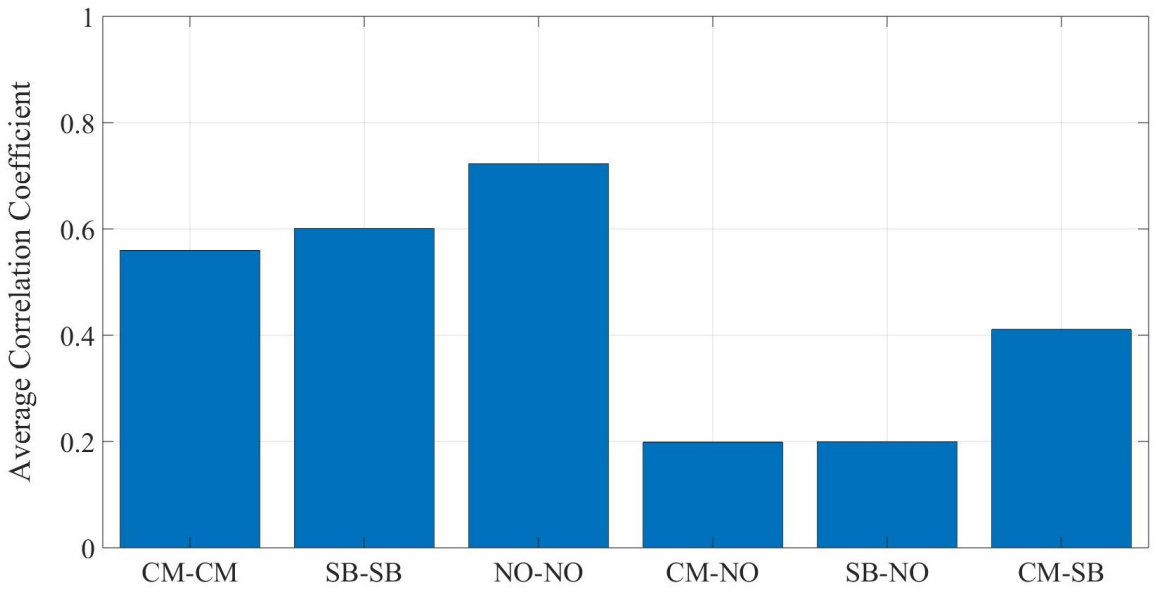


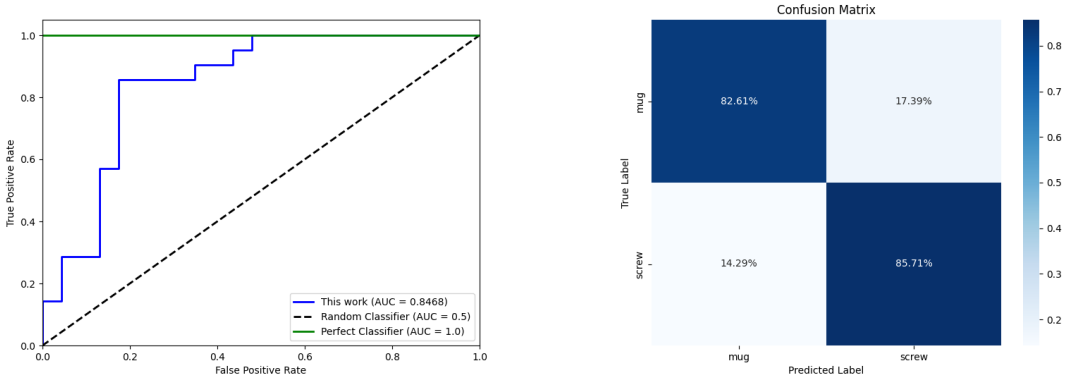

**Figure 4.47:** Average auto- and cross-correlation coefficient across the considered classes of items.

The performance of the 1D CNN model is summarized in Table 4.18, while Fig. 4.48 provides further insights into the recognition task. Results show that detection (object vs. no-object) achieved perfect performance across all metrics, while recognition (differentiating between CM and SB) yielded solid results with an accuracy of 84% and balanced precision, recall, and F1-scores above 80%.

The model can perfectly detect the presence of items in front of the antennas, i.e. it

**Table 4.18:** Performance of the 1D CNN model for detection and recognition.

| Task | Accuracy | Precision | Recall | F1-score | AUC |
|---|---|---|---|---|---|
| Detection | 100% | 100% | 100% | 100% | 100% |
| Recognition | 84% | 82% | 86% | 84% | 85% |




(a) AUC curve for recognition.

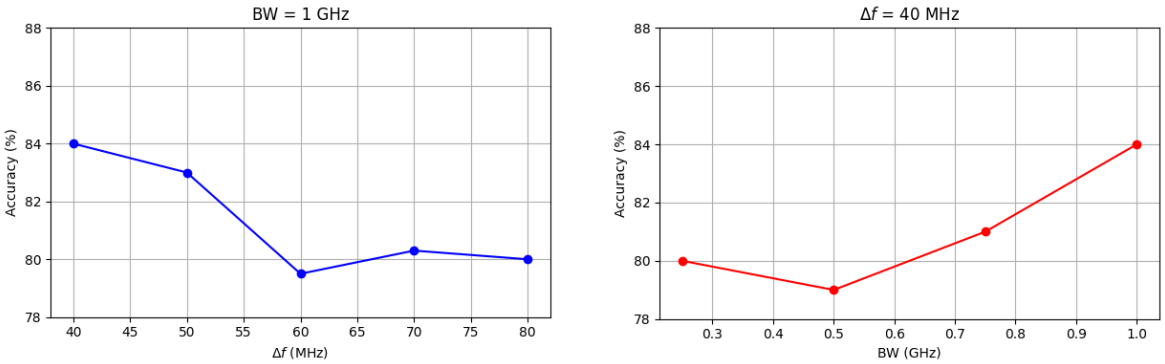
(b) Confusion matrix for recognition.

**Figure 4.48:** Performance metrics for recognition task.

behaves as a perfect detector. According to Fig. 4.47, auto-correlation inside the NO class turns out to be far greater than cross-correlation against both the CB and the SB class. Although a 1D CNN is not just a correlation-based classifier, this clear difference can help explaining the perfect object detection skill.

With reference to OR, the model performance, while strong, got slightly worse, with KPIs undergoing some corresponding reduction (Table 4.18). Overall, the outcome is anyway better than blind (or random) recognition (Fig. 4.48a), which would have KPIs values equal to 0.5. This is also supported by Fig. 4.48a, where the AUC curve of the trained 1D CNN model is clearly closer to that of a perfect classifier rather than to the curve of a blind classifier.

The confusion matrix in Fig. 4.48b shows that performance reduction affects both the CM and the SB classes basically to the same extent. Every time the 1D CNN is fed with the signal trace backscattered by either a ceramic mug or a box filled with screws, it is correctly labeled with a probability equal to about 82% (Precision). At the same time, the output prediction from the neural network (Recall) turn out reliable in the 86% of cases approximately regardless of the class. Once again, Fig. 4.47 can contribute to the interpretation of results. Although the auto-correlation values computed over the CM and the SB classes are still greater than their cross-correlation, the difference is not so large, and this can somehow explain the overall performance decrease in recognition compared to detection. Although a 1D CNN is expected to take into account (much) more features than correlation only - let's stress this aspect again - it is anyway interesting noting that the small difference between the auto-correlation




(a) Impact of $\Delta f$ on test accuracy. (b) Impact of $BW$ on test accuracy.

**Figure 4.49:** Sensitivity analysis: (a) frequency step $\Delta f$, (b) bandwidth $BW$.

of the CM and the SB class in Fig. 4.47 somehow correspond to the slight performance imbalance in Fig. 4.48b in favour of the SB class.

Although the choice of having more than one item per class has represented a harder challenge compared to [138, 139], the final, overall performance is similar, as the test accuracy per class achieved in [138, 139] is in the range [87.3% ÷ 92.7%] and [67.5% ÷ 100%], respectively.

With reference to possible, real-time application of OR, a critical factor is the time required to accomplish the recognition task. Since the decision time of the trained 1D CNN is basically negligible, any possible latency comes from the frequency scanning for the acquisition of the item back-scattered response. In this respect, the importance of synchronization between the transmitting and the receiving stage is clear and has already been stressed. Furthermore, the amplitude of the frequency band under sweep ($BW$) and the scanning frequency step ($\Delta f$) can also affect the final reaction time. Since they can also have an impact on the performance, the sensitivity of the test accuracy to $BW$ and $\Delta f$ is investigated in Fig. 4.49a and 4.49b. Increasing the frequency step from 40 MHz to 80 MHz results in a slight decrease in accuracy, which drops from 84% to approximately 80%. A similar reduction is observed when the bandwidth shrinks from 1 GHz to 250 MHz. It can be also noted that in both pictures the accuracy does not steadily decrease all over the range considered for $\Delta f$ and BW. Rather, the trend gets somehow blurred as $\Delta f$ and $BW$ approach 40 MHz and 250 MHz, respectively. This is explained by the corresponding reduction in the number of frequency samples ($N_{fs}$) collected for each backscattered signal: if $N_{fs} = 25$ when $\Delta f$= 40 MHz and $BW$=1 GHz, it drops to 12 if $\Delta f$= 80 MHz and to 6 if $BW$=250 MHz. A lower number of collected samples generally corresponds to some lost of information that could have helped in recognition, thus leading to the performance reduction on the average. At the same time, each step of reduction in $N_{fs}$ does not automatically bring a performance reduction, as the level of separability between frequency responses does not necessarily linearly decrease with $N_{fs}$.

# Conclusions

This thesis presented a study on the application of ML to electromagnetic and wireless propagation problems, encompassing both regression and classification tasks. The work relied on deterministic simulations or experimental measurements for dataset generation and employed data-driven models to predict key propagation parameters—such as PL, shadowing, fading, and DS—as well as to perform classification tasks including LoS detection and OR through backscattering analysis. In some cases, simple analytical formulations were also developed to complement the ML models, providing interpretable expressions consistent with the physical behavior of the channel. The overarching objective was to design fast, accurate, and generalizable ML frameworks capable of addressing a wide range of electromagnetic modeling and classification challenges across different environments and frequency bands.

The first major contribution focused on the prediction of LoS conditions in both industrial and urban scenarios. In the industrial environment, ML models were first trained on a large dataset generated through RT simulations to classify LoS and NLoS links. The proposed model accurately reproduced the deterministic behavior of LoS probability across a wide range of configurations, validating the reliability of ML-based inference for complex industrial layouts. An analytical expression was then derived from the same dataset, providing a compact and interpretable formulation of LoS probability as a function of key environmental variables. In urban scenarios, LoS prediction was investigated using both numerical-feature-based and image-based approaches. For the cities of Bologna, Munich, and San Francisco, an XGBoost model was employed to estimate LoS probability from geometric and spatial features extracted from city maps. In parallel, an image-based UNet architecture was trained exclusively on Bologna, leveraging paired datasets of map images and RT-derived LoS maps. The UNet successfully learned the spatial correlations between building distributions and propagation behavior, achieving accurate and smooth reconstruction of LoS maps while drastically reducing computational time compared to RT simulations. Overall, both XGBoost and DeepUNet models effectively captured the geometric dependencies of LoS propagation, with the image-based UNet clearly outperforming the tabular-data XGBoost approach in terms of accuracy and spatial consistency, offering a faster and more scalable alternative to traditional analytical or simulation-based methods.

The second contribution focused on the estimation of fundamental channel parameters in industrial environments. Narrowband modeling included the prediction of the PLE,

shadowing standard deviation ($\sigma_{dB}$), and Rice factor ($K$), while wideband modeling analyzed the DS. Extensive RT simulations were conducted over multiple industrial configurations, enabling the development of a combined linear regression–MLP modeling approach that achieved high prediction accuracy, with RMSE values significantly smaller than the corresponding variability range. In parallel, simple analytical expressions were derived to capture the main dependencies on the key environmental features, providing interpretable yet computationally efficient alternatives. Together, the ML and analytical approaches yielded a comprehensive and scalable framework for accurate characterization of both large- and small-scale propagation phenomena in industrial environments.

The third contribution addressed the prediction of REMs in realistic urban scenarios, with a focus on the city of Bologna. A complete dataset was generated using deterministic RT simulations, where each portion of the city map was paired with its corresponding received power coverage map. The main objective was to develop an image-based deep learning framework capable of predicting coverage maps directly from 2D urban layouts. To this end, a DeepUNet architecture was implemented and systematically evaluated under different configurations, including the use of auxiliary inputs such as LoS maps and Hata-like maps, as well as data augmentation strategies to enhance model generalization. The results demonstrated that the proposed DeepUNet effectively learned the spatial dependencies between buildings and signal distribution, achieving high prediction accuracy and visual consistency with the simulated coverage maps. These findings confirmed the capability of image-based DL to accelerate REM generation, offering a powerful tool for data-driven and real-time network planning.

Finally, a fourth contribution explored OR through electromagnetic backscattering in the frequency domain. Using frequency-swept measurements of reflected signals from different objects, a 1D CNN was developed for automatic object classification. The model achieved high accuracy and robustness against variations in frequency resolution and bandwidth, proving that ML techniques can effectively interpret electromagnetic signatures for sensing and recognition applications.

Overall, this thesis demonstrated that ML can serve as a unifying and scalable framework for multiple wireless propagation tasks—ranging from LoS and channel parameter estimation to coverage prediction and electromagnetic OR—bridging the gap between deterministic modeling and real-time prediction.

Future research will extend these methodologies to other channel parameters such as angular and Doppler spreads, investigate the impact of directional antennas in coverage prediction.

# List Of Acronyms

**AI** Artificial Intelligence.

**ANN** Artificial Neural Network.

**AUPRC** Area Under Precision Recall Curve.

**BCE** Binary Cross-Entropy.

**CL** Convolutional Layer.

**CM** Ceramic Mug.

**CNN** Convolutional Neural Network.

**DL** Deep Learning.

**DS** Delay Spread.

**EM** Electromagnetic.

**FCL** Fully Connected Layer.

**FN** False Negative.

**FP** False Positive.

**FPR** False Positive Rate.

**GBDT** Gradient Bossting Decision Tree.

**GO** Geometrical Optics.

**IoT** Internet of Things.

**ISI** Inter Symbol Interference.

**LoS** Line of Sight.

**MH** Machine Height.

**ML** Machine Learning.

**MLP** Multilayer Perceptron.

**MS** Machine Size.

**MSE** Mean Square Error.

**NO** No Object.

**OR** Object Recognition.

**PDF** Probability Density Function.

**PDP** Power Delay Profile.

**PEC** Perfect Electric Conductor.

**PG** Path Gain.

**PL** Path Loss.

**PLa** Pooling Layer.

**PLE** Path Loss Exponent.

**ReLU** Rectified Linear Unit.

**REM** Radio Environmental Map.

**RL** Ray Launching.

**RMS** Root Mean Square.

**ROC** Receiver Operating Characteristic.

**RT** Ray Tracing.

**Rx** Receiving Antenna.

**RxH** Receiver Height.

**SA** Spectrum Analyzer.

**SB** Screw Box.

**SG** Signal Generator.

**SGD** Stochastic Gradient Descent.

**SP** Spacing between machines.

**SSIM** Structural Similarity Index.

**SVMs** Support Vector Machines.

**SVRs** Support Vector Regressors.

**tanh** Hyperbolic Tangent.

**TN** True Negative.

**ToA** Time of Arrival.

**TP** True Positive.

**TPR** True Positive Rate.

**Tx** Transmitting Antenna.

**TxH** Transmitter Height.

**UTD** Uniform Theory of Diffraction.

**XGBoost** eXtreme Gradient Boosting.

# References


[1] Henry L. Bertoni. 'Radio Propagation for Modern Wireless Systems'. In: Upper Saddle River, NJ: Prentice Hall, 1999. ISBN: 978-0130811585.

[2] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2002. ISBN: 978-0130422323.

[3] Andrea Goldsmith. *Wireless Communications*. Cambridge, UK: Cambridge University Press, 2005. ISBN: 978-0521837163.

[4] Andreas F. Molisch. *Wireless Communications*. 2nd ed. Chichester, UK: Wiley, 2011. ISBN: 978-0470741873.

[5] H.T. Friis. 'A Note on a Simple Transmission Formula'. In: *Proceedings of the IRE* 34.5 (1946), pp. 254–256. DOI: 10.1109/JRPROC.1946.234568.

[6] Simon R. Saunders and Alejandro Aragón-Zavala. *Antennas and Propagation for Wireless Communication Systems*. 2nd ed. Chichester, UK: Wiley, 2007. ISBN: 978-0470848794.

[7] J. D. Parsons. *The Mobile Radio Propagation Channel*. 2nd ed. Chichester, UK: Wiley, 2000. ISBN: 978-0471498790.

[8] Gordon L. Stuber. *Principles of Mobile Communication*. 4th ed. Cham, Switzerland: Springer, 2017. ISBN: 978-3319515898.

[9] R. H. Clarke. 'A Statistical Theory of Mobile-Radio Reception'. In: *The Bell System Technical Journal* 47.6 (1968), pp. 957–1000. DOI: 10.1002/j.1538-7305.1968.tb00070.x.

[10] William C. Jakes. *Microwave Mobile Communications*. Reprinted by IEEE Press, 1994. New York, NY: Wiley, 1974. ISBN: 978-0780311452.

[11] S. O. Rice. 'Mathematical Analysis of Random Noise, Part II'. In: *Bell System Technical Journal* 24.1 (1945), pp. 46–156. DOI: 10.1002/j.1538-7305.1945.tb00453.x.

[12] Christian Doppler. *Über das farbige Licht der Doppelsterne und einiger anderer Gestirne des Himmels*. Prague, Czech Republic: K. Bohm, 1842.

[13] John G. Proakis and Masoud Salehi. *Digital Communications*. 5th ed. New York, NY: McGraw-Hill, 2007. ISBN: 978-0072957167.

[14] Masaharu Hata. 'Empirical Formula for Propagation Loss in Land Mobile Radio Services'. In: *IEEE Transactions on Vehicular Technology* 29.3 (1980), pp. 317–325. DOI: 10.1109/T-VT.1980.23859.

[15] COST Action 231. *Urban Transmission Loss Models for Mobile Radio in the 900 and 1800 MHz Bands*. 1999.

[16] Joseph Walfisch and Henry L. Bertoni. 'A Theoretical Model of UHF Propagation in Urban Environments'. In: *IEEE Transactions on Antennas and Propagation* 36.12 (1988), pp. 1788–1796. DOI: 10.1109/8.14401.

[17] 3GPP. *Study on Channel Model for Frequencies from 0.5 to 100 GHz*. Technical Report TR 38.901 v16.1.0. 2019.

[18] J. Meinilä, P. Kyösti, L. Hentilä et al. *WINNER II Channel Models*. Deliverable D1.1.2 V1.2. 2007.

[19] George R. MacCartney and Theodore Ted S. Rappaport. 'Millimeter-Wave Base Station Diversity for 5G Coordinated Multipoint (CoMP) Applications'. In: *IEEE Transactions on Wireless Communications* 18 (2019), pp. 3395–3410. URL: https://api.semanticscholar.org/CorpusID:181776099.

[20] Franco Fuschini, Enrico M. Vitucci, Marina Barbiroli, Gabriele Falciasecca and Vittorio Degli-Esposti. 'Ray tracing propagation modeling for future small-cell and indoor applications: A review of current techniques'. In: *Radio Science* 50.6 (2015), pp. 469–485.

[21] Vittorio Degli-Esposti. 'A Diffuse Scattering Model for Urban Propagation Prediction'. In: *IEEE Transactions on Antennas and Propagation* 49.7 (2001), pp. 1111–1113. DOI: 10.1109/8.933489.

[22] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th ed. Hoboken, NJ: Pearson, 2020. ISBN: 978-0134610993.

[23] Nils J. Nilsson. *Artificial Intelligence: A New Synthesis*. San Francisco, CA: Morgan Kaufmann, 1998. ISBN: 978-1558604674.

[24] Tom M. Mitchell. *Machine Learning*. New York, NY: McGraw-Hill, 1997. ISBN: 978-0070428072.

[25] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006. ISBN: 978-0387310732.

[26] Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016. ISBN: 978-0262035613.

[27] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 'Deep Learning'. In: *Nature* 521 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[28] Jerome H. Friedman. 'Greedy Function Approximation: A Gradient Boosting Machine'. In: *Annals of Statistics* 29.5 (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451.

[29] Tianqi Chen and Carlos Guestrin. 'XGBoost: A Scalable Tree Boosting System'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[30] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press, 2018. ISBN: 978-0262039246.

[31] Trevor Hastie, Robert Tibshirani and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York, NY: Springer, 2009. ISBN: 978-0387848571.

[32] Ron Kohavi. 'A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection'. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 2. 1995, pp. 1137–1143.

[33] George A. F. Seber and Alan J. Lee. *Linear Regression Analysis*. 2nd ed. Hoboken, NJ: Wiley, 2003. ISBN: 978-0471415404.

[34] Corinna Cortes and Vladimir Vapnik. 'Support-Vector Networks'. In: *Machine Learning* 20.3 (1995), pp. 273–297. DOI: 10.1007/BF00994018.

[35] J. Ross Quinlan. 'Induction of Decision Trees'. In: *Machine Learning* 1.1 (1986), pp. 81–106. DOI: 10.1007/BF00116251.

[36] Leo Breiman. 'Random Forests'. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[37] J. B. MacQueen. 'Some Methods for Classification and Analysis of Multivariate Observations'. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. University of California Press, 1967, pp. 281–297.

[38] Karl Pearson. 'On Lines and Planes of Closest Fit to Systems of Points in Space'. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

[39] Ian T. Jolliffe. *Principal Component Analysis*. 2nd ed. Springer, 2016. DOI: 10.1007/978-1-4899-7512-7.

[40] Laurens van der Maaten and Geoffrey Hinton. 'Visualizing Data using t-SNE'. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[41] L. McInnes, J. Healy, N. Saul and L. Grossberger. 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'. In: *arXiv preprint arXiv:1802.03426* (2018).

[42] Jürgen Schmidhuber. 'Deep Learning in Neural Networks: An Overview'. In: *Neural Networks* 61 (2015), pp. 85–117. DOI: 10.1016/j.neunet.2014.09.003.

[43] Warren S. McCulloch and Walter Pitts. 'A Logical Calculus of the Ideas Immanent in Nervous Activity'. In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133. DOI: 10.1007/BF02478259.

[44] Frank Rosenblatt. 'The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain'. In: *Psychological Review* 65.6 (1958), pp. 386–408. DOI: 10.1037/h0042519.

[45] Simon Haykin. *Neural Networks and Learning Machines*. 3rd ed. Pearson, 2009. ISBN: 978-0131471399.

[46] Vinod Nair and Geoffrey E. Hinton. 'Rectified Linear Units Improve Restricted Boltzmann Machines'. In: *Proceedings of the 27th International Conference on Machine Learning (ICML)*. 2010, pp. 807–814.

[47] Xavier Glorot, Antoine Bordes and Yoshua Bengio. 'Deep Sparse Rectifier Neural Networks'. In: *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2011, pp. 315–323.

[48] David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams. 'Learning Representations by Back-Propagating Errors'. In: *Nature* 323 (1986), pp. 533–536. DOI: 10.1038/323533a0.

[49] Leon Bottou. 'Stochastic Gradient Descent Tricks'. In: *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 421–436. DOI: 10.1007/978-3-642-35289-8_25.

[50] Diederik P. Kingma and Jimmy Lei Ba. 'Adam: A Method for Stochastic Optimization'. In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. 2017. URL: https://arxiv.org/abs/1412.6980.

[51] Kurt Hornik, Maxwell Stinchcombe and Halbert White. 'Multilayer Feedforward Networks Are Universal Approximators'. In: *Neural Networks* 2.5 (1989), pp. 359–366. DOI: 10.1016/0893-6080(89)90020-8.

[52] Yann LeCun, Léon Bottou, Yoshua Bengio and Patrick Haffner. 'Gradient-Based Learning Applied to Document Recognition'. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[53] Serkan Kiranyaz, Turker Ince and Moncef Gabbouj. 'Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks'. In: *IEEE Transactions on Biomedical Engineering* 63.3 (2016), pp. 664–675. DOI: 10.1109/TBME.2015.2468589.

[54] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mahmoud Hussein, Moncef Gabbouj and Daniel J. Inman. 'A Review of Vibration-Based Damage Detection in Civil Structures: From Traditional Methods to Machine Learning and

Deep Learning Applications'. In: *Mechanical Systems and Signal Processing* 147 (2021), p. 107077. DOI: 10.1016/j.ymssp.2020.107077.

[55] Osama Abdeljaber, Onur Avci, Serkan Kiranyaz, Moncef Gabbouj and Daniel J. Inman. 'Real-Time Vibration-Based Structural Damage Detection Using One-Dimensional Convolutional Neural Networks'. In: *Journal of Sound and Vibration* 388 (2017), pp. 154–170. DOI: 10.1016/j.jsv.2016.10.043.

[56] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj and Daniel J. Inman. '1D Convolutional Neural Networks and Applications: A Survey'. In: *Mechanical Systems and Signal Processing* 151 (2021), p. 107398. DOI: 10.1016/j.ymssp.2020.107398.

[57] Wenjing Zhang, Chenglong Li, Guoqiang Peng and Rui Gao. 'A Deep Convolutional Neural Network with New Training Methods for Bearing Fault Diagnosis under Noisy Environment and Different Working Load'. In: *Mechanical Systems and Signal Processing* 100 (2018), pp. 439–453. DOI: 10.1016/j.ymssp.2017.06.022.

[58] Sepp Hochreiter. 'The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions'. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.2 (1998), pp. 107–116. DOI: 10.1142/S0218488598000094.

[59] Olaf Ronneberger, Philipp Fischer and Thomas Brox. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[60] James Bergstra and Yoshua Bengio. 'Random Search for Hyper-Parameter Optimization'. In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305.

[61] Jasper Snoek, Hugo Larochelle and Ryan P. Adams. 'Practical Bayesian Optimization of Machine Learning Algorithms'. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Vol. 2. 2012, pp. 2951–2959.

[62] Anqi Mao, mehryar Mohri and Yutao Zhong. 'Cross-Entropy Loss Functions: Theoretical Analysis and Applications'. In: *arXiv preprint arXiv:2304.07288* (2023).

[63] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh and Eero P. Simoncelli. 'Image Quality Assessment: From Error Visibility to Structural Similarity'. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[64] Chen Huang, Ruisi He, Bo Ai, Andreas F. Molisch, Buon Kiong Lau, Katsuyuki Haneda, Bo Liu, Cheng-Xiang Wang, Mi Yang, Claude Oestges and Zhang-

dui Zhong. 'Artificial Intelligence Enabled Radio Propagation for Communications—Part I: Channel Characterization and Antenna-Channel Optimization'. In: *IEEE Transactions on Antennas and Propagation* 70.6 (2022), pp. 3939–3954. DOI: 10.1109/TAP.2022.3149663.

[65] Chen Huang, Ruisi He, Bo Ai, Andreas F. Molisch, Buon Kiong Lau, Katsuyuki Haneda, Bo Liu, Cheng-Xiang Wang, Mi Yang, Claude Oestges and Zhangdui Zhong. 'Artificial Intelligence Enabled Radio Propagation for Communications—Part II: Scenario Identification and Channel Modeling'. In: *IEEE Transactions on Antennas and Propagation* 70.6 (2022), pp. 3955–3969. DOI: 10.1109/TAP.2022.3149665.

[66] Aristeidis Seretis and Costas D. Sarris. 'An Overview of Machine Learning Techniques for Radio Wave Propagation Modeling'. In: *IEEE Transactions on Antennas and Propagation* 70.6 (2022), pp. 3970–3985. DOI: 10.1109/TAP.2021.3098616.

[67] Y.-H. Cho and J.-J. Kim. 'Line-of-Sight MIMO Channel in Millimeter-Wave Beamforming System: Modeling and Prototype Results'. In: *IEEE 81st Vehicular Technology Conference (VTC Spring)*. Glasgow, UK, 2015, pp. 1–5. DOI: 10.1109/VTCSpring.2015.7145912.

[68] A. Maltsev, V. Erceg, A. Pudeyev, A. Lomayev and A. Khoryaev. 'Experimental Investigations of 60 GHz WLAN Systems in an Office Environment'. In: *IEEE Journal on Selected Areas in Communications*. 2009, pp. 1488–1499. DOI: 10.1109/JSAC.2009.091018.

[69] Alexander Shurakov, Polina Rozhkova, Abdukodir Khakimov, Evgeny Mokrov, Anatoliy Prikhodko, Vyacheslav Begishev, Yevgeni Koucheryavy, Mikhail Komarov and Gregory Gol'tsman. 'Dynamic Blockage in Indoor Reflection-Aided Sub-Terahertz Wireless Communications'. In: *IEEE Access* 11 (2023), pp. 134677–134689. DOI: 10.1109/ACCESS.2023.3337050.

[70] M. K. Samimi and T. S. Rappaport. '3-D Millimeter-Wave Statistical Channel Model for 5G Wireless System Design'. In: *IEEE Transactions on Microwave Theory and Techniques* 64.7 (July 2016), pp. 2207–2225. DOI: 10.1109/TMTT.2016.2574851.

[71] A. A. M. Saleh and R. A. Valenzuela. 'A Statistical Model for Indoor Multipath Propagation'. In: *IEEE Journal on Selected Areas in Communications* 5.2 (Feb. 1987), pp. 128–137. DOI: 10.1109/JSAC.1987.1146527.

[72] D. Solomitckii, A. Orsino, S. Andreev, Y. Koucheryavy and M. Valkama. 'Characterization of mmWave Channel Properties at 28 and 60 GHz in Factory Automation Deployments'. In: *IEEE Wireless Communications and Networking Conference (WCNC)*. Barcelona, Spain, 2018, pp. 1–6. DOI: 10.1109/WCNC.2018.8377337.

[73] M. Mahdi Azari et al. 'Evolution of Non-Terrestrial Networks From 5G to 6G: A Survey'. In: *IEEE Communications Surveys and Tutorials* 24.4 (2022), pp. 2633–2672. DOI: 10.1109/COMST.2022.3199901.

[74] Daniel Fabian Külzer, Sławomir Stańczak and Mladen Botsov. 'CDI Maps: Dynamic Estimation of the Radio Environment for Predictive Resource Allocation'. In: *Proc. IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2021, pp. 892–898. DOI: 10.1109/PIMRC50174.2021.9569310.

[75] Shamik Sarkar, Aniqua Baset, Harsimran Singh, Phillip Smith, Neal Patwari, Sneha Kasera, Kurt Derr and Samuel Ramirez. 'LLOCUS: Learning-based Localization Using Crowdsourcing'. In: *Proc. Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 2020, pp. 201–210.

[76] Enes Krijestorac, Samer Hanna and Danijela Cabric. 'Spatial Signal Strength Prediction using 3D Maps and Deep Learning'. In: *ICC 2021 - IEEE International Conference on Communications*. 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500970.

[77] S. Mignardi, M. J. Arpaio, C. Buratti, E. M. Vitucci, F. Fuschini and R. Verdone. 'Performance Evaluation of UAV-Aided Mobile Networks by Means of Ray Launching Generated REMs'. In: *Proc. 2020 30th Int. Telecommun. Netw. and Appl. Conf. (ITNAC)*. Melbourne, VIC, Australia, 2020, pp. 1–6. DOI: 10.1109/ITNAC50341.2020.9315177.

[78] Zoran Utkovski, Patrick Agostini, Matthias Frey, Igor Bjelakovic and Slawomir Stanczak. 'Learning Radio Maps for Physical-Layer Security in the Radio Access'. In: *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 2019, pp. 1–5. DOI: 10.1109/SPAWC.2019.8815467.

[79] B. P. Nayak, L. Hota, A. Kumar, A. K. Turuk and P. H. J. Chong. 'Autonomous Vehicles: Resource Allocation, Security, and Data Privacy'. In: *IEEE Transactions on Green Communications and Networking* 6.1 (Mar. 2022), pp. 117–131. DOI: 10.1109/TGCN.2021.3110822.

[80] Ron Levie, Çağkan Yapar, Gitta Kutyniok and Giuseppe Caire. 'RadioUNet: Fast Radio Map Estimation With Convolutional Neural Networks'. In: *IEEE Transactions on Wireless Communications* 20.6 (2021), pp. 4001–4015. DOI: 10.1109/TWC.2021.3054977.

[81] Hazem Sallouha, Shamik Sarkar, Enes Krijestorac and Danijela Cabric. 'REM-U-Net: Deep Learning Based Agile REM Prediction With Energy-Efficient Cell-Free Use Case'. In: *IEEE Open Journal of Signal Processing* 5 (2024), pp. 750–765. DOI: 10.1109/OJSP.2024.3378591.

[82] M. Mahanty, D. Bhattacharyya and D. Midhunchakkaravarthy. 'A review on deep learning-based object recognition algorithms'. In: *Machine Intelligence and Soft Computing*. Ed. by D. Bhattacharyya, S.K. Saha and P. Fournier-Viger. Singapore: Springer Nature Singapore, 2022, pp. 53–59.

[83] R. Yurt, H. Torpj, P. Mahouti, A. Kizilay and S. Kozie. 'Buried object characterization using ground penetrating radar assisted by data-driven surrogate models'. In: *IEEE Access* 11 (2023), pp. 13309–13323. DOI: 10.1109/ACCESS.2023.3243132.

[84] N. Manakitsa, G.S. Maraslidis, L. Moysis and G.F. Fragulis. 'A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision'. In: *Technologies* 12 (2024), p. 15. DOI: 10.3390/technologies12020015.

[85] I. Daramouskas, D. Meimetis, N. Patrinopoulou, V. Lappas, V. Kostopoulos and V. Kapoulas. 'Camera-based local and global target detection, tracking, and localization techniques for UAVs'. In: *Machines* 11 (2023), p. 315. DOI: 10.3390/machines11020315.

[86] D. Jain, I. Nailwal, A. Ranjan and S. Mittal. 'Object recognition with voice assistant for visually impaired'. In: *Proceedings of International Conference on Paradigms of Communication, Computing and Data Analytics*. Ed. by A. Yadav, S.J. Nanda and M.H. Lim. Singapore: Springer Nature Singapore, 2023, pp. 537–545.

[87] C.B. Murthy, M.F. Hashmi, N.D. Bokde and Z.W. Geem. 'Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—A comprehensive review'. In: *Applied Sciences* 10 (2020), p. 3280. DOI: 10.3390/app10093280.

[88] A.R. Pathak, M. Pandey and S. Rautaray. 'Deep learning approaches for detecting objects from images: A review'. In: *Progress in Computing, Analytics and Networking*. Ed. by P.K. Pattnaik, S.S. Rautaray, H. Das and J. Nayak. Singapore: Springer, 2018, pp. 491–499. DOI: 10.1007/978-981-10-7871-2_47.

[89] X. Yang, J. Yan and W. Wang. 'Brain-inspired models for visual object recognition: An overview'. In: *Artificial Intelligence Review* 55 (2022). DOI: 10.1007/s10462-021-10130-z.

[90] I. Rodriguez-Conde, C. Campos and F. Fdez-Riverola. 'Optimized convolutional neural network architectures for efficient on-device vision-based object detection'. In: *Neural Computing and Applications* 34 (2022). DOI: 10.1007/s00521-021-06830-w.

[91] F.J. Abdu, Y. Zhang, M. Fu, Y. Li and Z. Deng. 'Application of deep learning on millimeter-wave radar signals: A review'. In: *Sensors* 21 (2021), p. 1951. DOI: 10.3390/s21061951.

[92] N. Kalbo, Y. Mirsky, A. Shabtai and Y. Elovici. 'The security of IP-based video surveillance systems'. In: *Sensors* 20 (2020), p. 4806. DOI: 10.3390/s20174806.

[93] R. Kokaly, R. Clark, G. Swayze, K. Livo, T. Hoefen, N. Pearson, R. Wise, W. Benzel, H. Lowers and R. Driscoll. *USGS Spectral Library Version 7*. Data Series 1035. U.S. Geological Survey, 2017. DOI: 10.3133/ds1035.

[94] I. Křivánek. 'Dielectric properties of materials at microwave frequencies'. In: *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* 56 (2008), pp. 125–131. DOI: 10.11118/actaun200856050125.

[95] M. Bennamoun and M.J. Mamic. 'Object recognition—Fundamentals and case studies'. In: *Object Recognition*. Ed. by S. Singh. London, UK: Springer, 2002, pp. 19–57. DOI: 10.1007/978-1-4471-3722-1.

[96] R.T. Chin and C.R. Dyer. 'Model-based recognition in robot vision'. In: *ACM Computing Surveys* 18 (1986), pp. 67–108. DOI: 10.1145/6462.6464.

[97] R. Salman, T. Schultze, M. Janson, W. Wiesbeck and I. Willms. 'Robust radar UWB object recognition'. In: *Proceedings of the 2008 IEEE International RF and Microwave Conference*. Kuala Lumpur, Malaysia, 2008. DOI: 10.1109/RFM.2008.4897451.

[98] R. Salman and I. Willms. 'A novel UWB radar super-resolution object recognition approach for complex edged objects'. In: *Proceedings of the 2010 IEEE International Conference on Ultra-Wideband*. Nanjing, China, 2010. DOI: 10.1109/ICUWB.2010.5616741.

[99] R.G. Kouyoumjian and P.H. Pathak. 'A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface'. In: *IEEE* (1974).

[100] Joseph B. Keller. 'Geometrical Theory of Diffraction'. In: *Journal of the Optical Society of America* 52.2 (1962), pp. 116–130. DOI: 10.1364/JOSA.52.000116.

[101] R. Matschek. 'A geometrical optics and uniform theory of diffraction based ray tracing optimization'. In: *Comptes Rendus Physique* (2005).

[102] S. Hussain and C. Brennan. 'An efficient ray tracing method for propagation prediction along a mobile route in urban environments'. In: *Radio Science* 52.7 (2017), pp. 862–873. DOI: 10.1002/2017RS006275.

[103] M.H. Zadeh, E. M. Vitucci, F. Fuschini, M. Barbiroli and V. Degli-Esposti. 'Line of Sight Detection in Industrial Environment: A Machine Learning Approach'. In: *Proceedings of the IEEE Mediterranean Communication and Computer Networking Conference (MeditCom)*. 2024.

[104] Enrico Maria Vitucci, Vittorio Degli-Esposti, Franco Fuschini, Jonathan S. Lu, Marina Barbiroli, Jeffrey N. Wu, Marco Zoli, J. J. Zhu and Henry L. Bertoni. 'Ray Tracing RF Field Prediction: An Unforgiving Validation'. In: *International Journal of Antennas and Propagation* 2015 (2015), pp. 1–11.

[105] ITU-R. 'Guidelines for the evaluation of radio interface technologies for IMT-Advanced'. In: *Report ITU-R M.2135-1* (2009).

[106] S. Del Prete, N. Di Cicco, M. H. Zadeh, F. Fuschini, M. Barbiroli, V. Degli-Esposti and E. M. Vitucci. 'A Study on Line-of-Sight Prediction in Urban Manhattan-like Environments Using an Optimized Machine Learning Algorithm'. In: *Proceedings of the URSI GASS 2023*. Sapporo, Japan, Aug. 2023.

[107] M.H. Zadeh, M. Barbiroli and F. Fuschini. 'A Simple Empirical Expression for Line of Sight Probability in Industrial Environment'. In: *IEEE Antennas and Wireless Propagation Letters* (2024). DOI: 10.1109/LAWP.2024.XXXXXXX.

[108] M.H. Zadeh, M. Barbiroli, E.M. Vitucci, F. Fuschini, V. Degli Esposti and S. Del Prete. 'Site-Specific Machine Learning Approach for Line-of-Sight Detection'. In: *Proceedings of the 2023 IEEE Antennas and Propagation in Wireless Communications (APWC)*. Venice, Italy, 2023, pp. 1–4.

[109] M.H. Zadeh, M. Barbiroli and F. Fuschini. 'Deep Learning Approach to Line of Sight Detection in Urban Environments'. In: *Proceedings of the 2024 IEEE International Conference on Electromagnetics in Advanced Applications (ICEAA)*. Palermo, Italy, 2025, pp. 1–6.

[110] Carole Helene Sudre et al. 'Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations'. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA*. Springer, 2017, pp. 240–248. DOI: 10.1007/978-3-319-67558-9_28.

[111] M.H. Zadeh, M. Barbiroli and F. Fuschini. 'A Machine Learning Approach to Wireless Propagation Modeling in Industrial Environment'. In: *IEEE Open Journal of Antennas and Propagation* 5.3 (2024), pp. 727–738. DOI: 10.1109/OJAP.2024.3391835.

[112] F. Zadeh, F. Fuschini, F. Vitucci and V. Degli-Esposti. 'Fast-Fading Modeling in Wireless Industrial Communications'. In: *Sensors* (2025).

[113] S. Vitturi, F. Tramarin and L. Seno. 'Industrial wireless networks: the significance of timeliness in communication systems'. In: *IEEE Industrial Electronics Magazine* 7.2 (2013), pp. 40–51. DOI: 10.1109/MIE.2013.2253837.

[114] M. Cheffena. 'Propagation channel characteristics of industrial wireless sensor networks'. In: *IEEE Antennas and Propagation Magazine* 58.1 (2016), pp. 66–73. DOI: 10.1109/MAP.2016.2517501.

[115] P. Stenumgaard, J. Chilo, J. Ferrer-Coll and P. Angskog. 'Challenges and conditions for wireless machine-to-machine communications in industrial environments'. In: *IEEE Communications Magazine* 51.6 (2013), pp. 66–73. DOI: 10.1109/MCOM.2013.6525614.

[116] Emmeric Tanghe, Wout Joseph, Leen Verloock, Luc Martens, Henk Capoen, Kobe Van Herwegen and Wim Vantomme. 'The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz'. In: *IEEE Transactions on Wireless Communications* 7.7 (July 2008), pp. 2740–2751. DOI: 10.1109/TWC.2008.070131.

[117] J. Karedal, S. Wyne, P. Almers, F. Tufvesson and A. F. Molisch. 'A measurement-based statistical model for industrial ultra-wideband channels'. In: *IEEE Transactions on Wireless Communications* 6.8 (Aug. 2007), pp. 3028–3037. DOI: 10.1109/TWC.2007.051050.

[118] Y. Wang, Y. Lv, X. Yin and J. Duan. 'Measurement-based experimental statistical modeling of propagation channel in industrial IoT scenario'. In: *Radio Science* 55.9 (2020), pp. 1–14. DOI: 10.1029/2019RS007013.

[119] Abien Fred Agarap. 'Deep Learning using Rectified Linear Units (ReLU)'. In: *arXiv preprint arXiv:1803.08375* (2018).

[120] A. S. Berahas, M. Jahani, P. Richtárik and M. Takáč. 'Quasi-Newton methods for machine learning: forget the past, just sample'. In: *Optimization Methods and Software* (2021), pp. 1668–1704. DOI: 10.1080/10556788.2021.1977806.

[121] Gerald J. Hahn. 'The Hazards of Extrapolation in Regression Analysis'. In: *Journal of Quality Technology* 9.4 (1977), pp. 159–165. DOI: 10.1080/00224065.1977.11980791.

[122] B. B. Cebecioglu et al. 'Sub-6 GHz Channel Modeling and Evaluation in Indoor Industrial Environments'. In: *IEEE Access* 10 (2022), pp. 127742–127753. DOI: 10.1109/ACCESS.2022.3227052.

[123] T. S. Rappaport and C. D. McGillem. 'UHF Fading in Factories'. In: *IEEE Journal on Selected Areas in Communications* 7.1 (1989), pp. 40–48. DOI: 10.1109/49.16842.

[124] B. Holfeld et al. 'Radio Channel Characterization at 5.85 GHz for Wireless M2M Communication of Industrial Robots'. In: *IEEE Wireless Communications and Networking Conference (WCNC)*. Doha, Qatar, 2016. DOI: 10.1109/WCNC.2016.7564890.

[125] A. Al-Saman, M. Mohamed, M. Cheffena and A. Moldsvor. 'Wideband Channel Characterization for 6G Networks in Industrial Environments'. In: *Sensors* 21.6 (2015).

[126] N. Bjorsell S. R. Panigrahi and M. Bengtsson. 'Power Delay Profile Investigation in Industrial Indoor Environments at the 24 GHz ISM Band'. In: *IEEE International Conference on Industrial Technology (ICIT)*. Shanghai, China, 2022. DOI: 10.1109/ICIT48603.2022.10002732.

[127] D. Dupleich, R. Mueller, M. Landmann, J. Luo, G. Del Galdo and R. S. Thomae. 'Multi-Band Characterization of Propagation in Industry Scenarios'. In: *14th European Conference on Antennas and Propagation (EuCAP)*. Copenhagen, Denmark, 2020. DOI: 10.23919/EuCAP48036.2020.9135630.

[128] M.H. Zadeh, M. Barbiroli, E.M. Vitucci, V. Degli-Esposti and F. Fuschini. 'Wireless Industrial Channel Characterization Through Machine Learning and Ray Tracing Simulations'. In: *Proceedings of the 2025 European Conference on Antennas and Propagation (EuCAP)*. Glasgow, UK, 2025, pp. 1–5.

[129] F. Zadeh, E. M. Vitucci, V. Degli-Esposti and F. Fuschini. 'DeepUNet for Radio Frequency Coverage Map Prediction'. In: *IEEE Transactions on Antennas and Propagation* (2025). Under Review.

[130] V. V. Ratnam et al. 'FadeNet: Deep Learning-Based mm-Wave Large-Scale Channel Fading Prediction and its Applications'. In: *IEEE Access* 9 (2021), pp. 3278–3290. DOI: 10.1109/ACCESS.2020.3048583.

[131] J. -H. Lee, O. G. Serbetci, D. P. Selvam and A. F. Molisch. 'PMNet: Robust Pathloss Map Prediction via Supervised Learning'. In: *Proc. IEEE Global Communications Conference (GLOBECOM)*. 2023, pp. 4601–4606. DOI: 10.1109/GLOBECOM54140.2023.10437562.

[132] S. Zhang, A. Wijesinghe and Z. Ding. 'RME-GAN: A Learning Framework for Radio Map Estimation Based on Conditional Generative Adversarial Network'. In: *IEEE Internet of Things Journal* 10.20 (2023), pp. 18016–18027. DOI: 10.1109/JIOT.2023.3278235.

[133] Qi Chen, Ming Huang and JingJing Yang. 'A Distant-range Content Interaction Network for Radio Map Construction'. In: *ICT Express* 10.5 (2024), pp. 1145–1150. DOI: 10.1016/j.icte.2024.08.008.

[134] Harshit Agrawal, Ari Hietanen and Simo Särkkä. 'Utilizing U-Net architectures with auxiliary information for scatter correction in CBCT across different field-of-view settings'. In: *Proc. SPIE*. 2024. DOI: 10.1117/12.3004168.

[135] S. Bakirtzis, C. Yapar, M. Fiore, J. Zhang and I. Wassell. 'Empowering Wireless Network Applications with Deep Learning-based Radio Propagation Models'. In: *arXiv preprint arXiv:2408.12193* (2024). DOI: 10.48550/arXiv.2408.12193.

[136] M. H. Zadeh, M. Barbiroli, S. Del Prete and F. Fuschini. '1D Convolutional Neural Network for Object Recognition Through Electromagnetic Backscattering in the Frequency Domain'. In: *MDPI Sensors* (2025). Submitted for publication.

[137] SAF Tehnika. *Spectrum Compact Device*. https://spectrumcompact.com/. Accessed: 9 September 2025.

[138] V. Erdélyi, H. Rizk, H. Yamaguchi and T. Higashino. 'Learn to see: A microwave-based object recognition system using learning techniques'. In: *Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking (ICDCN '21)*. Nara, Japan: ACM, 2021, pp. 145–150. DOI: 10.1145/3427477.3429459.

[139] K. Pathel, K. Rambach, T. Visentin, D. Rusev, M. Pfeiffer and B. Yang. 'Deep learning-based object classification on automotive radar spectra'. In: *Proceedings of the IEEE Radar Conference*. Boston, MA, USA, 2019, pp. 1–6. DOI: 10.1109/RADAR.2019.8835775.

[140] Marina Sokolova and Guy Lapalme. 'A systematic analysis of performance measures for classification tasks'. In: *Information Processing and Management* 45.4 (2009), pp. 427–437. DOI: 10.1016/j.ipm.2009.03.002.

[141] Z. Yun and M. F. Iskander. 'Ray tracing for radio propagation modeling: principles and applications'. In: *IEEE Access* 3 (2015), pp. 1089–1100. DOI: 10.1109/ACCESS.2015.2453991.

[142] J. W. McKown and R. L. Hamilton. 'Ray tracing as a design tool for radio networks'. In: *IEEE Network* 5.6 (1991), pp. 27–30. DOI: 10.1109/65.103807.

[143] Daniel Romero and Seung-Jun Kim. 'Radio Map Estimation: A data-driven approach to spectrum cartography'. In: *IEEE Signal Processing Magazine* 39.6 (2022), pp. 53–72. DOI: 10.1109/MSP.2022.3200175.

[144] Mi Yang, Ruisi He, Bo Ai, Chen Huang, Chenlong Wang, Yuxin Zhang and Zhangdui Zhong. 'AI-Enabled Data-Driven Channel Modeling for Future Communications'. In: *IEEE Communications Magazine* 62.4 (2024), pp. 112–118. DOI: 10.1109/MCOM.019.2300072.