Reconstruction of Radio Power Maps with Deep Neural Networks in a Weakly Labeled Learning Context

Aleksandra Malkova

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques et Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Aleksandra MALKOVA

Thèse dirigée par **Massih-Reza AMINI**,Professeur, Université Grenoble Alpes
et co-encadrée par **Benoît DENIS**,Ingénieur de recherche
et **Christophe VILLIEN**

préparée au sein du **Laboratoire Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Reconstruction de Cartes de Puissance Radio avec des réseaux de neurones profonds dans un contexte d'apprentissage faiblement étiqueté

Reconstruction of Radio Power Maps with Deep Neural Networks in a Weakly Labeled Learning Context

Thèse soutenue publiquement le **8 juillet 2022**,
devant le jury composé de :

Monsieur MASSIH-REZA AMINI
Professeur des Universités, UNIVERSITE GRENOBLE ALPES, Directeur de thèse
Monsieur ANDRZEJ DUDA
Professeur des Universités, GRENOBLE INP, Présidente
Monsieur BADIH GHATTAS
Maître de conférences HDR, AIX-MARSEILLE UNIVERSITE, Examinateur
Madame MARIANNE CLAUSEL
Professeur des Universités, UNIVERSITE DE LORRAINE, Rapporteure
Monsieur HACHEM KADRI
Maître de conférences HDR, AIX-MARSEILLE UNIVERSITE, Rapporteur

Acknowledgements

First of all, I would like to thank to my supervisors Massih-Reza Amini, Benoit Denis and Christophe Villien for always giving me useful suggestions, guidance, for being time-involved and sharing the expertice. I am especially grateful for the experience of connection of two fields and acquisition of the knowledge in both of them.

Then, I would like to extend my gratitude to Hachem Kadri and Marianne Clausel, who kindly agreed for the PhD manuscript revision. I am also grateful to Andrzej Duda and Badih Ghattas for their agreeing to be the part of the jury committee. Special thanks to all the members of APTIKAL team of LIG and my colleagues from the STSF service for their support and willing to help me in the hard times.

Last but not least, I would like to thank to my friends for the encouragement and interest in my work. And this work certainly would not have been possible without the love, care and support of my family.

Abstract

The positioning of connected objects has become a major enabling feature for a variety of applications and services in the Internet of Things (IoT) domain. To avoid prohibitive power consumption and hardware complexity at the IoT nodes to be localized, one alternative to the conventional Global Positioning System (GPS) consists in making opportunistic use of location-dependent radio metrics, out of the sensor data packets that may be transmitted from these end-devices to their serving gateways.

In this thesis, we focus on the Received Signal Strength Indicator (RSSI) radio metrics, which are available by default in Long Range (LoRa) low data rate transmissions, while targeting fingerprinting-based localization for the final application. So as to accurately position LoRa nodes at the city scale through fingerprinting, one first needs to build an accurate prior RSSI map of the radio environment (offline), typically based on a few real field measurements. This prior radio map is subsequently used as a reference (for comparison) during the online localization phase. One main goal of this work is hence to obtain an exhaustive and accurate prior map, given sparse and non-uniformly distributed RSSI measurements, while applying advanced machine learning approaches.

Firstly, we discuss the main general challenges of the map reconstruction itself, in light of our specific fingerprinting-based localization context. In particular, by means of theoretical bounds characterizing the best RSSI-based positioning accuracy in case of both fingerprinting (i.e., considering a prior radio map) or parametric positioning (i.e., considering a link-wise range-dependent power path loss model), we illustrate and discuss the impact of RSSI dynamics as a function of space with respect to the final positioning performance. Further, we also describe and analyze the experimental datasets used in our study, including a dataset related to the Grenoble city area, which is still under extension, while devising metrological and pre-processing aspects. These experimental data have been exploited to feed both the underlying radio models with

realistic parameters, as well as the learning algorithms used for map reconstruction with realistic inputs.

Secondly, we briefly recall the main ideas behind supervised and semi-supervised learning approaches, their main working assumptions, as well as their classical models. In particular, we present a Neural Architecture Search (NAS) approach, which is a recent technique allowing to automatically find a Neural Network (NN) model with an optimized architecture suited to a given problem.

We then cover a first application of this NAS to the RSSI map reconstruction problem, in combination with data augmentation techniques, where only a limited amount of labeled input data is available for learning.

Finally, taking also advantage from having access to side (meta-)information about the local area (e.g., city layout, terrain elevation, gateway location, etc...), we solve out the generalization problem for each city (typically, over gateways). The NAS algorithm is then applied again to find the NN model with the best architecture for each of the supposed settings, depending on the amount of such prior side information. On this occasion, we show for instance that using additional views improves the final accuracy of the RSSI map reconstruction, especially in sub-areas close to the gateways where larger variations of the average received signal power are usually observed (ultimately, with a prominent beneficial impact onto positioning performance accordingly).

Keywords: Learning with weakly labeled data, Neural Networks with optimized architecture, Reconstruction of Radio Power Maps, Radio Fingerprinting

Résumé

La détermination de la position d'objets connectés est devenue une fonctionnalité majeure pour une variété d'applications et de services dans le domaine de l'Internet des objets (`IoT`). Afin d'éviter une consommation d'énergie prohibitive et une complexité matérielle importante au niveau des nœuds `IoT` à localiser, une alternative au système conventionnel de positionnement global par satellites (GPS) consiste à faire usage, de manière opportuniste, de métriques radio dépendantes de la position, en s'appuyant en particulier sur les paquets de données qui peuvent être transmis par ces mêmes noeuds-capteurs à leurs stations de base de rattachement.

Dans cette thèse, nous nous concentrons sur la puissance moyenne reçue, ou RSSI (Received Signal Strength Indicator), qui est une métrique disponible par défaut dans le cadre de communications à longue portée et à faible débit de données de type LoRa (Long Range), tout en ciblant une localisation basée sur la reconnaissance de signatures radio, ou *fingerprinting*, en termes d'application finale. Afin de positionner avec précision les nœuds LoRa à l'échelle d'une ville grâce via ces techniques de *fingerprinting*, il est nécessaire d'établir préalablement une carte RSSI suffisamment précise de l'environnement radio (hors ligne), généralement à partir de quelques mesures réelles collectées sur le terrain. Cette carte radio préalable est ensuite utilisée comme référence (pour comparaison) pendant la phase de localisation temps-réel (en ligne). L'objectif principal de ce travail est donc d'obtenir une carte préalable exhaustive et précise, à partir de mesures RSSI éparses et non-uniformément distribuées, en appliquant des approches avancées d'apprentissage automatique.

Tout d'abord, nous discutons des principaux défis inhérents à la reconstruction de la carte radio elle-même, à la lumière de notre application spécifique de localisation. En particulier, à l'aide de limites théoriques caractérisant la meilleure précision de positionnement atteignable à partir de mesures RSSI dans le cas du *fingerprinting* (c.-à-d., en considérant une carte radio préalable) ou d'un positionnement paramétrique (c.-à-d., en considérant un modèle d'atténuation au niveau d'un lien radio, en fonction de la distance entre émetteur et récepteur), nous illustrons et discutons l'impact de la

dynamique spatiale du RSSI (c.-à-d., en fonction de l'espace 2D) sur les performances finales de positionnement. En outre, nous décrivons et analysons les jeux de données expérimentales utilisés dans le cadre de notre étude, dont un jeu relatif à la zone urbaine de Grenoble, qui est actuellement encore en cours d'extension, tout en revenant sur quelques aspects métrologiques, ainsi que sur les premiers post-traitements appliqués. Ces données expérimentales ont été exploitées pour alimenter à la fois les modèles radio sous-jacents (avec des paramètres réalistes) et les algorithmes de reconstruction de cartes (avec des données d'entrée réalistes pour les étapes d'apprentissage, de validation et de test).

Dans un deuxième temps, nous rappelons brièvement les principales idées qui sous-tendent les approches d'apprentissage supervisé et semi-supervisé, leurs principales hypothèses de travail, ainsi que leurs modèles classiques. En particulier, nous présentons une technique récente dite de *Neural Architecture Search* (NAS), qui permet de trouver automatiquement un modèle de réseau neuronal (NN) avec une architecture optimisée, adaptée à un problème donné.

Nous appliquons ensuite ce NAS à notre problème spécifique de reconstruction de cartes RSSI, en combinaison avec des techniques d'augmentation de données, pour lesquelles seule une quantité limitée de données étiquetées en entrée est nécessaire pour réaliser l'apprentissage.

Enfin, en tirant également profit de l'accès à des (méta-)informations secondaires concernant la zone locale de travail (par exemple, le plan de la ville, l'élévation du terrain, l'emplacement des stations de base, etc...), nous résolvons un problème de généralisation (c.-à-d., entre stations de base "apprises" et "non-apprises") pour chacune des villes testées. L'algorithme NAS est ensuite appliqué à nouveau afin de trouver le modèle NN présentant la meilleure architecture, en fonction de la quantité d'information secondaire disponible. A cette occasion, nous montrons par exemple que l'utilisation de vues supplémentaires en complément des mesures de puissance améliore la précision finale de la reconstruction de la carte RSSI, en particulier dans les sous-zones proches des stations de base, où des variations plus importantes de la puissance moyenne du signal reçu sont généralement observées (avec, en fin de compte, un impact positif présumément plus important en termes de performances de positionnement).

Mots clés : Apprentissage à partir de données faiblement étiquetées, Réseaux de neurones avec architectures optimisées, Reconstruction de cartes de puissance radio, Reconnaissance de signatures radio

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

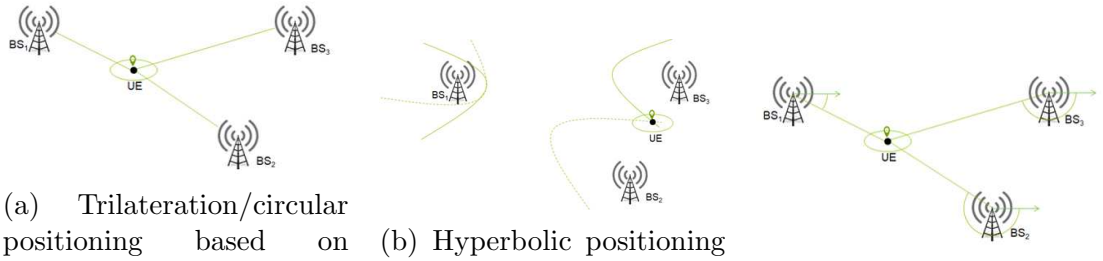
1.1 Context and Motivations

From the Oxford language dictionary, Internet of Things (IoT), refers to the "interconnection via the internet of computing devices embedded in everyday objects, enabling them to send and receive data." In other words, connected devices (or tags) – i.e., objects including sensors (and/or even actuators) – are expected to interact with their physical environment, while collecting sensor data and sending them to a centralized database. The recent integration of low-cost sensor and low-power radio chips in this IoT context has thus been contributing to the fast development of large-scale physical monitoring and crowdsensing systems in various kinds of smart environments (e.g., smart cities, smart homes, smart transportations, smart factories, etc). Beyond, the ability to associate accurate location information with sensor data collected on the field opens appealing perspectives in terms of both location-enabled applications and services (e.g., geo-referenced crowd-sensing, cartography, asset tracking for wide-area logistics...), so that wireless localization has clearly become a key feature of IoT [Khelifi et al., 2019]. For instance, in the environmental research domain, water, pollution, pressure sensors, as well as cameras, microphones and GPS modules of mobile phones, can be used to perform a geo-referenced and multimodal survey of the physical environment [Kamilaris and Ostermann, 2018]. As another example, the fine-grain and dynamic update of air pollution and/or weather maps could benefit from geo-referenced mobile sensing [SAR,] (e.g., aboard taxis, buses, bicycles...), thus continuously complementing the data from static stations.

Among possible localization technologies, the Global Positioning System (GPS) has been widely used in outdoor environments for the last past decades. More recently, low-cost advanced GPS solutions (e.g., RTK, Bi-band...), which can provide

centimeters-level accuracy in open sky conditions, have been made commercially available and are expected to penetrate the mass market, even though mainly for consumer electronics (e.g., smartphones, non-professional drones...). However, they still suffer from high power consumption, what is hardly compliant with the targeted IoT applications. As already mentioned above, the latter indeed most often rely on the massive dissemination of ultra low cost, low complexity, and low consumption communicating elementary nodes or tags (e.g., sensors, actuators) in the environment. Moreover, while using GPS for the geo-tagging of sensor data, localization and communication functionalities would work separately, thus limiting the integration potential and accordingly, increasing both the form factor and the manufacturing cost of IoT devices.

In order to preserve both nodes' low complexity and fairly good localization performances, as an alternative, one can opportunistically measure location-dependent radio metrics, such as the Received Signal Strength (Indicator) (RSSI), Time (Difference) of Arrival (TDoA), Round Trip-Time of Flight (RT-ToF) or Angle of Arrival (AoA), as sensor nodes are communicating with one or several base stations/gateways (i.e., typically, while sending a packet to the infrastructure to push their sensor data to a cloud/server, or to an `IoT` application). Then, based on these radio measurements, various methods can be applied to determine the node position, such as triangulation, trilateration, proximity detection, or fingerprinting [Burghal et al., 2020, Dargie and Poellabauer, 2010, Cheng et al., 2012, Tahat et al., 2016, Yu et al., 2009], see Figures 1.1 and 1.2. We will hereafter focus on the latter technique with the use of RSSI data.



(a) Trilateration/circular positioning based on Round-trip Time of Flight (or power-based ranging)

(b) Hyperbolic positioning based on Time Difference of Arrival

(c) Triangulation based on Angle of Arrival

Figure 1.1: Possible radio-based localization techniques

Among those localization methods, the fingerprinting technique, as one can tell from its name, is similar to the concept of human fingerprints recognition, in the sense it is assumed that a given position of the device on the map can always be associated

with a unique set of radio measurements (i.e., a fingerprint). The fingerprinting technique thus consists in comparing the current fingerprint(s) with an existing database of already known fingerprints associated with their corresponding locations.

Typical fingerprinting methods applied to wireless localization ([Vo and De, 2016, Krisp, 2013]) are based on two main steps, namely *offline* and *online* steps (See Figure 1.2). In the former, an offline radio map (of location-dependent radio metrics) is generated/learnt with respect to each receiving base station, by relying on a collection of geo-referenced measurements (i.e., knowing the locations of the transmitting nodes when measurements are collected). This can be done once for all on the occasion of a preliminary dedicated field measurements campaign, which is usually tedious and time consuming, and/or through crowdsensing (i.e., as new available geo-tagged measurements are incorporated to continuously refine the map). During the latter step, mobile nodes are subsequently localized through patterns recognition, given the prior radio maps and the current sequence of received radio measurements. The monitored area (i.e., the zone where the nodes are expected to be localized a priori) can be typically discretized into pixels.

The offline map can thus be represented by a tabular dataset of presumed positions and their corresponding measurements (e.g., in our case, RSSI readings with respect to each receiving base station). More formally speaking, a typical fingerprinting (FP) database $\mathcal{D}$ consists of a set of signal features $\mathcal{G}$ (in our case, a vector of signal strength measurements for each considered base station), along with the set of locations where these measurements were collected, $\mathcal{L}$, so that $\mathcal{D} = \{\mathcal{G}, \mathcal{L}\}$. $\mathbf{g} = (g_1, g_2, ..., g_v) \in \mathcal{G}$, where v is the length of signal feature vectors in the signal features space $\mathcal{G} \in \mathcal{R}^v$, and the corresponding location $l \in \mathcal{L}$, $\mathcal{L} \in \mathcal{R}^2$. Each element (g, l) of this set $\mathcal{D}$ is unique in the dataset. Accordingly, to solve the positioning problem, the two FP stages can hence be formally summarized as follows:

1. Offline radio map learning out of sparse, non uniform, and site-specific field radio measurements: If D_{sub} represents only a known subset of the fingerprinting database with the corresponding sets of signal measurements $\mathcal{G}_{sub}$ and its locations $\mathcal{L}_{sub}$ (typically, resulting from real field measurements), one may need to complement this subset to have full sets of signal measurements $\mathcal{G}$ and related locations $\mathcal{L}$ in the area of interest, that is, $\hat{\mathcal{G}} = f(\mathcal{L}|(\mathcal{G}_{sub}, \mathcal{L}_{sub}))$, where f is for instance the result of some learning algorithm. Then $\hat{D} = \{\hat{\mathcal{G}}, \mathcal{L}\}$ represents the reconstructed/interpolated map.

2. Online nodes localization through patterns recognition, given the reconstructed a priori radio map $\hat{D}$ and a sequence of current radio measurements collected in an unknown position $\mathbf{g}$: $\hat{l} = \{l \in \mathcal{L} : (\mathbf{g} \times l) \in (\hat{\mathcal{G}} \times \mathcal{L}), \hat{D} = \{\hat{\mathcal{G}}, \mathcal{L}\}\}$.

Ideally, the generated maps should be as complete as possible, so as to cover the whole area of interest for the online positioning step. Unfortunately, one main difficulty of getting such maps lies in the fact that, in real life cases, it is impractical -not to say unfeasible- to collect measurements from every single location of the map. On the contrary, one can usually rely on spatially sparse, non-uniformly distributed, possibly ill-labelled (e.g., wrongly geo-referenced), highly dispersed and time-varying (i.e., with large variations in a given location) and/or asynchronous field measurements. Besides map initialization issues, the continuous update of created maps is also of paramount importance to account for possible environmental changes. In such fingerprinting positioning contexts, the quality of prior map reconstruction is indeed absolutely crucial to achieve high localization accuracy, especially in sub-areas where the radio signal dynamics is the most meaningful from a localization interpretation standpoint. Accordingly, advanced spatial interpolation techniques must be considered to fill the missing regions.

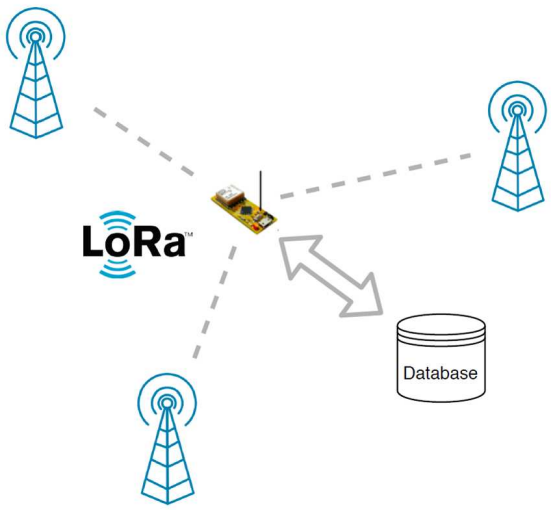


Figure 1.2: A fingerprinting algorithm estimates the current location by matching an online observed radio measurement with a prior radio map known/learnt during a preliminary offline phase.

Another difficulty lies in the possibly large discrepancy/mismatch observed between a priori offline radio maps and the instantaneous online measurements used for real-time localization, due to the cumulative effects of complex factors (incl. both

propagation phenomena and usage/user's behaviour). These challenges will be detailed further in the following sections.

1.2 Overall Research Problem

In the diagram shown in Figure 1.3, we represent two parts of the fingerprinting approach discussed above. The upper branch is dedicated to offline map reconstruction out of given measurements and, possibly, additional meta information. This meta information could consist of binary maps of the buildings [Levie et al., 2020], satellite images [Lin et al., 2020], terrain elevation maps, or any other type of relevant information available in the zone of interest. The bottom branch corresponds to the second step of fingerprinting, namely the online localization step, where for a given set of current measurements associated with the `IoT` node, we need to find the corresponding location.

1.2.1 Fingerprinting with received power maps

Having access to the full map of radio signatures for each base station in the zone of interest, which is stored in the database, it is possible to localize the object on this map. Indeed, once a set of RSSI measurements has been acquired in the occupied position with respect to a set of in-range base stations (in other words, getting a multi-dimensional vector of RSSI measurements), we obtain a so-called fingerprint of this point, which is then compared with the database of known fingerprints, as already shown in Figure 1.2.

The main difficulty thus lies in the reconstruction of these prior radio maps, or, in other words, in the generation of databases containing known signal-location pairs, as detailed further in the following.

1.2.2 Offline RSS received power map reconstruction

As mentioned above, it is complicated to collect the required data in each point on the map. So usually the values of the missing radio metrics (i.e., for unmeasured locations) must be predicted by some algorithms. There are different methods of map reconstruction or interpolation out of given measurements, which will be described in Section 1.3. The quality of the maps issued by these algorithms can be evaluated by the Mean Absolute Error (MAE) or Mean Squared Error (MSE) indicators, calculated over the points that are not considered as inputs in the interpolation process. Classical

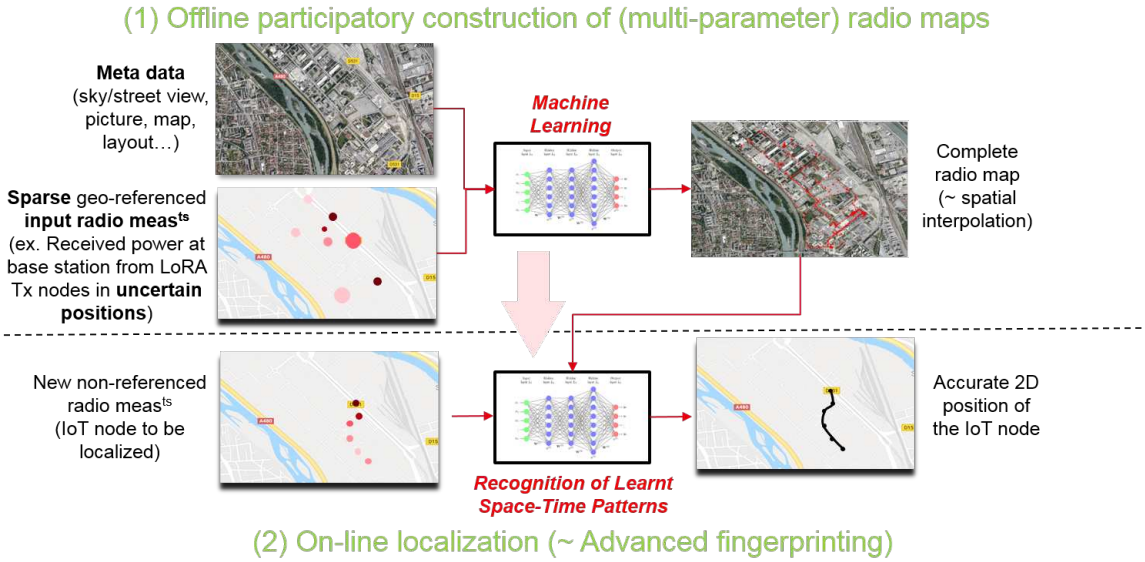


Figure 1.3: Simplified block diagram of a 2-step learning-aided fingerprinting positioning approach based on sparse LoRA RSS measurements and meta data.

mathematical methods do not really take into account the spatial 2D position of the given measurements (but mostly the distance between the pixels), nor the additional meta information that could help to find the underlying distribution of the signal of interest with respect to, e.g., obstacles. Another problem lies in the positions of the available measurements, which may be collected mostly over roads/streets, or with different spatial densities depending on their relative distances to the base stations. So the main goal here is to reconstruct RSS maps out of non-uniform and sparse measurements based on machine learning tools, which could possibly find relevant "hidden" spatial relations (i.e., relations that would not be perceived or accounted with more conventional approaches based on a priori parametric models) and possibly be improved even further by the injection of additional heterogeneous (meta) information besides measurements.

1.3 Related State of the Art

In order to address the map interpolation problem while facing the issues listed in Section 1.1, classical techniques, such as radial basis functions (`RBF`) or kriging [Choi et al., 2018], have already been considered in the context of interest. These approaches are simple and fast, but they are quite weak in predicting the complex and heterogeneous spatial patterns usually observed in real life radio signals (e.g., sudden

and/or highly localized transient variations in the received signal due to specific environmental or topological effects, such as particular building shapes, presence of pieces of public furniture, ultra-narrow streets, crowd activity depending on working hours, etc...). Besides, data augmentation techniques have also been proposed for artificially increasing the number of measurements in such radio map reconstruction problems. The purpose is then to use the generated synthetic data as additional data to train complex models for map interpolation. However, these tools require a very detailed description of the physical environment and can hardly anticipate on its dynamic changes over time. Their high computational complexity is also a major bottleneck.

Hereafter, we detail further related works on `RSSI` map reconstruction, covering both interpolation and data-augmentation techniques, as well as machine learning approaches.

1.3.1 Interpolation and data-augmentation techniques

Various spatial interpolation methods have been proposed for radio map reconstruction in the wireless context.

One first approach, known as kriging or Gaussian process regression [Li and D.Heap, 2011], exploits the distance information between measured points, while trying to capture their underlying spatial dependencies. Another popular method is based on radial basis functions (`RBF`) [Choi et al., 2018, Enrico and Redondi, 2018, Redondi, 2018]. This technique is somehow more flexible, makes fewer assumptions regarding the input data (i.e., considering only the dependency on the distance) and is shown to be more tolerant to some uncertainty [Rusu and Rusu, 2006]. In [Choi et al., 2018] for instance, the authors have divided all the points of a database of outdoor `RSSI` measurements into training and testing subsets, and compared different kernel functions for the interpolation. The two methods above, which rely on underlying statistical properties of the input data (i.e., spatial correlations) and kernel techniques, require a significant amount of input data to provide accurate interpolation results. Accordingly, they are particularly sensitive to sparse initial datasets. They have thus been considered in combination with crowdsensing. In [Liao et al., 2019] for instance, so as to improve the performance of basic kriging, one calls for visiting new positions/cells where the interpolated value is still presumably inaccurate. A quite similar crowdsensing method has also been applied in [Fan et al., 2018] after stating the problem as a matrix completion problem using singular value thresholding, where it is possible to request additional measurements in some specific points where the algorithm has

a low confidence in the interpolated result. In our case though, we assume that we can just rely on a few ground-truth initial RSSI measurements.

Another approach considered in the context of indoor wireless localization relies on both collected field data and an a priori path loss model that accounts for the effect of walls attenuation between the transmitter and the receiver [Kubota et al., 2013]. In outdoor environments, local path loss models (and hence, particularized RSSI distributions) have been used to catch small-scale effects in clusters of measured neighbouring points, instead of using raw RSSI data [Ning and et al., 2016], where in each cluster the appropriate RSSI distribution model is built. However, those parametric path loss models are usually quite inaccurate and require impractical in-site (self-)calibration, thus with very limited generalization capabilities. A quite similar approach, except the use of additional side information, is followed in [Lin et al., 2020], where they propose an algorithm called SateLoc. Based on satellite images, it is then suggested to perform a segmentation of the areas "crossed" by a given radio link, depending on their type (e.g., terrain, water, forest, etc.). Then, proportionally to the size of the crossed region(s), power path loss contributions are computed according to a priori model parameters (i.e., associated with each environment type) and summed up to determine the end-to-end path loss value.

One more way to build or complete radio databases envisaged in the context of fingerprinting based positioning relies on deterministic simulation means, such as Ray-Tracing tools (e.g., [Raspopoulos and et al., 2012, Raspopoulos et al., 2012, Sorour. and et al, 2012, Sorour et al., 2015, Laaraiedh et al., 2012]). The latter aim at predicting in-site radio propagation (i.e., simulating electromagnetic interactions of transmitted radio waves within an environment). Once calibrated with a few real field measurements, such simulation data can relax initial metrology and deployment efforts (i.e., the number of required field measurements) to build an exploitable radio map, or even mitigate practical effects that may be harmful to positioning, such as the cross-device dispersion of radio characteristics (typicaly, between devices used for offline radio map calibration and that used for online positioning). Nevertheless, these tools require a very detailed description of the physical environment (e.g., shape, constituting materials and dielectric properties of obstacles, walls...). Moreover, they usually require high and likely prohibitive computational complexity. Finally, simulations must be re-run again, likely from scratch, each time minor changes are introduced in the environment, e.g. the impact of human activity (like changing crowd density, temporary radio link obstructions).

1.3.2 NN based models trained after data-augmentation

Machine and deep learning approaches have also been recently applied into the RSSI map reconstruction problem. These methods have shown to be able to retrieve unseen spatial patterns with highly localized topological effects and hidden correlations. Until now, to the best of our knowledge, these methods have been mainly trained over simulated datasets generated by data-augmentation approaches.

In [Levie et al., 2020], given a urban environment, the authors introduce a deep neural network called RadioUNet, which outputs radio path loss estimates trained on a large set of generated using the Dominant Path Model data and UNet architecture [Ronneberger et al., 2015b]. To estimate the power spectrum maps in the underlay cognitive radio networks, a two-phase transfer learning GAN (TPTL-GAN) which consists of two stages has been proposed in [Han et al., 2020]. First, it projects the source domain into an adjacent domain (domain projecting (DP) framework). Then, extracted features from the adjacent domain are used to reconstruct the full map in the target domain (domain completing (DC) framework). For training of the DP, complete signal distribution maps have been used. In another contribution, the authors have shown that using the feedforward neural network for path loss modelling could improve the kriging performance [Sato et al., 2019], as conventional parametric path loss models admit a small number of parameters and do not necessarily account for shadowing besides average power attenuation.

Apart from wireless applications, similar problems of map restoration also exist in other domains. In [Zhu et al., 2020] for instance, the authors try to build topographic maps of mountain areas out of sparse measurements of the altitudes. For this purpose, they use a Generative Adversarial Network (GAN) architecture, where in the discriminator they compare pairs of the input data and the so-called "received" map, either generated by the generator or based on the full true map. Another close problem making extensive use of neural networks is the image inpainting problem, where one needs to restore missing pixels in a single partial image, as in [Ulyanov et al., 2017] for example, where they solve the image reconstruction problem given a small amount of input data in one single image. By analogy, this kind of framework could be applied in our context too, by considering the radio map as an image, where each pixel corresponds to the RSSI level for a given node location. Usually, such image inpainting problems can be solved by minimizing a loss between true and predicted pixels, where the former are artificially and uniformly removed from the initial image. This

is however not realistic in our case, as only a few ground-truth field measurements collected on the field can be used to reconstruct the entire map.

In contrast to the previous approaches, in our study, we consider practical situations where data-augmentation techniques cannot be used, mainly because of unknown environment characteristics and computational limitations, and where only a small amount of ground-truth measurements is available.

Finally, a few contributions aim at predicting the received power value based on neural networks and additional information. For instance, in [Hayashi et al., 2020, Nagao and Hayashi, 2020, Inoue et al., 2020], RSS values are predicted in exact points, given meta information such as the radio characteristics (e.g., transmission specifications or relationship between Rx and Tx, like horizontal/vertical angle, mechanical/electrical tilt angle, 2D/3D distance, base station antenna orientation, etc.) and/or prior information about the buildings (e.g., height and presence). In case the latter information is missing, predictions can be made also by means of satellite images (e.g., paper [Inoue et al., 2020]). In these papers though, the map reconstruction cannot be performed directly. As the prediction is realized for each point separately, it is thus time consuming. Moreover, the authors do not take into account the local signal values, but only the physical parameters and physical surroundings (similarly to standard path loss models).

In our study, we look at how well neural networks can extract complex features and their relationships to signal strength in the local area or under similar conditions, as well as their ability to take into account additional environmental information without having access to more complex physical details. This is performed through a search for a model with an optimized architecture adapted to the task.

1.4 Followed Research Methodology

Before applying machine learning techniques for map reconstruction and taking into account all the difficulties mentioned above, we will first analyze the given experimental data, as well as the best localization accuracy achievable via power-based fingerprinting, by means of theoretical performance bounds. We will also analyze the sensitivity of fingerprinting with respect to the quality of the prior (interpolated) radio map and accordingly, identify the main dominating factors impacting the localization performance based on realistic measurements and models.

To summarize, to address the problem stated in section 1.2, we propose to split the narration into the following chapters:

- In Chapter 2, we will first describe and analyze experimental datasets resulting from different measurement campaigns and settings, with a focus on a multi-BS dataset generated in the city of Grenoble, for which we contributed to collect field data. We will also derive and analyze theoretical performance bounds on positioning accuracy in light of RSSI dynamics and model parameters, while emphasizing the impact of both the local spatial behavior of RSSI 2D gradients and the quality of prior RSSI reconstruction.

- Then, in Chapter 3 we will recall some general machine learning techniques and approaches in relation to our stated problem.

- Further, in Chapter 4, we will introduce our first contribution regarding the coupling of self-learning and neural architecture search techniques for a single-snapshot radio map setting.

- In Chapter 5, we will present an algorithm that can operate with better generalization capabilities and takes into account side information about the test environment.

- Finally we will present the main outcomes of this work in Chapter 6, along with their identified limitations and perspectives.

Part I

State-of-the-art

Chapter 2

Localization-oriented Discussion on the Radio Map Reconstruction Problem

As already alluded in the previous chapter, in order to draw maximum benefits from the deterministic approach taken in fingerprinting localization, one should ideally rely on *a priori* maps of geo-referenced radio metrics (i.e., one map per Base Station), which must be i) as complete a possible (e.g., preferably, with reference RSS values available over the entire discretized 2D space where the mobile tag must be localized) and ii) reliable enough (i.e., fairly representative of the instantaneous measurements used for comparison during the online phase, while still reflecting spatial propagation effects that somehow "indicate" the tag position). Regarding the former point, when such reference radio maps are built out of real field measurements (typically, based on GNSS-tagged packet transmissions from sensor nodes, which can be collected at one or several BSs simultaneously), they usually need to be completed in non-visited locations (i.e., pixels of the map for which no real measurement has been collected), thus leading to a **problem of map reconstruction** or equivalently, a problem of spatial interpolation. Then, in each "visited" pixel of the map, it is also quite common to average series of instantaneous measurements to generate the corresponding reference RSS value associated with this pixel, so as to reduce unwanted dispersion effects resulting from specific user's behaviour and/or non-deterministic propagation phenomena, as it will be discussed hereafter in more details.

Given our underlying fingerprinting positioning application, we thus herein devise not only the quality/fidelity of the reference RSS maps that are reconstructed out of sparse, non-uniform and highly dispersed field measurements, but also the methodology to assess this quality (e.g., spatial granularity, prior smoothing...), as follows:

- **RSS measurements are notoriously dispersed over both time** (e.g., due to small-scale fast fading of the received power in dynamic multipath environments, fast changing tag attitude/orientation...) **and space** (e.g., due to both over-the-air range-dependent power losses and correlated shadowing caused by

physical obstacles...). Intuitively, from a fingerprinting standpoint, the latter spatial effects should be properly "captured" and accounted in reconstructed radio maps (so as to efficiently indicate the tag position), while getting rid of all other nuisance fluctuations.

  - First of all, the large dispersion of RSS measurements observed in each pixel (i.e., over users, time periods, tag IDs, tag orientations...) may have a harmful impact on the practicality of radio map reconstruction itself (*a fortiori* under spatially or temporally sparse measurements, See also below). For instance, it is sometimes hard to tell if some strong discontinuities observed in the signal dynamics (typically between adjacent pixels of the map "visited" during field data collection) shall be imputed to desired location-dependent effects such as highly localized radio obstructions, or just to residual time-varying fluctuations that could not be properly averaged out (typically, in case too few measurements are available before averaging in a pixel);
  - Secondly, as fingerprinting ultimately intends to recognize similarities between online received signals and expected radio signatures, even when the reference radio maps adequately capture location-dependent propagation effects (e.g., in case a sufficiently large number of measurements is averaged per visited pixel), they may still differ significantly from online measurements, again due to large instantaneous fluctuations of the latter around their "expected" average value. This tends to degrade the performance of fingerprinting positioning performance, regardless of the quality of the reconstructed reference map though.

- Obviously, real RSS cannot be measured in all the points of the map during the field data collection phase, as it would be impractical and time-consuming. Moreover, these measurements are bounded to topological constraints, while being collected uniquely along roads or streets (e.g., when the tag is aboard a car or carried by a pedestrian or a bicycle...), leading to over-represented data in particular dimensions of space (typically, along the track followed during data collection, to the detriment of the cross-track data). Finally, some measurements can be ill-labelled (e.g., due to erroneous GNSS readings in urban canyons) or simply missing from the database, due to detected artefacts, failures occurred during the measurement process or out-of-range transmissions. All those aspects are somehow cumulative, hence contributing to increase the **spatial sparsity and non-uniformity of the measurements used as inputs for map reconstruction** in non-negligible areas.

- Overall, the average RSS dynamics as a function of space is usually much higher when the transmitting tag is close to the Base Station but almost flat at larger transmission ranges (and even beyond, possibly sticking systematically to the receiver sensitivity). This may be imputed to the fact that, at shorter distances, range-dependent propagation losses are somehow dominating, by the way with

a higher probability to lie in free-space conditions. As a result, beyond a certain distance to the BS, the RSS gradient values calculated out of measurements are shown to be small (See e.g., Figure 2.6), thus limiting the amount of exploitable location-dependent information conveyed in those sub-areas of the map for fingerprinting purposes, as it will be theoretically shown here in the following (See Section 2.2). This tends to suggest that **the quality of the reconstructed map** (i.e., in terms of the relative amount of exploitable "location information" it conveys for fingerprinting with respect to each contributing Base Station) **shall hence not be equally important over the entire 2D scene** in a typical fingerprinting application context.

- In line with the previous point on RSS dynamics but regardless of the map reconstruction quality, the question of **system observability** is thus also critical in our fingerprinting problem (i.e., a measure of the capability to estimate the tag position out of available RSS measurements and a priori maps). It depends on the available amount of location information conveyed by each reference radio map individually, as well as on the number of available maps with respect to distinct BSs in a given tag position (i.e., the number of reachable BSs). Intuitively, as a toy example, if numerous BSs (hence, numerous radio maps) were reachable from the tag but not informative enough with respect to its occupied location (e.g., typically, all with nearly-flat RSS signal dynamics as a function of space), the system would be non-observable. The other way round, under constrained infrastructure deployment conditions, if a RSS map with respect to one single BS has fairly high dynamics but ambiguous values (typically, when several points of the map have similar reference RSS values), the system would not be observable either.

Overall, the remarks above question:

- the spatial resolution to be considered in our analysis in comparison with the spatial dynamics of the observed signals (i.e., typically, the optimal pixel granularity used for initial measurements aggregation, as well as for further post-processing such as data smoothing to preserve mostly low spatial frequencies);
- the rules/criteria for properly detecting and rejecting measurement outliers from our available experimental dataset (which may introduce strong biases on the analysis otherwise);
- the underlying models to be chosen while accounting for undesired noise effects (i.e., affecting both instantaneous RSS measurements besides and reconstructed reference maps);
- the physical effects that dominate positioning performance in multi-BS RSS map-based fingerprinting approaches (e.g., radio map dynamics, minimum number of deployed BSs);
- the most relevant analytical tools to be used in the study of our radio map reconstruction problem in the very fingerprinting problem.

We herein tackle some of these challenges by analyzing the theoretical positioning performance of map-based fingerprinting techniques, while considering various sets of experimental data as illustrative application examples. The latter sets rely on both real field measurements and realistic ray-tracing simulations, all generated in representative urban environments. On this occasion, we highlight the importance of underlying RSS modeling assumptions (more specifically, as regards to the random variations assumed around "useful" deterministic location-dependent components) on the one hand, as well as the difficulty to establish meaningful and exploitable theoretical performance bounds under sparse and dispersed input measurements on the other hand. The findings of this first qualitative discussion are intended to guide the design of radio map reconstruction algorithms through machine learning in the following chapters (typically, in terms of the used performance indicators and the performance targets for the map quality, the required pre-processing of the input data before feeding learning algorithms, as well as the expected orders of magnitude for final positioning performance...).

In Section 2.1, we first discuss two different RSS models, where the received power is expressed in dB. They correspond respectively to a classical single-link log-normal model, which includes a deterministic range-dependent power path loss (besides normally distributed random variations accounting for both shadowing and fast-fading residual after averaging), and a fingerprinting-oriented model, where the deterministic part includes both range-dependent power path loss and position-dependent shadowing effects (besides normally distributed random variations accounting for residual fast-fading only). Based on these two models, in Section 2.2, we will then recall the close-form expressions of the Cramer Rao Lower Bounds (CLRB) that characterize the location accuracy of any unbiased estimator, respectively in case of conventional so-called "parametric" positioning, where link-wise RSS measurements with respect to BSs are mainly interpreted in terms of their transmission ranges, and fingerprinting based positioning, where RSS values shall directly indicate the tag position (in a deterministic way). In the latter case, we will more specifically point out the central role of RSS 2D space gradient values in the bound calculation (i.e., gradient with respect to 2D Cartesian dimensions). Then Section 2.3 will describe the experimental datasets used in our analysis. Finally, based on these experimental data, we will provide concrete numerical illustrations of the RSS model parameters and accordingly, we will evaluate CRLB position error bounds in realistic deployment settings (typically, comparing the two positioning approaches as a function of the number of available BSs). On this occasion, we will also discuss the necessity to smooth the empirical RSS gradient values, so as to limit the effects of real measurements artefacts, dispersion and sparsity on the evaluation of positioning performance bounds, while aiming at preserving only the main spatial trends (i.e., only 2D processes with slower spatial frequency components).

2.1 Main Radio Propagation Phenomena and Related Received Power Models

The way the electromagnetic waves are behaving and interacting within their physical environment in the context of a radio transmission between a transmitter and a receiver is generally depicted as *radio propagation*. There are different propagation conditions, such as Line of Sight (LOS), when there is no obstruction/obstacle on the direct way of the radio waves, or Non Line of Sight (NLOS) in the opposite case. In dense urban environments typically, the direct path between a tag and a Base Station may be may be subject to such blockage with high probability. The radio wave can thus propagate by multiple paths. As a result, the received signal usually contain not only the direct path (if only present), but also other secondary multipath components which arise after the interaction of the radio wave with different obstacles like terrain, leafage, buildings, etc. These electromagnetic interactions include various phenomena, such as reflection, transmission, absorption, scattering and diffraction, ultimately impacting both the average power loss of the received signal along its way, as well as its dispersion. This obviously tends to complexify the problem of map reconstruction in the sense of trying to predict the actual path loss in some non-visited points, by means of a priori parametric models (e.g., conditional range-dependent path loss models) or data-oriented approaches (e.g., through machine learning based on real field measurements or deterministic simulations).

In the literature, several path loss models can be found, which can be roughly divided into three groups [Luo, 2013, Sati and Singh, 2014, Seybold, 2005]: *Deterministic*, *Statistical* and *Empirical*[1]. First, the *Deterministic* models rely on the presumably "exact" calculation of electromagnetic interactions, hence requiring a detailed description of both the physical environment and the geometry of the radio link. In case significant changes are introduced to the latter operating environment, the output of the model needs to be recomputed. Ray-Tracing models (see Section 1.3.1) are examples of such deterministic models. On the contrary, *Statistical* models are parametric models that describe the received signal as a random variable (at least in part, although they may still include some deterministic components). Even if they need less exhaustive a priori information than deterministic approaches, they may still be ruled by a few key parameters, which must be set a priori for a given type of environment and/or which must be conditioned to the operating context itself (e.g., in terms of LoS/NLoS). On the other hand, they are also supposed to be less accurate, as they mostly account for the random dispersion of the received signal around its average trends or behavior (typically, by means of conditional probability density functions). Finally, most of the radio propagation models practically in use combine both empirical and analytical approaches, which can be viewed as an simplified extension to the previous statistical models. These *Empirical* methods thus try to find the best analytical expression for the model parameters that fits the statistics of real collected measurements.

[1]The term "Empirical" may sound abusive or at least questionable here in the sense it still refers to underlying statistical models, even though fed by field experimental data.

One relevant example of such a model is the classical average power path loss model between the transmitter and receiver, whose parameters are empirically derived from collected data measurements (as it will be shown further in this section). This model can be expressed as (from [Rappaport, 2002]):

$$PL(d) = PL(d_0) + 10n\log_{10}(\frac{d}{d_0}) + \epsilon,\ \epsilon \sim N(0, \sigma^2) \tag{2.1}$$

where $PL(d)$ is an average path loss value, expressed in dB, as a function of the distance d between the transmitter and receiver, $PL(d_0)$ is a reference path loss value, measured in dB, in a reference distance d_0 close to the transmitter ((theoretically) corresponding to a free-space path loss, usually 1 m), n is a constant parameter denoted as the path loss exponent, ϵ is a random variable typically accounting for the dispersion of the average received power (typically due to shadowing effects in our case), which is usually assumed zero-mean with standard deviation σ (in dB). As mentioned above, $PL(0)$ and n must be empirically evaluated from the given set of measurements (see Equation 2.18); Moreover, n is theoretically equal to 2 in free-space single-path conditions, while in more obstructed conditions this value is expected to increase, meaning that the received power decay is more rapid as a function of range.

Based on Equation 2.1, the average received signal strength value P^{dBm}, expressed in dBm, for a given distance between the transmitter and the receiver is hence expressed as:

$$\begin{aligned} P^{\text{dBm}}(d) &= P_{Tx}^{\text{dBm}} + G_{Tx} + G_{Rx} - PL(d) \\ &= P_{Tx}^{\text{dBm}} + G_{Tx} + G_{Rx} - PL(d_0) - 10n\log_{10}(\frac{d}{d_0}) - \epsilon \\ &= P_0^{\text{dBm}} - 10n\log_{10}(\frac{d}{d_0}) - \epsilon, \end{aligned} \tag{2.2}$$

where P_{Tx}^{dBm} is the transmit power (in dBm), G_{Tx} and G_{Rx} are the transmit and receive antenna gains respectively (in dBi).

2.2 Link between Theoretical Positioning Performance and RSS Models

As already mentioned before, so as to accurately localize a tag based on RSS measurements through fingerprinting techniques, one need to build a reliable RSS map first, that is to say, a map which faithfully respects the actual average and deterministic signal variations as a function of the tag position (hopefully including e.g., path loss and shadowing effects). It is hence worth investigating how the theoretical performance bounds of positioning (and more specifically, that of fingerprinting-based positioning approaches) can be influenced by the RSS signal dynamics that could be retrieved in such a priori maps (typically, through their spatial gradients as a function of the 2D Cartesian dimensions), but also by key model parameters asides (e.g., the standard deviation of random terms accounting for the difference between the RSS

measurements performed during the online phase, which could be also advantageously averaged to get rid of the fastest fading fluctuations, and the interpolated RSS map values, which would likely include interpolation errors). Ultimately, the latter factors can hence be somehow related to the map interpolation quality, for instance while pointing out (i) that interpolated RSS maps shall have smooth gradient variations as a function of space (preferably with no sudden artefacts or border effects) so as to avoid being over-optimistic in the prediction of fingerprinting positioning performance, or (ii) that the interpolation error (in dB) shall not exceed a certain threshold.

In our localization context, the Cramer-Rao Lower Bound (CRLB) is a lower bound on the variance of any unbiased estimator of the tag position $\theta = (x, y)$, given the a priori known statistical distribution of RSS observations related to the estimated coordinates. Hereafter, for benchmark purposes, we first recall the theoretical performance bounds of RSS-based positioning while relying on the standard range-dependent parametric path loss model already introduced above. Even if this localization technique is presumably less accurate than the map-based fingerprinting approaches at the core of our PhD investigations, this will not only give us an order of magnitude for the expected positioning errors, given typical and realistic RSS dynamics, but also a sense of their spatial distribution (typically, as a function of the number and position of Base Stations). Then, as a second step, we characterize similar performance bounds in the fingerprinting-based localization case of interest, assuming an a priori RSS map that is expressed as an explicit 2D function of the Cartesian coordinates (typically, the interpolated map).

2.2.1 CRLB of positioning based on a single-link log-normal path loss model

For any tag position $\theta = (x, y)$, we consider a set of independent average received power measurements $\{P_i^{\mathrm{dBm}}\}$, $i = 1..N$ (in dBm) with respect to N base stations, which are assumed to be zero-mean and normally distributed with respective variances $\{\sigma_i^2\}$, according to the classical log-normal path loss model of Eq. 2.2):

$$P_i^{\mathrm{dBm}}(\theta) = P_{0,i}^{\mathrm{dBm}} - 10n \log_{10} \frac{d_i(\theta)}{d_{0,i}} + w_i, \tag{2.3}$$

where $d_i(\theta) = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ is the distance from a base station i of 2D Cartesian coordinates (x_i, y_i) to a tag of Cartesian coordinates $\theta = (x, y)$, $P_{0,i}^{dBm}$ is the free-space average received power at the reference distance $d_{0,i}$, $i = 1...N$, $w_i \sim \mathcal{N}(0, \sigma_i^2)$. For simplicity in the following, we note $P_i^{\mathrm{dBm}}(\theta) = P_i$, $d_{0,i} = 1$ m and $P_{0,i}^{\mathrm{dBm}} = P_0$, $\forall i = 1..N$ and $d_i(\theta) = d_i$.

Accordingly, the Fisher Information Matrix (FIM) $\mathbf{F}$ characterizing the amount of information conveyed by the RSS observations for the estimation of position $\theta = (x, y)$ consists of four elements, as follows:

$$\mathbf{F} = \begin{pmatrix} F_{xx} & F_{xy} \\ F_{yx} & F_{yy} \end{pmatrix} \tag{2.4}$$

where F_{xx} are demonstrated to be:

$$F_{xx} = \sum_{i=1}^{N} \frac{b_i^2}{\sigma_i^2} \frac{(x - x_i)^2}{d_i^4} \tag{2.5}$$

with $a_i = P_i - P_0$ and $b_i = \frac{10n_i}{\ln 10}$. Similarly, the term F_{yy} is calculated as:

$$F_{yy} = \sum_{i=1}^{N} \frac{b_i^2}{\sigma_i^2} \frac{(y - y_i)^2}{d_i^4} \tag{2.6}$$

and the term F_{xy} (or F_{yx}) as:

$$F_{xy} = \sum_{i=1}^{N} \frac{b_i^2}{\sigma_i^2} \frac{(x - x_i)(y - y_i)}{d_i^4} \tag{2.7}$$

All the intermediary calculation steps can be found in Appendix A.1.

Finally, to compute the Position Error Bound (PEB) PEB_p in case of such parametric positioning, which characterizes the best variance for the position estimation, we simply take the trace of the inverse of the FIM in Eq. 2.4, leading to:

$$var \geq PEB_p = \frac{F_{xx} + F_{yy}}{F_{xx}F_{yy} - F_{xy}^2} \tag{2.8}$$

Note that, in case $b_i = b$ and $\sigma_i = \sigma \ \forall i$, the final formulas are similar to that in [Patwari and Hero, 2002] for the non-cooperative localization of one mobile tag.

2.2.2 CRLB of positioning through map-based fingerprinting

The underlying model with respect to each base station is now represented as a function of the potential tag position $\theta = (x, y)$ (rather than the range), as follows:

$$P = g(\theta) + w^l(\theta) + w^t = g(\theta) + w, \tag{2.9}$$

where $g(\theta)$ is the so-called "ground-truth" RSS value in the position $\theta = (x, y)$ (hopefully, containing both deterministic range-dependent average power losses and dominating macroscopic shadowing effects resulting from the environment topology, such as buildings obstructions, etc... that should indicate tag's position), $w^l(\theta)$ and w^t are additive random noise terms caused respectively by map interpolation errors and/or specific smaller-scale measurement-dependent slow-fading effects besides the ground truth (e.g., resulting from body shadowing or specific antenna orientation during the online measurements, etc.) on the one hand, and fast-fading fluctuations due to multipath under mobility on the other hand. In general, we have $w \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = (\sigma^l)^2 + (\sigma^t)^2$. However, assuming that a large amount of instantaneous measurements are available in a given position during the online phase too (i.e., before comparision with the map), one can perform averaging so that the residual fast-fading fluctuations can be neglected in first approximation, i.e., $w^t(\theta) \to 0$.

The latter model thus somehow statistically accounts for the deviation between a RSS measurement collected during the online phase and the a priori interpolated RSS map that is used for comparison in the fingerprinting database. In other words, in first approximation, it also and foremost accounts for the quality[2] of the interpolated map.

For each base station, we hence simplify the expression of the average received power P_i, $i \in \{1, .., N\}$ into:

$$P_i(\theta) = g_i(\theta) + w_i \tag{2.10}$$

where $w_i \sim \mathcal{N}(0, \sigma_i^2)$.

Like in the parametric case, so as to compute the FIM from a position estimation standpoint[3], we first express the joint log-likelihood function of the set of N independent measurements, $\mathbf{P}$, along with its successive derivatives with respect to the estimation variables:

$$f(\mathbf{P}|\theta) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2\sigma_i^2}(P_i - g_i(\theta))^2} \tag{2.11}$$

$$\frac{\partial \log f}{\partial \theta} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2}(P_i - g_i(\theta)) \frac{\partial g_i(\theta)}{\partial \theta} \tag{2.12}$$

$$\frac{\partial^2 \log f}{\partial \theta^2} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left((P_i - g_i(\theta)) \frac{\partial g_i^2(\theta)}{\partial \theta^2} - \frac{\partial g_i(\theta)}{\partial \theta} \cdot \frac{\partial g_i(\theta)}{\partial \theta} \right) \tag{2.13}$$

The expectation of the expression $P_i - g_i(\theta)$ being equal to zero (over the random noise distribution), it simply comes:

$$F_{xx} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(\frac{\partial g_i}{\partial x} \cdot \frac{\partial g_i}{\partial x} \right)$$

$$F_{yy} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(\frac{\partial g_i}{\partial y} \cdot \frac{\partial g_i}{\partial y} \right)$$

$$F_{xy} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(\frac{\partial g_i}{\partial x} \cdot \frac{\partial g_i}{\partial y} \right)$$

Just like in the parametric case too, the PEB in case of fingerprinting PEB_f can finally be calculated in any cell/position of the map, as the trace of the inverse of

[2]By "quality", we mean here both interpolation accuracy in comparison with the ground-truth map, and further dispersion effects that could not be accounted in the map (i.e., independently of the interpolation accuracy), typically over different tags or carriers.

[3]Note that this way of presenting fingerprinting as a pure estimation problem may look somehow restrictive though, in the sense the fingerprinting problem is also sometimes rather presented as a decision problem (typically, when one needs to chose the entries in a database that best fit the current online measurement, that is, making a decision among a subset of discrete possibilities.). However, other fingerprinting techniques such as that based on non-linear regression or interpolation between database entries would still fall under the class of estimation problems.

the FIM. The latter hence strongly depends on both the 2D spacial gradients of the ground-truth RSS map (which can be calculated numerically for analysis) and the respective BS-dependent standard deviations $\{\sigma_i\}$ of random noise terms $\{w_i\}$:

$$var \geq PEB_f = \frac{F_{xx} + F_{yy}}{F_{xx}F_{yy} - F_{xy}^2} \tag{2.14}$$

2.3 Experimental Datasets

In the area of data-based research and in the field of machine learning singularly, it is usually hard to find large open-source datasets made of real data. In some works however, alternatively (or as a complement) to using real data, synthetic data can be generated, for instance through deterministic simulations (as already mentioned in Section 1.3).

In our study, we make use of three distinct databases of outdoor RSS measurements with respect to multiple base stations. The first one was generated through a Ray-Tracing tool in the city of Paris, France. The second database, which is publicly available (See [Aernouts et al., 2018]), consists of real GPS-tagged LoRaWAN measurements that were collected in the city of Antwerp (The Netherlands). Finally, a third database, which is also made of real GPS-tagged LoRaWAN measurements, was specifically generated in the city of Grenoble (France), in the context of this PhD work.

2.3.1 Paris dataset

This first dataset is made of synthetic outdoor RSS measurements, which were simulated in a urban Long Term Evolution (LTE) cellular context[4] with a ray-tracing propagation tool named VOLCANO (commercialized by SIRADEL). Those simulations were calibrated by means of side field measurements [Brau et al., 2012]. As already mentioned in section 1.3.1, this kind of deterministic tool makes use of both the deployment information (typically, the relative positions of mobile nodes and base stations) and the description of the physical environment (i.e., a city layout with a faceted description of the buildings, along with their constituting materials) to predict explicitly the electromagnetic interactions of the multipath radio signal between a transmitter and a receiver. Beyond the main limitations already mentioned in section 1.3.1 regarding mostly computational complexity and prior information, we acknowledge a certain number of discrepancies or mismatches in comparison with

[4]Although the operating frequency (2.1GHz) here differs from that of the LPWA technology targeted in our study (868MHz), these simulations were exploited as the most complete and most realistic outdoor simulated database available at CEA by the time this study was conducted, thus making possible the evaluation of first map interpolation algorithms at an early stage. By the way, from a radio map reconstruction perspective, it was mostly intended for benchmark purposes with real field measurements (See Antwerp and Grenoble datasets hereafter), while providing more favorable (idealized) conditions, for instance in terms of the spatial density/granularity of available "measurements".

the two other datasets based on real measurements. For example, in the simulated scenario, the dynamic range of observed RSS is continuous in the interval [-190, -60] dBm, while with the real measurement data, a receiver sensitivity floor of -120 dBm is imposed. Moreover, the available simulation data was already pre-aggregated into cells, thus imposing somehow the finest granularity. The overall scene is 1000 m $\times$ 1000 m, each pixel being 2 m $\times$ 2 m, thus forming a matrix of size 500 $\times$ 500. The area considered in these simulations is located in Paris between Champ de Mars (South-West), Faubourg Saint Germain (South), Invalides (Est), and Quai Branly / d'Orsay (North), as shown in Figure 2.2. For each pixel, the RSS value was simulated with respect to 6 different Base Stations. An example is given for one of these base stations in Figure 2.1. Further details regarding the considered simulation settings can be found in [Brau et al., 2012].

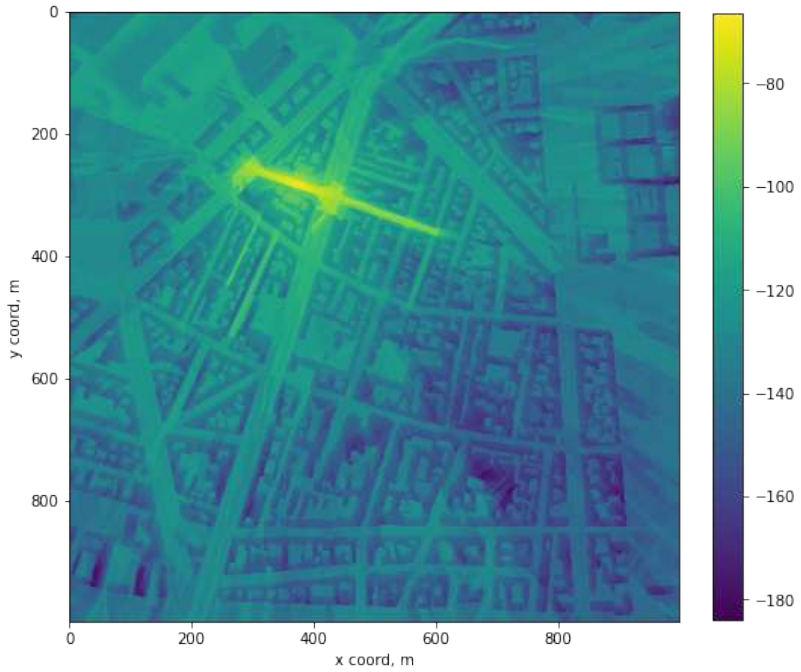


Figure 2.1: Example of signal strength distribution (in dBm) generated through Ray-Tracing in the Paris dataset, with respect to one particular base station roughly located in (300m, 300m).

2.3.2 Antwerp dataset

Measurement campaign and experimental settings The LoRaWAN dataset was collected in the urban area in the city centre of Antwerp (the Netherlands) from 17 November 2017 until 5 February 2018, [Aernouts et al., 2018], [Aernouts, 2022]. The dataset consists of 123,529 LoRaWAN messages including the GPS coordinates corresponding to the position occupied on the map, along with the RSSI measurements associated to that location. It was collected over a network driven by Proximus (which is a nation-wide network) by twenty postal service cars equipped with The

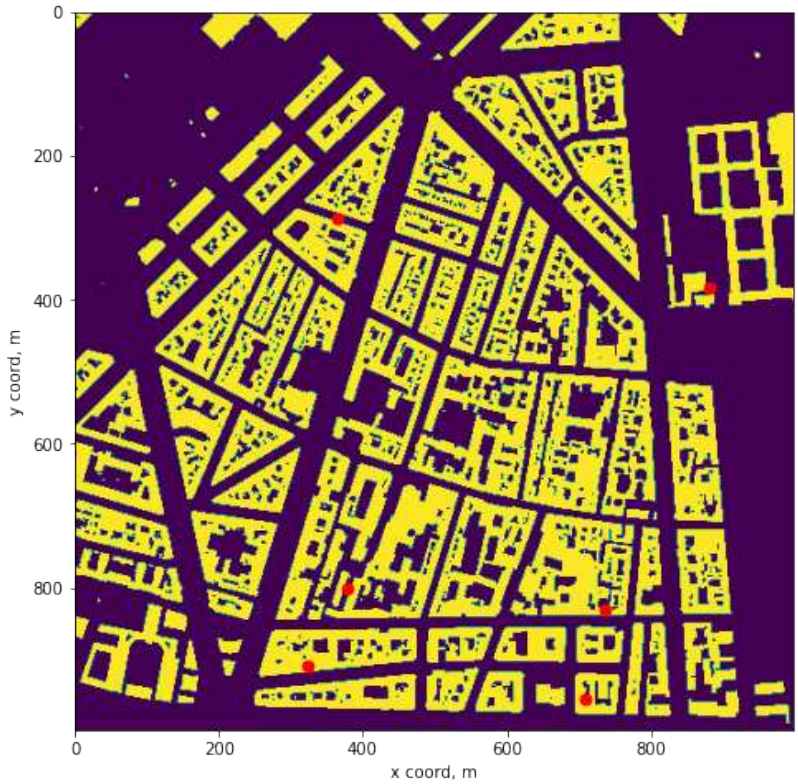


Figure 2.2: Buildings map and corresponding Base Stations positions (in red) for Paris dataset

City of Things hardware. The latitude, longitude and Horizontal Dilution of Precision (HDoP) information were obtained by the Firefly X1 GPS receiver and then sent in a LoRaWAN message by the IM880B-L radio module in the 868 MHz band. The interval between adjacent messages was spanning from 30s to 5 min depending on the used Spreading Factor (SF).

The information was collected for 68 detected base stations in the initial database. We have filtered out some stations which have overall less than 10000 messages and/or which were located too far away from the collection zone, thus exhibiting a flat received power behavior as a function of space. Finally we considered up to 9 base stations – from BS'_1 to BS'_9 (see Figure 2.3), where three of them (BS'_1, BS'_2, BS'_3) marked as black points correspond to that used in Chapter 4, while the overall nine will be considered in Chapter 5.

The initial dataset, including information about each BS or gateway (GW), Receiving time of the message (RX time), Spreading Factor (SF), Horizontal Dilution of Precision (HDoP), Latitude and Longitude, looks as in Table 2.1.

Dataset preprocessing and analysis. In this part, we explain the way the previous dataset was processed for future use in Chapter 4. First, we aggregated the received power measurements into cells of 10 meters $\times$ 10 meters ($10m \times 10m$) and then averaged those measurements, before translating the results into signal strength

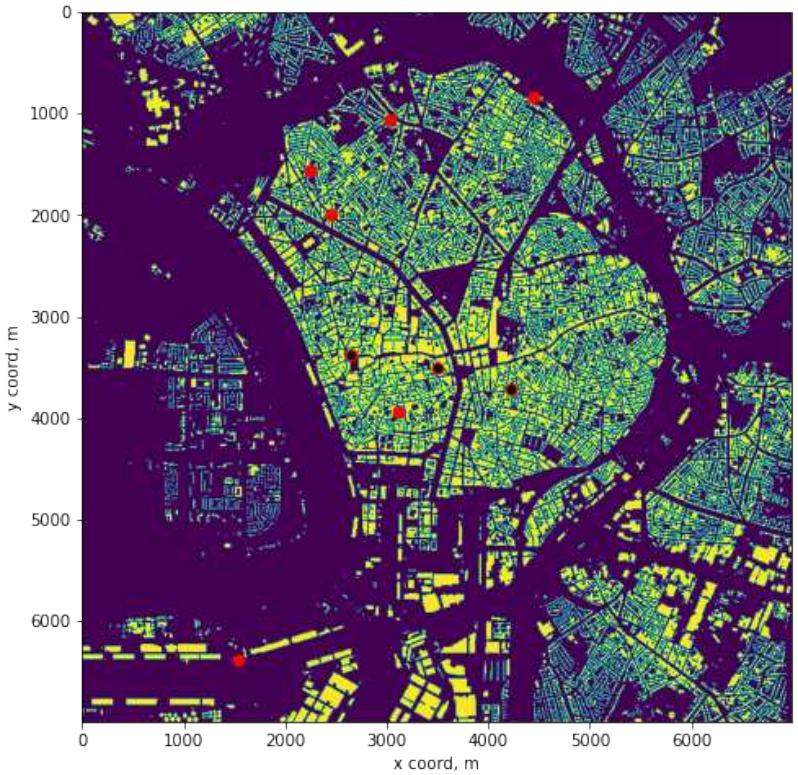


Figure 2.3: Buildings map and corresponding considered 9 Base Stations positions (in red) for Antwerp dataset

GW_1	...	GW_{68}	RX Time	SF	HDoP	Latitude	Longitude
-101	...	-200	2018-01-09T23:42:19	9	0.60	51.194046	4.418624
-200	...	-200	2018-01-31T10:01:27	12	1.08	51.200042	4.411672
...	...	...	...	...	...	...	...

Table 2.1: Example of dataset structure for the measurements collected in the city of Antwerp [Aernouts et al., 2018].

values. To perform this aggregation, we measured the distance from the base station location based on local East, North, Up (ENU) coordinates.

Table 2.2 below reports the amount of measurements in the considered area of interest of $3680m \times 3680m$ around each of the considered base stations after data aggregation into cells of size $10m \times 10m$, with each base station located in the center of this area, as used in the experiments reported in the following Chapter 4.

In Chapter 5, we considered an area of $7000m \times 7000m$ that corresponds to the full city, covering most of positions used to collect field measurements. Similarly, we considered a $10m \times 10m$ data aggregation in this case too. Information about the resulting number of exploitable input measurements can be found in Table 5.1.

In the initial dataset, if in the visited point on the map there was no captured signal, this point was marked as -200 dBm for the corresponding base station, so that,

Base station number	Amount of measurements after aggregation	Spatial density (per km^2)
BS'_1	6450	440
BS'_2	5969	389
BS'_3	7118	525

Table 2.2: Amount of measurements for each base station located in the center of 368×368 image size after $10m \times 10m$ aggregation, Antwerp dataset. Base stations with the highest amount of measurement points around the base station location were selected

as shown in Figure 2.4, the dynamic range for practically exploitable signal values lies in the interval [-120; -60] dBm, where the left boundary corresponds to the sensitivity of the receiving device.

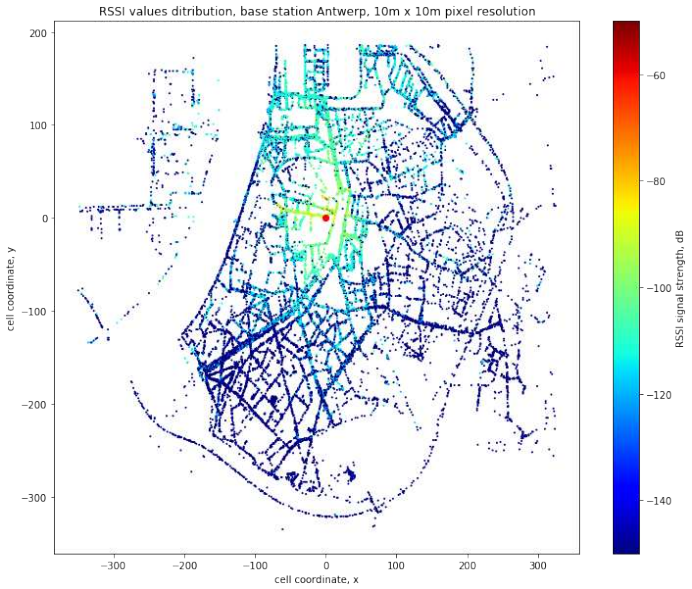


Figure 2.4: Example of received power dynamics, as a result of measurement data aggregation in cells of 10mx10m, for one of the base stations, in the Antwerp experimental dataset.

2.3.3 Grenoble dataset

Measurement campaign and experimental settings During our PhD, an experimental campaign was conducted by my colleagues and me at CEA[5], collecting LoRa measurements in the real urban environment of Grenoble city. We believe that

[5]By the time this PhD manuscript is written, this dataset is still in the process of being extended in terms of both the total amount of collected measurements and the number of receiving base stations.

this data is somehow valuable on its own, especially because of the limited amount of publicly available data for research purposes.

The data is collected for several base stations installed in Grenoble and consists of several parameters such as latitude, longitude, and corresponding RSSI value for each recognized base station. The collected data is hence quite similar to that described for Antwerp, but has with a slightly different structure, as shown in Table 2.3. The total amount of stored lines (i.e., unitary received packets) collected in the database from 13-01-2021 to 11-01-2022 is 1574588. The data was collected in the frequency band of 868 MHz in the LoRaWAN network by several users equipped with personal tags, with different types of mobility patterns (while walking, riding a bike or driving a car).

$GateWay\_ID$	$Device\_ID$	RSS	$latitude$	$longitude$	$time$
7276ff002e0701e5	70b3d5499b4922dc	-95	45.199308	5.712627	2021-08-29T13:46:35
7276ff002e0701f1	70b3d5499b4922dc	-103	45.199308	5.712627	2021-08-29T13:46:35
...	...	...	...	...	...

Table 2.3: Example of the dataset format for the field measurements collected in the Grenoble area, where for each connected gateway-tag pair, we report the corresponding device position, time of collection, RSS value.

To collect the data, COTS telecom grade gateways iBTS from Kerlink manufacturer (see Figure 2.5a) based on the LoRaWAN technology have been used. These gateways have fine time-stamping capability, and are synchronized with the GPS time through a Pulse-per-Second (PPS) signal generated by GNSS receiver included in the gateway with an accuracy of a few nanoseconds. In LoRaWAN technology, each of the tag uplift packets can be received by more that one base station, depending on the local structure (which cause interference) and mainly path loss. To store the received information from the gateways the LoRaWAN Network Server (LNS) was used, as it is shown in Table 2.3 (the amount of stored metrics is bigger – like Signal-to-Noise Ratio (SNR), Time of Arrival (ToA), uplink network parameters : frequency, DataRate (DR), etc. but here we will focus on the data used in our study).

During the data collection, different deployment characteristics and amounts of base stations were considered. Two main settings were hence used at different stages of this work: Grenoble-1 (which was used in the experimental part of Chapter 4) and Grenoble-2 (which was used in the experimental part of Chapter 5).

The first version of this dataset (Grenoble-1) consisted of two base stations with a sufficient amount of collected measurements (overall, up to 6 base stations were available, but only two of them had more than 10000 received packets). This version will be later used in the validation of the results in the experiments section of Chapter 4.3.

A second version of this dataset (Grenoble-2) consisted of four base stations located in the city of Grenoble and one more base station, which was located far from the city area and thus discarded because of flattened signal characteristics over most of its collected measurements. This version will be further used for the validation of the results in the experiments of Chapter 5.4. As an example, the positions of four

(a) Gateway BS_3 which is installed on the roof near the cable car station at Bastille, Grenoble

(b) Tag which was used to collect the data

Figure 2.5: Devices (i.e., tags and gateways) that were used during the measurement collection campaign for the Grenoble dataset.

considered base stations are shown in Figure 2.7b. This deployment is also interesting as one of the base stations (BS_3) is installed on a mountain peak higher than all the other base stations, thus making this database quite specific.

Dataset preprocessing and analysis.

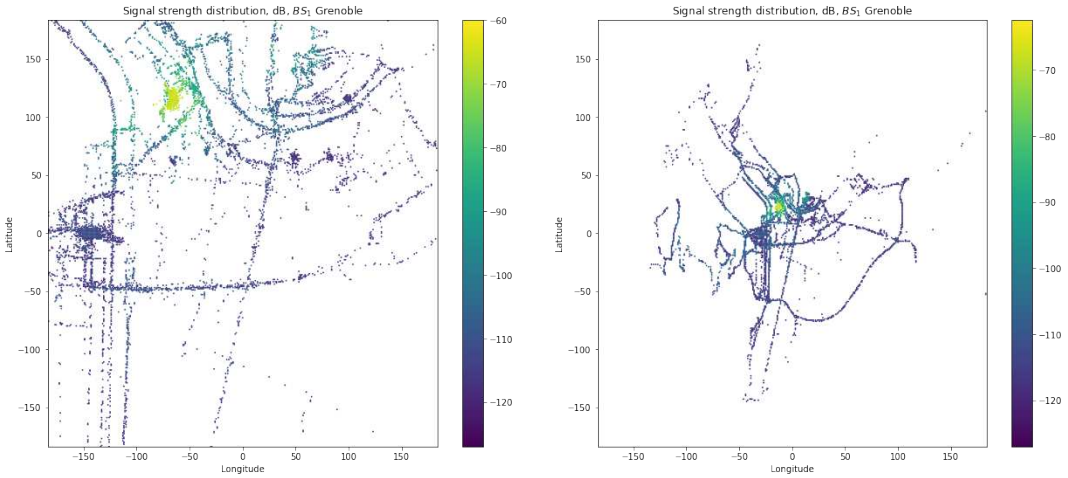
Removal of measurement outliers and artefacts First, we needed to filter out of the dataset the unreliable data resulting from errors during the data collection or from measurement artefacts. For example, the exact GPS position could be significantly different from the real position (due to a lack of visibility to satellites) or, due to a specificity in the tag design, data transmission still occur while charging indoor, thus giving both the wrong RSS value and/or the wrong GPS position. For instance, regarding the latter issue that is quite obvious to detect, we simply rejected all the measurements exhibiting too high RSS values and/or being static for a long time, which were most likely collected during the charging of the device. Being more precise, first we detected the base stations for which the received signal strength was higher than -55 dBm for the entire acquisition time sequence and then removed the corresponding measurement points collected at the same time for this device with re-

spect to all the other base stations. However, RSS values saturating at short distances or reaching receiver sensitivity at large distances were preserved in the database, for being somehow indirectly indicative of the tag distance to the BS. Finally, we filtered out all the measurements for which the GPS latitude was not valid (out of tolerated range).

Data aggregation per cell Just like in the previous dataset, after removing outliers/artefacts, we then aggregated the signal in the cells. We converted the RSS into milliWatts (as [dBm] = $10\log_{10}$[mW]), computed its average per cell in the cells of size $10m \times 10m$, and converted back the result into signal strength values. To perform this aggregation, we measured the distance from the base station BS_1 location considered to be (0,0) 2D Cartesian coordinate based on local East, North, Up (ENU) coordinates. Finally, we considered an overall area of interest of $3680m \times 3680m$ (also for the radio mapping application), which covers the entire city, while containing most of the deployed base stations, as shown in the Figure 2.7b.

Base station number	Amount of measurements after aggregation	Spatial density (per km^2/ per 10^4 cells)
BS_1	16577	1231
BS_2	7078	515

Table 2.4: Amount of measurements for 2 base station in the Grenoble-1 dataset (first version). Only the Base stations with the highest amount of points were selected.

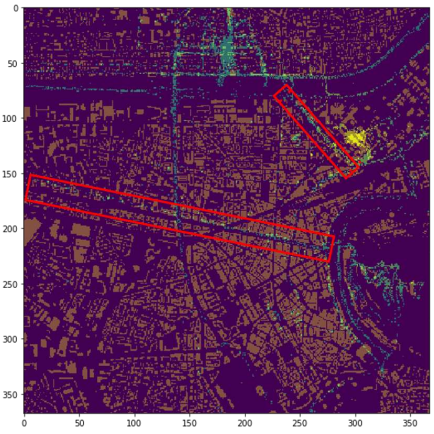


(a) 10x10 aggregation (b) 50x50 aggregation

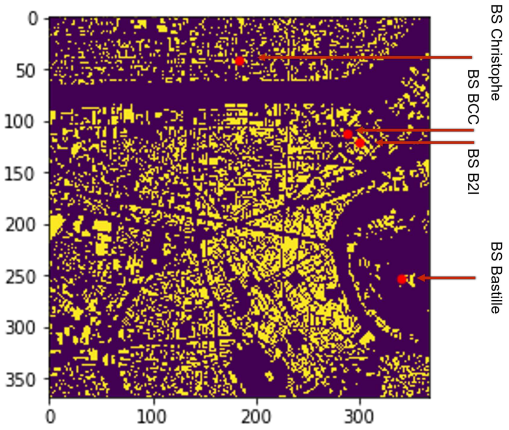
Figure 2.6: Signal distribution for different aggregation cell sizes, BS_1, Grenoble

To compare the measurements *std* values per cell in different conditions, we considered two types of aggregation cell sizes: $50m \times 50m$ and $10m \times 10m$, as shown in Figure 2.6. In case of $50m \times 50m$ aggregation cells, the amount of informative pixels (i.e., visited pixels with sufficient measurements) obviously reduces but at the same

time, the aggregated value is more stable as a function of space (from pixels to pixels), while with $10m \times 10m$ aggregation cells, we can see significantly larger fluctuations of the average received power as a function of space but making available a larger amount of informative points for mapping.



(a) Example of roads in red boxes in LoS with respect to at least one of the deployed base stations.



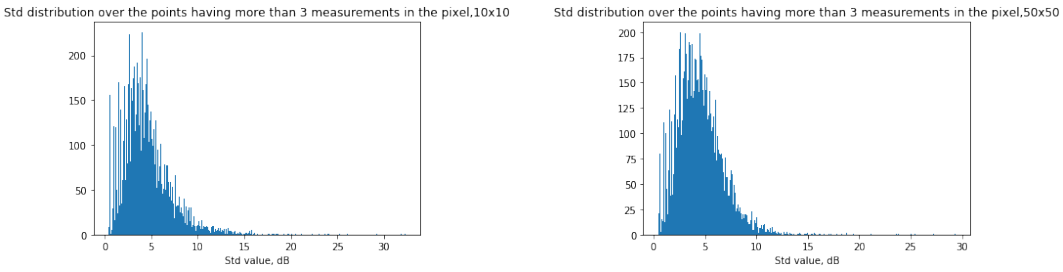
(b) Positions and names of some of the base stations deployed in Grenoble.

Figure 2.7: Part of the Grenoble map, with a selection of the deployed base station positions (right) and two canonical streets in LoS with respect to the latter.

RSS dispersion per cell The empirical standard deviation *std* of the collected measurements per cell after data aggregation has also been calculated for the two previous cell sizes, so as to study its distribution over cells having at least 3 measurements. First, the dependence of *std* on the cell size is analyzed on both Figure 2.8 and Table 2.5. While comparing the two settings, the distribution mean looks similar, while other characteristics differ only marginally. For the $50m \times 50m$ cell size, it turns out that the points with high *std* are mostly located closer to the base station, while for the $10m \times 10m$ granularity, the *std* values are distributed more uniformly over the entire considered region of 368 by 368 cells. It comes from the fact that, within typical $50m \times 50m$ cells close to the BS, the average RSS signal dynamic is such that the dispersion around the cell average value (i.e., the average of all the measurements collected in this cell) between the minimum and the maximum measurement values (i.e., even besides fast fading fluctuations) is naturally much larger than in the $10m \times 10m$ case. In other words, the fine-grain average deterministic range-dependent power decay is interpreted as extra random fluctuations in $50m \times 50m$ cells, due to a loose spatial grid. Thereby, to preserve more information about the variability of the signal while solving the problem of map reconstruction, we will consider a $10m \times 10m$ cell granularity in the following.

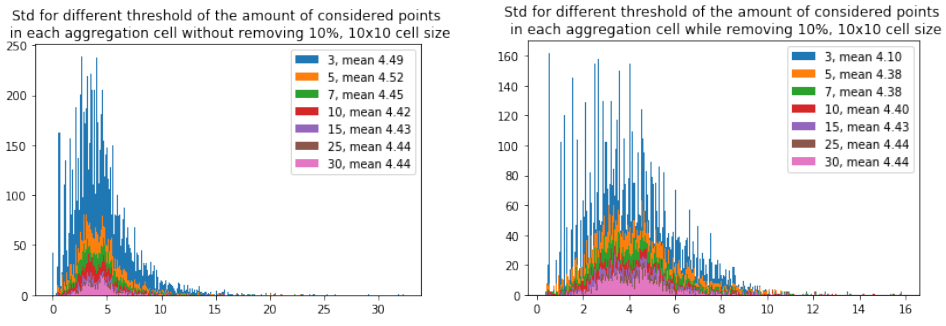
Then, keeping a $10m \times 10m$ cell size, we further investigate the influence of the minimum amount of available measurement points per cell (spanning from 3 up to

30 measurements), with or without removing 10% of the points having the largest variance of RSS measurements divided by the number of samples after in-cell data aggregation (See Figure 2.9). This indicator indeed gives a hint on the capability to reduce fast fading dispersion through the coherent integration of in-cell instantaneous RSS measurements (i.e., variance of residual dispersion after in-cell averaging). After filtering out the data (Figure 2.9b), the overall empirical distribution shape looks rather similar, even if its standard deviation is clearly decreased, as expected. This contributes typically to limit the number of cell occurrences hosting a RSS *std* larger than 10dB, which are expected to very harmful to the fingerprinting process (typically, by limiting the suppression of fast-fading through averaging). Beyond, as a relatively limited amount of input points could be visited physically during the collection campaign within this experimental dataset, when the minimum number of points to keep the cell is too demanding, the number of exploitable cells decreases drastically while the distribution characteristics over the cells do not vary much. Accordingly, in terms of data preprocessing strategy, in the following (for further model parameters extraction or before applying our map interpolation algorithms), we will systematically reject 10% of the cells with the highest RSS variance divided by number of measurements, while keeping $10m \times 10m$ cells with at least 3 measurements.



(a) $10m \times 10m$ data aggregation cells (b) $50m \times 50m$ data aggregation cells

Figure 2.8: Distribution of the standard deviation of RSS measurements per cell (over all the cells), for different aggregation cell sizes.



(a) Without removing the points. (b) After removing the points.

Figure 2.9: Mean value over std with different threshold over amount of points in each cell, dB, for $10m \times 10m$ cell size

Granularity	Mean all, dB	Median all, dB	Std, dB
10x10	4.50	4.04	2.69
50x50	4.50	4.24	2.26

Table 2.5: Different characteristics for the distribution of the STD value per cell in one street, for the points with more than 3 aggregated real measurements.

Extraction of path loss model parameters Prior to addressing more explicitly the radio map reconstruction problem (and its implications in terms of theoretical positioning performances), our goal here is first to determine empirically from real measurement data the key parameters of the path loss model introduced in Eq. 2.2, namely n, P_0^{dBm} and σ, conditioned on propagation conditions. The objective is indeed two-fold. First, we want to observe in practice the global trends of the received signal strength in our concrete experimental context and set-up, as a function of both the transmission range and the operating conditions (e.g., the dispersion over LoS/NLoS conditions or over serving Base Stations, the practical ranges for reaching receiver sensibility or saturation, etc.). This will indeed be helpful to qualitatively interpret the dominating factors impacting the received signal dynamics (as a function of space). One more point is also to feed an analysis based on the evaluation of theoretical positioning performance bounds, while relying on synthetic models with representative radio parameters (See next section). For this purpose, we perform Least Squares (LS) data fitting out of real field RSS measurements from the Grenoble pre-processed datasets. Let us denote $\mathbf{m}^{BS} = (m_1^{BS}, m_2^{BS}, ..., m_K^{BS})^\intercal$ the vector of K measured values of the signal strength for the corresponding base station BS at distances d_k^{BS}, $\mathbf{P}(\eta^{BS}) = \mathbf{A}^{BS}.\eta^{BS}$ the vector of noise-free signal strength values calculated for the same distances d_k^{BS} based on Equation 2.2. For each base station BS, it hence comes:

$$\mathbf{m}^{BS} = \mathbf{P}(\eta^{BS}) + \epsilon^{BS} \tag{2.15}$$

$$\underbrace{\begin{pmatrix} m_1^{BS} \\ m_2^{BS} \\ \vdots \\ m_K^{BS} \end{pmatrix}}_{\mathbf{m}^{BS}} = \underbrace{\begin{pmatrix} 1 & -10.\log_{10}(\frac{d_1^{BS}}{d_0^{BS}}))) \\ 1 & -10.\log_{10}(\frac{d_2^{BS}}{d_0^{BS}}))) \\ \vdots & \vdots \\ 1 & -10.\log_{10}(\frac{d_K^{BS}}{d_0^{BS}}))) \end{pmatrix}}_{\mathbf{A}^{BS}} \underbrace{\begin{pmatrix} P_0^{BS} \\ n^{BS} \end{pmatrix}}_{\eta^{BS}} + \underbrace{\begin{pmatrix} \epsilon_1^{BS} \\ \epsilon_2^{BS} \\ \vdots \\ \epsilon_K^{BS} \end{pmatrix}}_{\epsilon^{BS}} \tag{2.16}$$

where $\epsilon_k^{BS} \sim N(0, (\sigma^{BS})^2)$ are assumed to be independent and identically distributed residual noise terms (i.i.d.), d_0^{BS} is the reference distance, $d_k^{BS} = \sqrt{(x^{BS} - x_k)^2 + (y^{BS} - y_k)^2}$ is the distance from the base station of 2D Cartesian coordinates (x^{BS}, y^{BS}) to some tag position (x_k, y_k).

To compute the required model parameters, we thus need to solve the set of equations for all the measured points by minimizing the sum of squared errors, as follows:

$$(\mathbf{m}^{BS} - \mathbf{A}^{BS}\eta^{BS})^{\intercal}(\mathbf{m}^{BS} - \mathbf{A}^{BS}\eta^{BS}) \rightarrow \min_{\eta^{BS}} \tag{2.17}$$

So the set of optimal parameters is calculated as follows :

$$\hat{\eta}^{BS} = ((\mathbf{A}^{BS})^{\intercal}\mathbf{A}^{BS})^{-1}(\mathbf{A}^{BS})^{\intercal}\mathbf{m}^{BS} \tag{2.18}$$

Retrospectively and in first approximation, residuals can be interpreted as noise in Equation 2.2, so that the standard deviation parameter σ^{BS} is simply determined with the optimal model parameters and Equation 2.2:

$$\hat{\sigma}^{BS} = std(\mathbf{A}^{BS}\hat{\eta}^{BS} - \mathbf{m}^{BS}) \tag{2.19}$$

The results of this LS data fitting process for a few representative Base Stations of the Grenoble dataset are reported in Tables 2.6 and 2.7. As it was mentioned above, we compared two settings with or without removing 10% of the cells with the highest variances of measured RSS values per cell. It was done because of the possible errors or artefacts during the experimental data collection phase (e.g., erroneous GPS position assignment due to satellite visibility conditions) or for a very small amount of collected data that were spotted to have non consistent values (e.g., due to tag failures, etc.). As we can see, removing even such a modest amount of those incriminated pathological cells can significantly impact the computation of the path loss model parameters ruling the deterministic dependency of average RSS as a function of transmission range, while the standard deviation accounting for the dispersion of RSS measurements around this average model remains unchanged. By the way, for all the considered Base Stations, the standard deviation value *std* is observed to be high, which could be imputed to the fact that the underlying average path loss model is too inaccurate and can hardly account for so complex propagation phenomena. Moreover, the amount of points is not so high (maximum around 10% of all the zone of interest), so that data fitting could be degraded by the sparseness and/or non-uniform distribution of the measurement points (with respect to the distance to the base station), which could lead to overweight the influence of some measurements in particular transmission range domains.

BS	$\hat{n}^{BS}$	$\hat{\sigma}^{BS}$, dB	$\hat{P}_0^{BS}$, dBm
B2I (BS_1)	3.20	8.31	-44.58
BCC (BS_4)	2.32	8.73	-55.21
Bastille (<500m) (BS_3)	2.34	7.87	-67.81
Bastille (>1 km) (BS_3)	2.87	8.07	-33.56
Christophe (BS_2)	4.01	11.90	-44.74

Table 2.6: Extracted path loss model parameters for some of the Base stations from Grenoble dataset, as shown on Figure 2.7b, with data aggregation in $10m \times 10m$ cells.

In Table 2.7, there are two sets of parameters for one of the base stations (namely "Bastille", BS_3), as we have identified two distinct zones in terms of topology (and

BS	$\hat{n}^{BS}$	$\hat{\sigma}^{BS}$, dB	$\hat{P}_0^{BS}$, dBm
B2I (BS_1)	3.46	7.90	-41.77
BCC (BS_4)	2.59	8.22	-52.18
Bastille (<500m) (BS_3)	2.31	8.01	-68.67
Bastille (>1 km) (BS_3)	2.10	8.62	-51.67
Christophe (BS_2)	3.65	11.18	-50.49
$BS_1 \cup BS_2 \cup BS_4$	2.89	10.51	-51.88

Table 2.7: Recomputed path loss model parameters (after removing 10% of the data) for the same Base stations from Grenoble dataset, as shown on Figure 2.7b, with data aggregation in $10m \times 10m$ cells.

hence, two propagation regimes): (i) on the hill hosting the base station, with a larger angular distribution of the incoming radio signals from the tags and (ii) in the rest of city where the direction of arrival of the signal at the base station does not vary much (and accordingly the receive antenna gain). Moreover, the position of the base station could not cover all the area around, similarly to BS "Christophe" (BS_2), being located on the Western border/part of the city, this base station naturally serves tags whose transmitted signals arrive systematically from the same side of the city (hence with a reduced span for possible angles of arrivals accordingly, which could somehow bias our extracted statistics). Another remark is that the path loss exponent n can differ significantly from one base station to another and is usually larger in more complex environment contexts (i.e., in case denser buildings are present in the surroundings of the BS), inducing more probable NLoS propagation conditions, and hence stronger shadowing effects even at short distances, which lead to lower RSS values as a function of the distance in average. But the dispersion around the fitted path loss model, as accounted here by the standard deviation σ, is also very high on its own, primarily due to a relative lack of accuracy of the single-slope path loss model, but also to the remaining effect of instantaneous received power fluctuations even after in-cell measurements averaging (e.g., caused by multipath under mobility, fast changing tag orientation during measurements collection...).

In Figure 2.10 below, we also show the overall RSS distribution over 3 base stations in town (all except the BS "Bastille", which again experiences a specific propagation regime due to the terrain elevation) as a function of the Tx-Rx distance, along with the superposed fitted function according to the path loss model. As expected, the dispersion accounted by the standard deviation is thus even worse here than that of previous BS-wise parameters extractions, while the other path loss parameters are close to their average values over the three considered base stations.

Illustration of RSS distribution along a LOS street As an illustration, we consider the longest street in LOS conditions served by the Bastille Base station (Figure 2.7a), where it is theoretically easier to model the behaviour of the signal distribution as a function of transmission range, due to the absence of any big obstacles on the way of the signal. Considering again the path-loss model of Section 2.1,

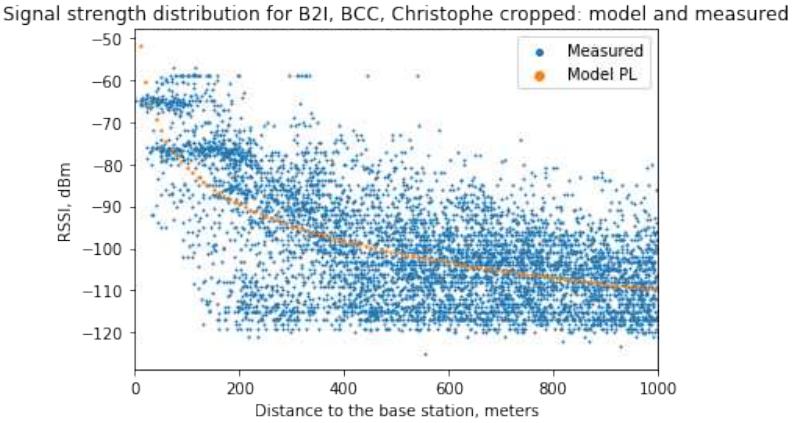
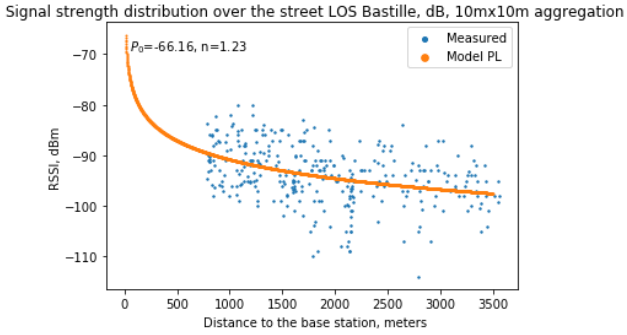


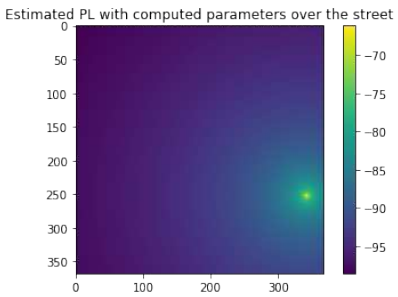
Figure 2.10: RSS distribution as a function of transmission range, over 3 base stations: measured values (blue) vs. model-based prediction (orange) with extracted parameters P_0 = -51.88dBm, n=2.89, σ = 10.51 dB.

the parameters have been determined through data fitting and the resulting model is confronted to the measurements, as shown in Figure 2.11. As we can see, the log-distance classical path-loss model fits fairly well the collected data in terms of average trend, but still with a very large dispersion around the expected/predicted value, indicating also that conventional parametric model-based positioning (i.e., not based on fingerprinting) would be challenging despite the LoS conditions.

As in this LOS case we can model the signal behaviour, then it is possible to com-



(a) RSS measurements distribution and model-based 1D average prediction in LoS, as a function of transmission range



(b) Model-based 2D prediction in LoS, as a function of tag's position, based on extracted path loss model parameters.

Figure 2.11: Illustration of RSS distribution over one particular street: measured RSS values vs. model-based line-of-sight predictions.

pare the 2D spatial RSSI gradients based on this theoretical model with that obtained after applying candidate interpolation methods and then use these gradients, as one of the quality metrics in the map reconstruction problem. The corresponding results will be later shown in Section 4.3.

2.4 Positioning Error Bound Illustrations and Qualitative Analysis

2.4.1 Numerical PEB illustrations for positioning based on the parametric path loss model

Leveraging the analytical bound expressions from 2.2.1, as well as the experimental model parameters per BS from Table 2.7 and the real BSs deployment described in section 2.3.3, we first illustrate through numerical examples the theoretical performance of positioning when based on a parametric link-wise path loss model (mostly for future benchmark with the fingerprinting approach).

In particular, Figure 2.12 shows PEB results, represented as heatmaps (i.e., displaying the theoretical 2D positioning error as a function of the possible tag position, given a certain BSs deployment), for 4 or 3 base stations whose positions correspond to some of the actual deployment in the city of Grenoble. One can hence clearly notice that the theoretical positioning error is not homogeneous but spatialized over the entire considered area, as the latter increases quite fast with the distance from the BS(s), when the number of deployed BSs decreases, and/or due to geometric dilution of precision (GDoP) phenomena (e.g., the error being larger when the tag lies on a line colinear to that formed by the deployed BSs, or more generally, out of the convex hull formed by the BSs). Beyond, from section 2.2.1, we know that the Fisher Information also depends on the ratio n_i/σ_i per link, so the contribution to the positioning error brought by one particular BS is all the minor since the standard deviation of its residual noise n_i (see Equation 2.19) is low and/or its path loss exponent is large. All in all, given our extracted measurements-based path loss parameters, it seems that a positioning error of several hundreds of meters could be achieved in the most optimistic case, by the way only in restrictive areas close to the BSs. The size of these areas also depends on the number of BSs involved in the positioning process.

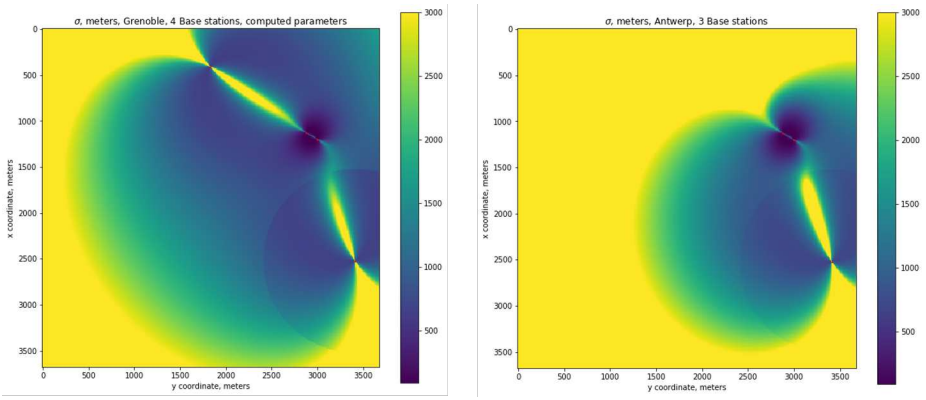


Figure 2.12: PEB (in meters) heatmap for parametric positioning based on the extracted experimental path loss model parameters, as a function of number and positions of base stations (Left: 4 base stations, Right: 3 base stations). Areas in yellow exhibit a PEB value larger than 3000m.

2.4.2 Numerical PEB illustrations for positioning based on fingerprinting

Now, leveraging the analytic expressions of section 2.2.2), we consider evaluating the theoretical positioning performance in case of fingerprinting, based on a canonical synthetic scenario, given that we cannot work yet on interpolated maps (as the interpolation methods will be introduced in the following chapters). 2D RSS maps based on Ray-Tracing simulations have thus been used in a canonical deployment scenario including up to 4 BSs in a $1000m \times 1000m$ area from Paris dataset 2.3.1. In the map-based fingerprinting context of interest, those BS-wise RSS maps (incl. 2D-correlated shadowing) would hence represent the deterministic ground-truth information that should be ideally contained in the offline fingerprinting database. In this synthetic scenario, we perform a small-scale evaluation of the PEB by "zooming" on an emulated street in particular, which is supposed to be in LoS conditions with respect to one of the base stations (BS indexed 3), as shown in Figure 2.13. For this part of the image, the 2D RSS gradient values (see Figures 2.14b, 2.14c) numerically computed out of the synthetic RSS surfaces with respect to the different BSs (which are strongly involved in the FIM calculation, and hence, in the PEB, according to section 2.2.2) are quite high on the street sides (border discontinuity effects), with a significant impact on PEB locally.

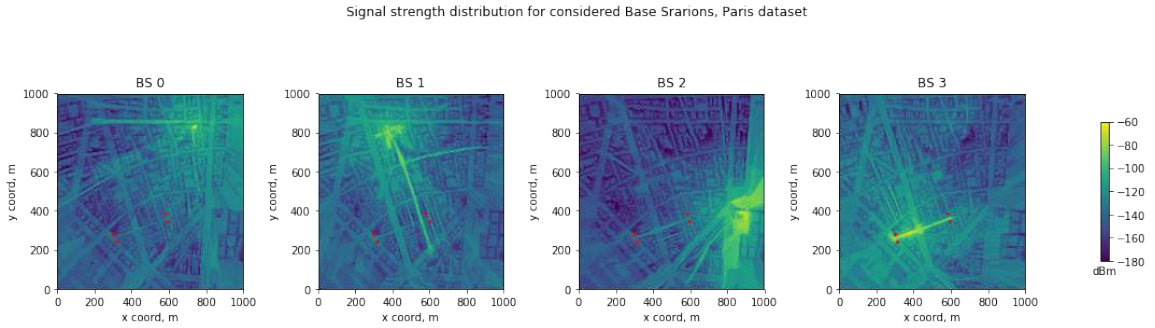
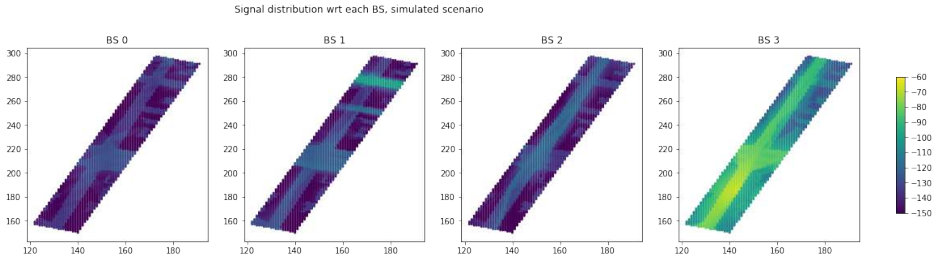


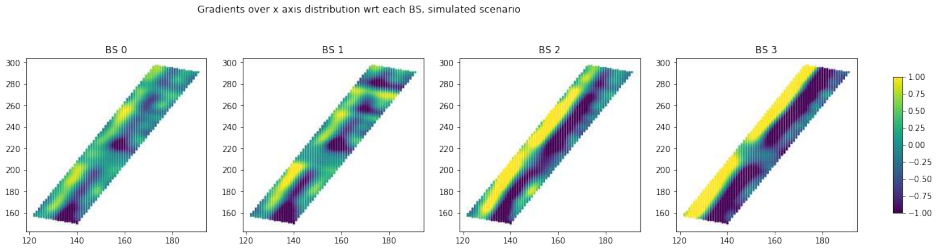
Figure 2.13: RSS maps based on Ray-Tracing from Paris dataset, with respect to 4 surrounding base stations (indexed from 0 to 3), along with the test street chosen for local PEB evaluation (between red dots).

It is then observed, as expected, that removing the base station in LoS increases the PEB along the emulated street and especially in the zone where the RSS value with respect to BS3 was high, while removing the other BSs leads to loose useful information mostly where their respective 2D RSS gradient values are locally higher, see Figure 2.14d. In terms of physical interpretation, these very large gradient values may be due to actual local transitions or discontinuities in the propagation regime (e.g., due to the presence of a cross-road intersection or a major obstacle such as a building, etc.), and/or to non-physical artefacts generated by the Ray-Tracing tool, thus biasing PEB calculation in the sense of an overoptimistic performance prediction.

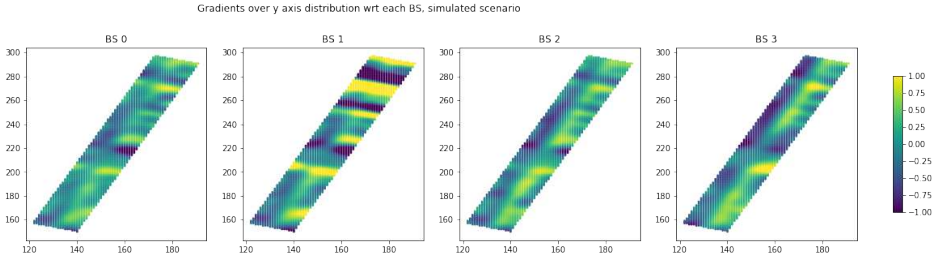
Beyond illustrating the importance of both the RSS 2D gradient values (on their own) and the reliability of their numerical evaluation, since LTE performance is not



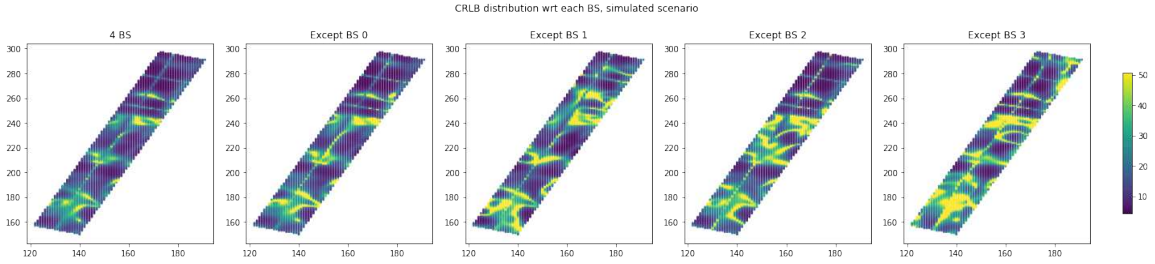
(a) RSS distributions as a function of tag position for the 4 considered Base Stations (in dBm).



(b) Corresponding RSS gradients, computed with respect to x coordinate (in dB).



(c) Corresponding RSS gradients, computed with respect to y coordinate (in dB).



(d) PEB heatmap (in meters), while discarding different Base Stations.

Figure 2.14: Test street extracted from Paris dataset, for 4 base stations (indexed from 0 to 3): RSS signal distribution, x and y gradients, computed PEB with different sub-sets of BSs (i.e., discarding one of the BSs at a time).

fully representative of that of the targeted IoT LoRa technology in terms of RSS dynamics (and thus, in terms of absolute orders of magnitude for the PEB), we thus provide another illustrative example based on synthetic RSS maps in a canonical deployment scenario including up to 4 BSs placed at the corners of a $1000m \times 1500m$ area (See Figure 2.15). Those maps have been generated by relying on the overall measurement-based experimental path loss model parameters from Grenoble dataset (See the parameters of the last line of Table 2.7, assuming $\sigma_i = \sigma \ \forall i \in \overline{1,4}$, while generating shadowing) and a 2D-correlated shadowing model. Like in the previous Ray-Tracing based Paris dataset, these synthetic RSS maps (incl. 2D-correlated shadowing) represent the ground-truth for fingerprinting. For this new test scenario, we have also performed a small-scale evaluation of the PEB along a test street, as shown in Figure 2.13. Again, one can note the same effect of removing sequentially one of the available BSs. We also remark that overall performance is degraded by one order of magnitude (in comparison with the former LTE ray-tracing predictions).

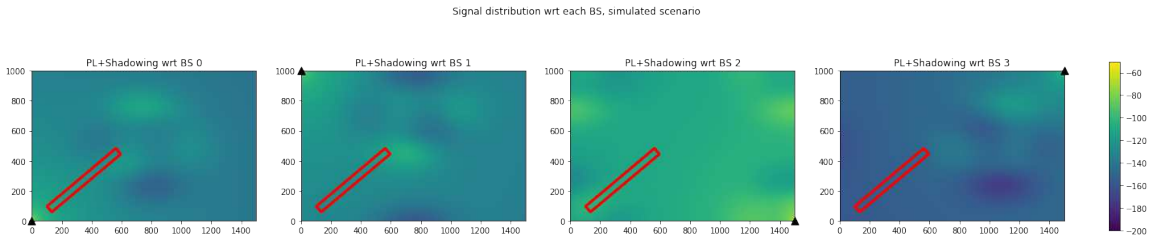


Figure 2.15: Synthetic RSS maps based on experimental path loss parameters extracted from the Grenoble dataset, with respect to 4 surrounding base stations (indexed from 0 to 3), along with the test street chosen for local PEB evaluation (red box).

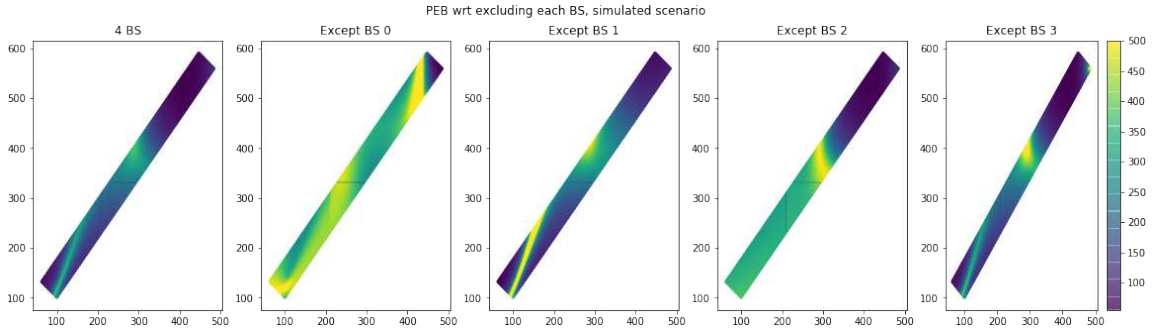


Figure 2.16: PEB heatmap (in meters) for LoRa fingerprinting along a test street, corresponding to the synthetic RSS maps of Figure 2.15, based on experimental path loss parameters extracted from the Grenoble dataset, while discarding different Base Stations.

Now, so as to assess the sensitivity of fingerprinting positioning performance with respect to RSS map interpolation accuracy (i.e., the quality of the map stored in the database), further random noise (say, equivalent to w in Eq. 2.9 of section 2.2.2) has

been artificially generated on top of the previous ground-truth RSS maps, while varying the standard deviations of this extra noise. This is intended to check the maximum tolerable interpolation error enabling to preserve the advantage of the data-aided fingerprinting approach in comparison with model-based parametric positioning, under equivalent deployment conditions. On Figure 2.17, we thus represent the resulting empirical cumulative density function (CDF) of all the PEB values over the considered test street of Figure 2.16. In this case, in approximately 80% (resp. 50%) of positioning attempts, the fingerprinting approach would hence lead to a 2D positioning error lower than 170m (resp. 150m) if a RSS map interpolation error lower than 3 dB (resp. 5-6 dB) could be committed (in comparison with the ground-truth map).

All in all and under these realistic experimental conditions, as for even more relaxed accuracy requirements (say, typically around 400m), one can hence assume in first approximation that an accuracy of 5-6dB should be ensured in the RSS map interpolation process (in comparison with the so-called ground-truth map) so that fingerprinting can systematically outperform conventional model-based parametric positioning, even when the latter is placed in favorable operating conditions (i.e., close to at least one BS; See Figure 2.12).

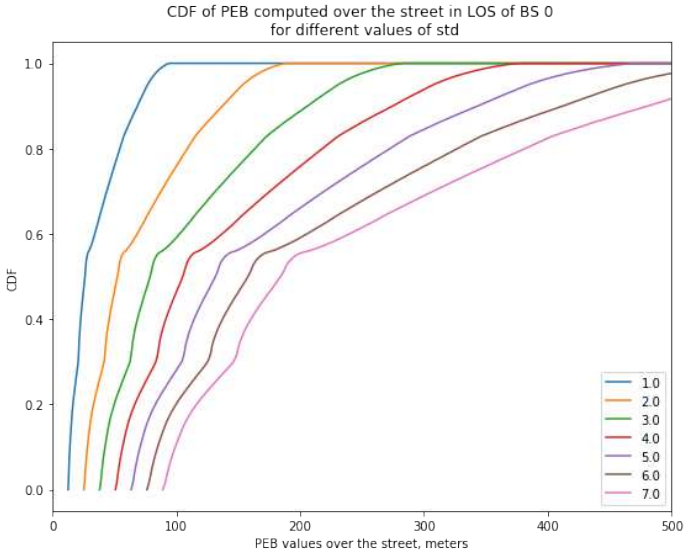


Figure 2.17: Empirical CDF of the PEB through fingerprinting over the tested area, for different standard deviations of the extra noise affecting ground-truth RSS maps generated based on experimental path loss model parameters from the Grenoble dataset (i.e., accounting for the quality of interpolated RSS maps).

2.5 Conclusion

In this section, by relying on both theoretical tools characterizing positioning performance bounds, as well as on experimental datasets made of real field measurements, we have analyzed the link between RSS measurement dynamics, main RSS map recon-

struction challenges, and finally positioning accuracy, under various RSS modelling options and in light of the final targeted fingerprinting application.

First, we have contributed to generate and process an extensive experimental dataset, based on real RSS measurements collected at multiple BSs deployed in the city of Grenoble, based on a plurality of mobile LoRa tags. The resulting database constitutes an outcome of the PhD work, on its own.

Then, we have also found out that:

- A $10m \times 10m$ size for the data aggregation cells, the inclusion of cells with at least 3 visited measurement points, as well as the rejection of 10% of presumed measurement artefacts from the experimental dataset, could be adequate for further processing, and more specifically, for RSS map interpolation. These parameters were hence used in the upcoming Chapters 4 and 5;

- The 2D gradients of RSS maps are dominating the PEB. Thus, their behaviour will have to be carefully analyzed, as an integral output and quality indicator of the interpolation methods tested in Chapters 4 and 5;

- Ideally, as a rough order of magnitude, the RSS map interpolation error should be limited to a few dBs (typically, 5-6 dB) so as to preserve the benefits from the deterministic positioning approach of fingerprinting (over more conventional model-based parametric positioning);

- Removing one of the base stations obviously affects the final localization accuracy both in conventional parametric and fingerprinting-based positioning approaches, and the impact will be all the more significant since the propagation regime with respect to this base station is favorable (typically, in Line-of-Sight), as expected.

Chapter 3

Supervised and Semi-supervised Learning

3.1 Introduction

Machine Learning (ML) is a branch of computer science that allows machines to make predictions without being specifically trained to do so. This is accomplished through inference, which entails training a machine over a finite collection of instances, known as the training set, and then applying the model to new data. Because it includes information from several research areas, inference in informatics has drawn many scholars from various study domains such as psychologists, neuroscientists, computer scientists and statisticians, to mention a few.

In recent years, the area has seen a rise in interest, owing to increased availability of computational resources, the emergence of open-source machine learning packages, and large-scale data sets. Due to the ability to improve existing results, find underlying hidden patterns, or solve complex mathematical problems, ML has proven to be valuable in many application areas such as natural language processing, computer vision, medicine, banking, social networks, signal processing, digital advertising, and so on.

Depending on the problem, available amount of the data and its nature, etc. there exist different types of learning paradigms. one can roughly divide it into following three main groups: supervised, semi-supervised, unsupervised learning.

* The goal of supervised learning is to identify the association between inputs and their *desired* outputs by using a training set that comprises a finite number of pairs of instances and their outputs. When the outputs are discrete, the problem is known as *classification*; when the outputs are continuous, the problem is known as *regression*.

* Unsupervised learning concerns situations where the outputs of examples are not available. In this situation, the main objective is to find groups that contain similar examples by exploiting the structure of the training data, and the task is referred to as *clustering*.

* Semi-supervised learning is a framework that sits in the middle of the two previous frameworks. The goal here is to identify the relationship between the input instances and their outputs by using both the label information in the labeled training set and the structure of the unlabeled training data.

In this chapter we will provide a brief overview of supervised learning and semi-supervised learning that correspond to the frameworks of our study. In the reminder of the chapter, we will present supervised learning in Section 3.2 by explaining the two key principles that underpin the construction of all supervised learning models. We will go through the Neural Networks models that we used in this thesis in particular. We introduce the semi-supervised learning (SSL) framework in Section 3.3, by outlining the three fundamental assumptions that support the construction of (SSL) models.

3.2 Supervised Learning

In the supervised learning approach, algorithms try to fit/find the mapping function (or general predictor) between an input space $\mathcal{X}$ and an output space $\mathcal{Y}$. From a training set where each example is composed of a pair *(observation, output)*, the goal is then to find this mapping which has the lowest probability of error on new examples that were not been observed.

3.2.1 Fundamental principles

In practice, a loss function, measureing the (dis)agreement between prediction and desired output (also known as *label*) is defined, and the learning algorithm selects the prediction function that minimizes the average error on the training samples, referred to as *empirical risk*. This is called the *Empirical Risk Minimization* (ERM) principle. We hope that by reducing the empirical risk, the prediction function will have a low *generalization* error, which means that it will make few errors on average on new examples. The fundamental assumption is that the new instances are identical to the training examples used to find the prediction function. Hence, the study of the relationship between empirical error and generalization error is at the basis of statistical learning theory [Vapnik, 1998].

Definition and notations We will start with certain definitions and notations that will be used throughout this thesis. Each example is constituted of an observation and a desired output. For a fixed $d \in \mathbb{N}^*$, each observation is represented by a set of characteristics defined in a vectorial space $\mathcal{X}$, which is often $\mathcal{X} \subset \mathbb{R}^d$. The corresponding output will be referred to as the *desired output*, and it is considered to be part of an output set $\mathcal{Y}$. In this case, a pair (x, y) designates an element of $\mathcal{X} \times \mathcal{Y}$.

The central assumption in Machine Learning is that all examples are distributed independently and identically (i.i.d.) by a fixed yet unknown probability distribution $\mathcal{D}$. Thus for any set $\mathcal{S}$, its examples $(x_i, y_i) \in \mathcal{S}$ are supposed to be generated i.i.d;

according to $\mathcal{D}$. We then say that $\mathcal{S}$ is a i.i.d. sample following $\mathcal{D}$. Informally, this hypothesis defines the notion of representativeness of a training set as well as a test set in relation to the problem: the training examples, and also test observations with their desired outputs, are generated i.i.d from a same probability distribution.

The notion of error, often known as risk or loss, is the second fundamental concept in machine learning. Given a class of functions $\mathcal{F} = \{f \mid f : \mathcal{X} \to \mathcal{Y}\}$, the agreement between the prediction $f(x)$ for a given observation $x \in \mathcal{X}$ and its desired output y is assessed using a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$. Intuitively, $\ell(f(x), y)$ quantifies how similar the expected and desired outputs are. As a result, it is usually defined as a distance across the set of outputs $\mathcal{Y}$.

In binary classification, where $\mathcal{Y} = \{-1, +1\}$, the most common loss is the 0/1 loss defined as:

$$\ell(f(x), y) = \mathbb{1}_{f(x) \neq y}, \tag{3.1}$$

where, $\mathbb{1}_\pi = 1$ is predicate π is true and 0 otherwise.

In regression, where $\mathcal{Y} = \mathbb{R}$, the most common losses are the absolute error :

$$\ell(f(x), y) = |f(x) - y|, \tag{3.2}$$

and the square loss:

$$\ell(f(x), y) = (f(x) - y)^2. \tag{3.3}$$

Generalization and empirical errors From this definition, the generalization error of a prediction function $f \in \mathcal{F}$ is expressed as

$$\mathcal{L}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big(\ell(f(x), y)\big) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(f(x), y) \mathrm{d}\mathcal{D}(x, y) \tag{3.4}$$

The function $f \in \mathcal{F}$ we are interested in is the one that produces the fewest prediction errors on new data, and hence, has the lowest generalization error. On the other hand, this generalization error cannot be computed since the probability distribution $\mathcal{D}$ is unknown. [Vapnik, 1998] showed that by minimizing the average error of f on a training set $\mathcal{S} = (x_i, y_i)_{1\leq i\leq m}$ that is generated i.i.d. from $\mathcal{D}$, the search for the function f may be done in a consistent manner. This amount is known as the empirical risk of f on $\mathcal{S}$ and it is an unbiased estimate of the generalization error.

$$\hat{\mathcal{L}}_m(f, S) = \frac{1}{m} \sum_{i=1}^{m} \ell(f(x_i), y_i) \tag{3.5}$$

Empirical Risk Minimization A learning algorithm receives a training set $\mathcal{S}$ as input and outputs the prediction function $f_S : \mathcal{X} \to \mathcal{Y}$. A learning algorithm is a function $\mathcal{A}$ that searches for the function f_S among a collection of functions $\mathcal{F}$, which is referred to as a class of functions. The ERM algorithm is intuitively understood as follows.

If the training instances in $\mathcal{S}$ are sufficiently representative of the distribution $\mathcal{D}$, then the empirical error $\hat{\mathcal{L}}_m(f, \mathcal{S})$ is a good estimate of the generalization error $\mathcal{L}(f)$.

We will thus minimize the empirical error on a particular training set $\mathcal{S}$ in order to minimize the generalization error. The ERM principle returns the function f_S when given an loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, a training set $\mathcal{S}$ comprising m samples, and a class of functions $\mathcal{F}$.

$$f_S = \arg\min_{f \in \mathcal{F}} \frac{1}{m} \sum_{(x_i, y_i) \in \mathcal{S}} \ell(f(x_i), y_i) \tag{3.6}$$

In practice, a real valued prediction function $h_{\mathbf{w}} : \mathcal{X} \to \mathbb{R}$ is learned over a continuous differentiable upper bound of the empirical loss (3.6) (called the surrogate loss) and the classification function is then defined as the sign of this function; i.e. $\forall x \in \mathbb{R}, f(x) = \text{sgn}(h_{\mathbf{w}}(x))$.

Generalization and consistency of ERM principle Let us emphasize that minimizing empirical error is not an objective in itself; what we are interested in is minimizing generalization error. As a result, the ERM approach is useless if the learnt function f_S has a low empirical error but a high generalization error. As a result, we may anticipate the ERM method to *generalize*, implying that the empirical error of f_S is a reasonable predictor of its generalization error. If this condition of *generalization* remains true, we may safely assume that if ERM provides a low empirical error function f_S, its generalization error will be low as well.

Let us further emphasize that the ERM method operates in a given and fixed set of functions $\mathcal{F}$ where the function having the lowest generalization error is searched. As a result, a second inherently desired aspect of the ERM method is that, given enough instances to learn from, it finally discovers the optimal function of $mathcal F$ (having the lowest generalization error). This is called the *consistency* of the ERM principle.

Overfitting and complexity of a class of functions For several classes of functions, the empirical error of the learnt function f_S is not a reliable predictor of the generalization error. In this sense, it is preferable to keep the learnt function as simple as possible in comparison to the training examples. Indeed, a function with a null empirical loss on a training set and arbitrarily large generalization error is simple to come by. The term *overfitting* describes this occurrence.

As a result, we want the ERM algorithm to be able to learn simple functions. Constraining the class of functions $\mathcal{F}$ to only contain simple functions is one technique to impose simplicity (the notion of simplicity remains to be defined). By doing so, we can demonstrate that the notions of generalization and consistency of the ERM principle are equivalent: we can ensure the generalization and consistency of the ERM algorithm by restricting the complexity of the class of functions $\mathcal{F}$.

On the other hand, if $\mathcal{F}$ is too simple in comparison to the distribution $\mathcal{D}$, the learnt function will most likely not generalize well. It will have a high empirical error as well as a high generalization error. As a result, the choice of the set of functions $\mathcal{F}$ is critical: it must be neither too complicated to avoid the problem of overfitting nor too basic to avoid the problem of *underfitting* while yet achieving good generalization

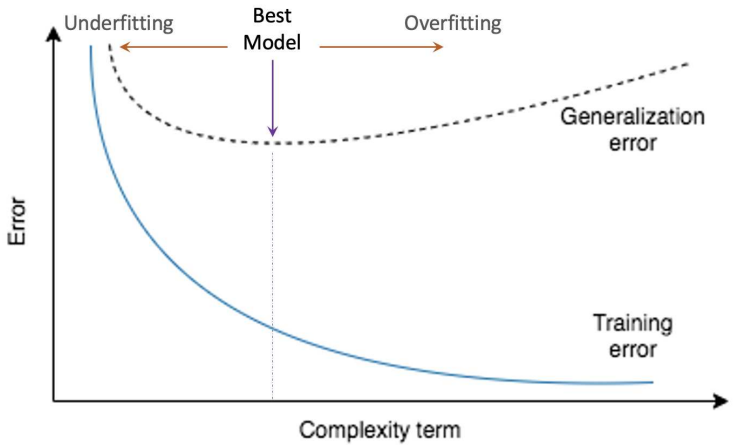


Figure 3.1: Empirical error and generalization error with respect to the complexity of the class of functions. The best class of functions where the prediction function has to be found is the one which has the best compromise between high complexity and low empirical error.

performance. The *bias-variance* trade-off, often known as the trade-off between low empirical error and a complicated class of functions, is crucial in machine learning. This notion is shown in Figure 3.1

Rademacher complexity The Rademacher complexity, is a fundamental measure of the complexity of a class of functions introduced by [Bartlett and Mendelson, 2002] for binary classification. This measure estimates the ability of a class of functions $\mathcal{F} = \{f : \mathcal{X} \to \{-1, +1\}\}$ to learn from a randomly noisy training set. Let $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_m\}$ be a set of m binary random variables, where each $\sigma_i \in \boldsymbol{\sigma}$, called the Rademacher variable, takes a value -1 or $+1$ with probability $\frac{1}{2}$; i.e. $\forall i \in \{1, \ldots, m\}; \mathbb{P}(\sigma_i = -1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}$. Then the empirical Rademacher complexity of $\mathcal{F}$ over a training set $S = (x_i, y_i)_{1 \leq i \leq m}$ of size m is defined as:

$$\hat{\mathfrak{F}}_m(\mathcal{F}, S) = \frac{2}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \left| \sum_{i=1}^{m} \sigma_i f(x_i) \right| \right], \tag{3.7}$$

As a result, the capacity of the class of function $\mathcal{F}$ to fit random noise increases as Rademacher complexity increases. The corresponding Rademacher complexity is then defined as

$$\mathfrak{F}_m(\mathcal{F}) = \mathbb{E}_S[\hat{\mathfrak{F}}_m(\mathcal{F}, S)]. \tag{3.8}$$

Generalization bounds The statistical learning theory is concerned with the link between empirical error, generalization error, and the complexity of a class of functions. The majority of these works take the form of probabilistic error bounds, such as the Rademacher generalization bound, which provides an upper bound on the generalization error that holds with high probability with respect to the empirical

error, the complexity of the considered class, and some residual term that controls the bound's precision.

Theorem 1 (Generalization bound [Bartlett and Mendelson, 2002]). *Let $\mathcal{X} \in \mathbb{R}^d$ be a vectorial space and $\mathcal{Y} = \{-1, +1\}$ an output space. Suppose that the pairs of examples $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are generated i.i.d. with respect to the distribution probability $\mathcal{D}$. Let $\mathcal{F}$ be a class of functions having values in $\mathcal{Y}$ and $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ a given instantaneous loss. Then for all $\delta \in]0, 1]$, we have with probability at least $1 - \delta$ the following inequality :*

$$\forall f \in \mathcal{F}, \mathcal{L}(f) \leq \hat{\mathcal{L}}_m(f, S) + \mathfrak{F}_m(\ell \circ \mathcal{F}) + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}} \tag{3.9}$$

Using the same steps we can also show that with probability at least $1 - \delta$

$$\mathcal{L}(f) \leq \hat{\mathcal{L}}_m(f, S) + \hat{\mathfrak{F}}_m(\ell \circ \mathcal{F}, S) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2m}} \tag{3.10}$$

Where $\ell \circ \mathcal{F} = \{(x, y) \mapsto \ell(f(x), y) \mid f \in \mathcal{F}\}$. To have a low risk, the two terms on the right of these inequalities (3.9) or (3.10) must be low: the empirical error, which is dependent on the prediction function f, and the second term, which is dependent on the complexity of the class of functions $\mathcal{F}$. To have a theoretical guarantee on the generalization error, it is therefore required to take a class of functions that is not excessively complicated in addition to minimizing the empirical error (having a high complexity term). However, it should not be overly simplistic, since this will result in a large amount of empirical inaccuracy. As a result, we find out the previously mentioned bias-variance trade-off.

Structural Risk Minimization The choice of the class of functions, as previously stated, is the most challenging aspect of supervised learning since it is this choice that implements the bias-variance trade-off. The preceding generalization bound, on the other hand, gives a straightforward technique for determining the appropriate class of functions. Consider many classes of candidate functions with known Rademacher complexity $\mathcal{F}_1, \ldots, \mathcal{F}_N$. The ERM technique may be used to discover a function for each class, and then the value of the bound on the generalization error can be calculated. Among the classes of candidate functions, the class of functions that minimizes this constraint has the strongest theoretical guarantee on the error in generalization. As a result, it is only logical that we choose this class of functions. This is called the structural risk minimization (SRM) principle [Vapnik, 1998]. The SRM principle is used in practice by concurrently reducing the norm of the function to be determined and the empirical loss across a given hypotheses space $mathcal F$:

$$\forall S; f_S = \text{argmin}_{f \in \mathcal{F}} \hat{\mathcal{L}}_m(f, S) + \lambda \|f\|^2,$$

where λ is a hyperparameter of the regularization term that balances complexity with empirical loss. The two principles of empirical risk minimization and strucutral minimization risk are at the origin of a large number of learning algorithms.

3.2.2 Neural Networks

In this section, we introduce neural networks, which are unquestionably one of the most prominent classification algorithms in the field of Machine Learning. These models are also used in our contributions, which are detailed in the next part.

A neural network is a set of elementary units, called artificial neurons, linked together and allowing to perform different non-linear transformations. These models come from the work of neuroscientists who, in the 1950s, began to work on mathematical models conceptualizing the functioning of natural neurons. These models had some success until the 1990s, when the field of machine learning experienced a new upheaval with the statistical learning theory proposed by [Vapnik, 1998]. During this period, little advancement has been made in the application and use of neural networks. Technological advances in hardware components in recent years, however, have led to a renewed interest in learning with neural networks. In particular, the new graphics processors (GPU - Graphics Processing Unit) and the increase in the capacity of RAMs have allowed the design and training of multilayer neural networks with more complex architectures, known as the deep learning. Since 2010, we have witnessed a major breakthrough in the application of these deep neural networks in several key areas of computer science such as computer vision or automatic natural language processing. The purpose of this chapter is to present the basic principles of learning these networks.

We will consider here an extremely general modular architecture in which a network implements a sequential composition of functions, each corresponding to a layer and being defined by its own set of parameters.

We consider the simple sequential case where the function h_W or $h(W)$ computed by the model and corresponding to the transformation performed globally by the network is the composition of N functions $h_n(W_n)$ with $1 \leq n \leq N$:

$$h_W = h^N_{W_N} \circ ... \circ h^2_{W_2} \circ h^1_{W_1} \tag{3.11}$$

We note here $h_W(x)$ is the prediction provided by the model for a given x in input. The composite function is determined by the vector W which is the concatenation of the vectors W_n with $1 \leq n \leq N$. Functions h^n may depend differently on their vector of parameters W_n and these vectors may have different sizes. Vectors W_n can also encapsulate more complex structures, such as matrices or weight tensors, but at the level considered here they will only appear as vectors, the structures they contain being always serialized. The functions h^n can be arbitrary provided that they are differentiable with respect to their data and with respect to their parameters.

In the classical presentation of neural networks, the application of a function generally corresponds to a layer of formal neurons. In practice however, a layer of neurons in this classical sense is often implemented in two successive functions, corresponding to two different types of layers (or sub-layers) : one, called *linear layer*, which corresponds strictly to the product of a matrix by a vector for the linear part and the other, called *point-to-point layer*, which corresponds to the application of the activation function component by component for the non-linear part. Additional functions, normalization or regularization for example, are also sometimes inserted.

The back-propagation method described below for training this type of network is very general and applies to any combination of differentiable and parameterizable functions. We describe here only the case of a sequential combination of functions, but the method is in fact even more general because it applies to combinations of functions assembled according to any acyclic graph.

Objective-function and stochastic gradient descent by mini-batches To train a network defined by a global function h_W depending on a vector of parameters W, one generally uses a method based on stochastic gradient descent by mini-batches. Considering a training set $\mathcal{S} = (x_i, y_i)_{1 \leq i \leq m}$; we seek to minimize a *regularized* objective function following the SRM principle :

$$\hat{\mathcal{L}}_m(h_W, \mathcal{S}) = \frac{1}{m} \sum_{(x,y) \in \mathcal{S}} \ell(h_W(x), y) + \lambda \Omega(W) \tag{3.12}$$

where the loss function ℓ is assumed to be continuously differentiable, and $\Omega(W)$ is a regularization term whose purpose is to prevent overfitting. The function Ω is also assumed to be continuously differentiable and the parameter λ is used to determine the relative *impact* of the regularization term.

In the stochastic mini-batch version of the gradient descent algorithm, one updates the vector W based on a subset of the training set $\mathcal{M}_S \subseteq \mathcal{S}$, called mini-lot, according to the formula:

$$W^{(t+1)} = W^{(t)} - \frac{\eta}{\mid \mathcal{M}_S \mid} \sum_{(x,y) \in \mathcal{M}_S} \nabla_W \mathcal{L}_{(x,y)}(W^{(t)}) \tag{3.13}$$

with :

$$\mathcal{L}_{(x,y)}(W) = \ell(h_W(x), y) + \lambda \Omega(W) \tag{3.14}$$

For example, $\ell(.)$ could be the square loss and $\Omega(.)$ the squared norm of the weights :

$$\mathcal{L}_{(x,y)}(W) = \frac{1}{2}(h_W(x) - y)^2 + \frac{\lambda}{2}\|W\|^2 \tag{3.15}$$

To implement gradient descent as described in the equations (3.13) and (3.14), we need to be able to compute $\nabla_W \ell(h_W(x), y)$ for the example (x, y). Note that, for the implementation of equations (3.13) and (3.14), the calculation of $\nabla_W \Omega(W)$ generally does not pose a problem. For example, if $\Omega(W) = \frac{1}{2}\|W\|^2$, then $\nabla_W \Omega(W) = W^\top$. This regularization is equivalent to applying a decay rate $\eta\lambda$ to the components of W at each iteration t (referred to as *weight decay* in the literature).

For the composite function, described in the previous section (3.11), we will note x^n with $0 \leq n \leq N$ the successive values of the input data x during the successive application of functions $h^n_{W_n}$ over the N layers. We will have $x^0 = x$ and $x^n = h^n_{W_n}(x^{n-1})$ for $1 \leq n \leq N$. $\ell(x^n, y)$ measures the error between the predicted value for input x and the desired output y. W being the concatenation of vectors W_n with $1 \leq n \leq N$, $\nabla_W \ell(x, y)$ is the concatenation of $\nabla_{W_n} \ell(x^n, y)$ which we will denote in a simplified way $\frac{\partial \ell}{\partial W_n}$.

The pieces of gradient $\frac{\partial \ell}{\partial W_n}$ which make up the global gradient $\nabla_W \ell(x, y)$ are calculated using the so-called *back-propagation* algorithm that comprises three parts.

- The first part, called *forward pass* or *spread*, corresponds to the normal operation of the network in prediction mode. In this pass, the values x^n are calculated by recursively from the current input $x^0 = x$ by successive application of the functions $h^n_{W_n}$. Additionally, the residual error (or associated cost) is calculated as $\ell(x^n, y)$.

- The second part, *backpropagation according to data*, is done as part of a *back pass*. In this part of the backward pass, we calculate by recurrence for decreasing values of n starting from N, the gradients with respect to the data x^n from error ℓ by using the chain rule :

$$\delta_N = \frac{\partial \ell(x^N, y)}{\partial x^N} \tag{3.16}$$

$$\forall n \in \{N, \dots, 2\}; \delta_{n-1} = \frac{\partial \ell(x^N, y)}{\partial x^{n-1}} = \frac{\partial \ell(x^N, y)}{\partial x^n} \frac{\partial h^n_{W_j}(x^{n-1})}{\partial x^{n-1}} \tag{3.17}$$

  From the definition, $x^n = h^n_{W_n}(x^{n-1})$.

- The third part, called *back-propagation according to parameters*, is considered as part of the backward pass but it does not involve a recurrence relation, the calculations not involving forward or backward dependence on the values of n when $\frac{\partial \ell(x^N, y)}{\partial x^n}$ has been calculated in the second part. The gradients with respect to the parameters W_n of the error ℓ are finally calculated by using the chain rule once again :

$$\begin{aligned} \forall n \in \{N, \dots, 1\}; \frac{\partial \ell(x^N, y)}{\partial W_n} &= \frac{\partial \ell(x^N, y)}{\partial x^n} \frac{\partial h^n_{W_n}(x^{n-1})}{\partial W_n} \\ &= \delta_n \frac{\partial h^n_{W_n}(x^{n-1})}{\partial W_n} \end{aligned} \tag{3.18}$$

3.2.3 Application of Neural Networks to RSSI map reconstruction

RSSI map reconstruction has recently seen a lot of use of neural networks. Most of the research entails estimating the signal's values based on known locations and employing side information [Hayashi et al., 2020, Nagao and Hayashi, 2020, Inoue et al., 2020, Suto et al., 2021, Zhu et al., 2020]. The radio parameters (such as transmission specifications or the connection between Rx and Tx) and building information (height and existence) are the most common side information used, and in the absence of the latter, they forecast it using satellite photos (like in paper [Inoue et al., 2020]). In these work, the problem is seen as a regression task, with data on the environment or technological characteristics as input and the value of the signal as output.

The signal map is considered as an image and the majority of these work adapted NN models that have been developed for image classification and segmentation.

Neural Network models for image analysis All Neural network models for image analysis use convolutional neurons that allow to extract high level features from row images. The main difference with the feedforward neural networks we have seen in Section 3.2.2), is that in the latter each layer consists of a set of formal neurons that are all connected to all the formal neurons of the previous layer (or all inputs for the first layer). In this *all to all* (or *fully connected*) connection scheme, all the weights of all the connections and all the biases of all the neurons are independent. Moreover, there is no topology in the organization of the neurons in the layers, these constituting a set and not a vector, a matrix or another structure, even if the input or output values are in practice collected in vectors of real numbers.

In the case of image analysis, the input data is organized according to a certain topology (a two-dimensional grid) and it is interesting to preserve this topology and to exploit it, at least in the first layers of the networks, to optimize the use of the available parameters and the computing capacity. This is done, on the one hand, by limiting the possible connections according to a maximum distance in the grid and, on the other hand, by sharing the weights between the units of the same layer which have different positions in the grid. This leads to operations of the convolution type, which are invariant under translation and which have limited support, instead of operations of the product type of a matrix by a vector. In fact, in practice, we combine the two. For this, we associate not a single neuron with each position in the grid, but a set of identical neurons from one position to another.

If the input data are organized according to a D-dimensional grid (i.e. $D = 2$ for an image signal), those manipulated by the input and output layers will have the same D topological dimensions plus a *non-topological* dimension which corresponds to the fact that this data is composed of multiple values at each point of the grid. The latter are found in the same way at each point of the grid where they constitute *cards*, each being a table with a value per position in the grid. The number of these cards is set independently for each step in the data processing.

For images, the input data X_{in} and output X_{out} are tensors of order 3, respectively of size $d_{in} \times w_{in} \times h_{in}$ and $d_{out} \times w_{out} \times h_{out}$; d_{in} and d_{out} are the number of cards in X_{in} and in X_{out}; $w_{in} \times h_{in}$ and $w_{out} \times h_{out}$ are the sizes of the grids on which X_{in} and X_{out} are defined. The linear part of a convolution layer is defined by:

$$X_{out}(l,i,j) = B(l) + \sum_{k=1}^{k=d_{in}} \sum_{m=-w}^{m=+w} \sum_{n=-w}^{n=+w} \kappa(l,k,m,n) X_{in}(k, i-m, j-n) \qquad (3.19)$$

for $1 \leq i \leq w_{out}$, $1 \leq j \leq h_{out}$ and $1 \leq l \leq d_{out}$; where w is the maximum distance in the grid at which points can be connected.

The d_{out} cards of X_{out} are calculated independently of each other and each from the set of d_{in} cards of X_{in}. An operator that calculates one of the output maps from the set of input maps is called a convolution *filter*. Each of these d_{out} filters is defined by a scalar $B(l)$ and by a tensor of order 3; $\kappa(l,.,.,.)$ called *kernel* of this filter. The size of this kernel is $d_{in} \times (2w+1) \times (2w+1)$ and we agree that the size of the filter is $(2w+1) \times (2w+1)$. The complete convolution layer is defined by the vector B and the kernel κ.

`UNet` model is a form of fully connected convolutional neural network, as mentioned above, with approximately symmetric architecture. It was first presented in [Ronneberger et al., 2015a] to handle the problem of image segmentation on medical data. It is based on the concept of encoder-decoder neural networks, which means that the encoder component of the network seeks to detect core patterns in the input and then transfers the extracted important information to the decoder part of the network. It consists of the convolutional and max pooling layers that follow. It performs deconvolution (also known as symmetric expanding route) in the decoder part, which entails upsampling the picture to the original input size using the retrieved information. Figure 3.2 depicts an example of such `UNet`-based architecture with skip connections for image reconstruction.

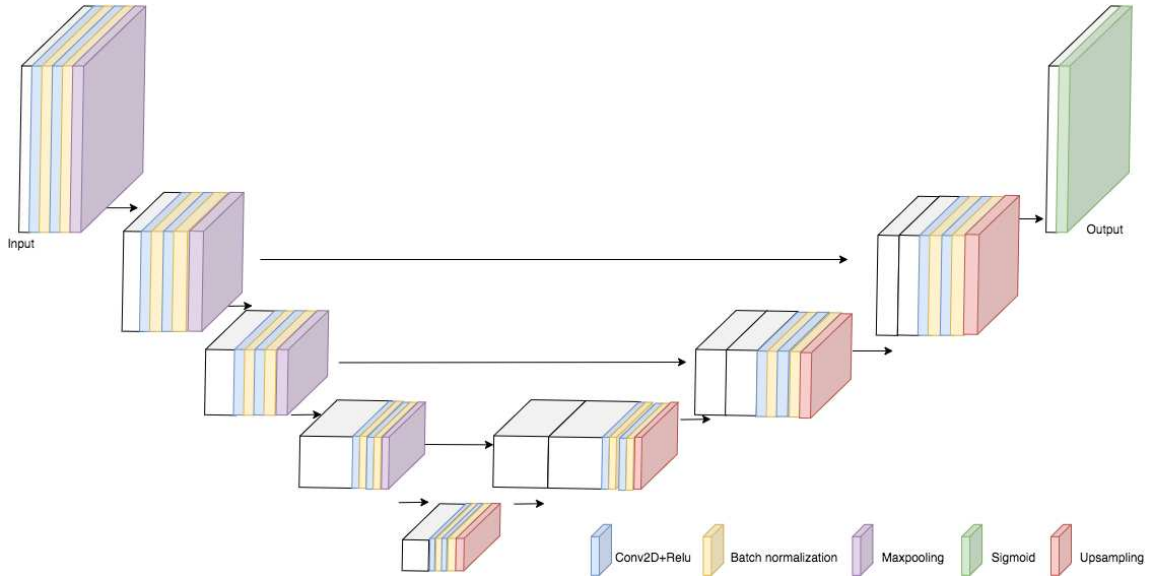


Figure 3.2: UNet-based architecture

`DIP` is a neural network model proposed for image reconstruction that is similar to `UNet` but with various modifications, such as the addition of convolutional, batch normalization, and activation layers at the top of the skip connections, see Figure 3.3, [Ulyanov et al., 2017]. The other, more basic distinction is that the model can only be trained on one image, making it more desirable for RSSI map reconstruction.

Generative adversarial networks is another technique that has been massively applied to image analysis, the model is based on a game in which two NNs referred to as, *Generator* and *Discriminator*, compete against each other [Goodfellow et al., 2014].

The task of the Generator is to generate observations, which are subsequently assessed by the Discriminator. In this sense, the Generator tries to learn the distribution p_G similar to the true distribution p_{data} of the input dataset. The input of this network is an input noise vector $\mathbf{z}$ from a fixed prior distribution $p_{\mathbf{z}}(\mathbf{z})$ and its output $G(\mathbf{z})$ is provided as input to the Discriminator, which also receive as input a

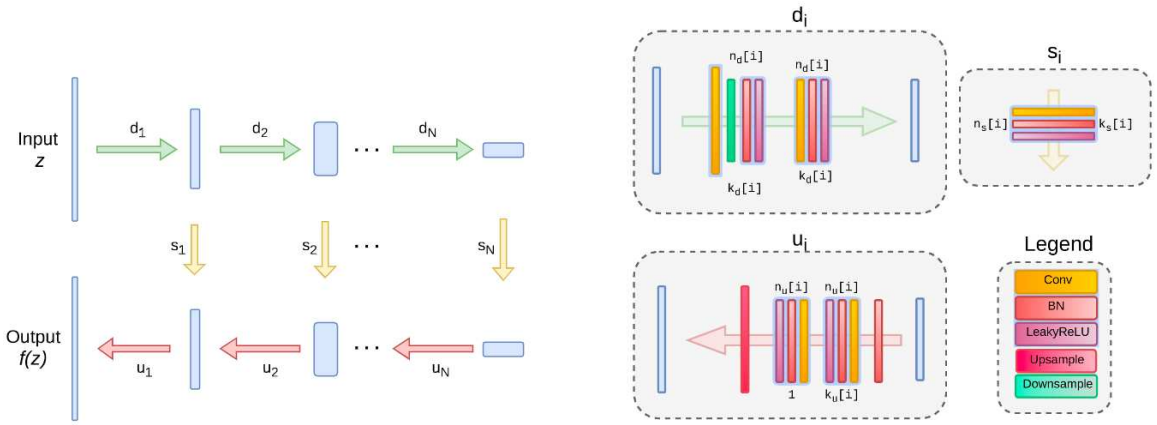


Figure 3.3: The "decoder-encoder" architecture with skip connections (yellow arrows). $n_u[i]$, $n_d[i]$, $n_s[i]$ correspond to the number of filters at depth i for the upsampling, downsampling and skip-connections respectively. The values $k_u[i]$, $k_d[i]$, $k_s[i]$ correspond to the respective kernel sizes. *Source:* [Ulyanov et al., 2017]

real observation. The output of this network is the probability that the input sample comes from the training set rather than the output of the Generator.

Both networks are trained simultaneously following a two-player minimax game so that the parameters/weights θ_G of the Generator network are regulated such that to maximally confuse the Discriminator, thus minimizing $\log(1 - D(G(\mathbf{z})))$, and the parameters θ_D of the Discriminator are regulated to make the best predictions of the class (true or false) belonging, thus maximizing $\log D(\mathbf{x}) + \log(1 - D(G(\mathbf{z})))$. Then the objective function of such a minimax game is following:

$$\min_{\theta_G} \max_{\theta_D} \big(\mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} (\log D(\mathbf{x})) + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} (\log(1 - D(G(\mathbf{z})))) \big) \tag{3.20}$$

The training procedure of this adversarial network is conducted with simultaneous update of both θ_D and θ_G by stochastic gradient descent of logistic loss functions:

$$\nabla_{\theta_D} \frac{1}{2} \frac{1}{n} \sum_{i=1}^{n} \big(\log(1 - D(\mathbf{x}^i)) + \log D(G(\mathbf{z}^i)) \big), \tag{3.21}$$

$$\nabla_{\theta_G} \frac{1}{n} \sum_{i=1}^{n} \big(\log(1 - D(G(\mathbf{z}^i))) \big), \tag{3.22}$$

where n is the size of the data batch during training process.

This GAN model could be further extended to the conditional case (namely CGAN) where as an input to both Generator and Discriminator there could be added also some auxiliary information which may help Generator to produce more similar to the data outputs or which may help Discriminator to distinguish better between input samples. Denoting this additional information as **cond**, we can rewrite the objective function from Eq. 3.20 as:

$$\min_{\theta_G} \max_{\theta_D} \big(\mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} (\log D(\mathbf{x}, \mathbf{cond})) + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} (\log(1 - D(G(\mathbf{z}, \mathbf{cond})))) \big) \tag{3.23}$$

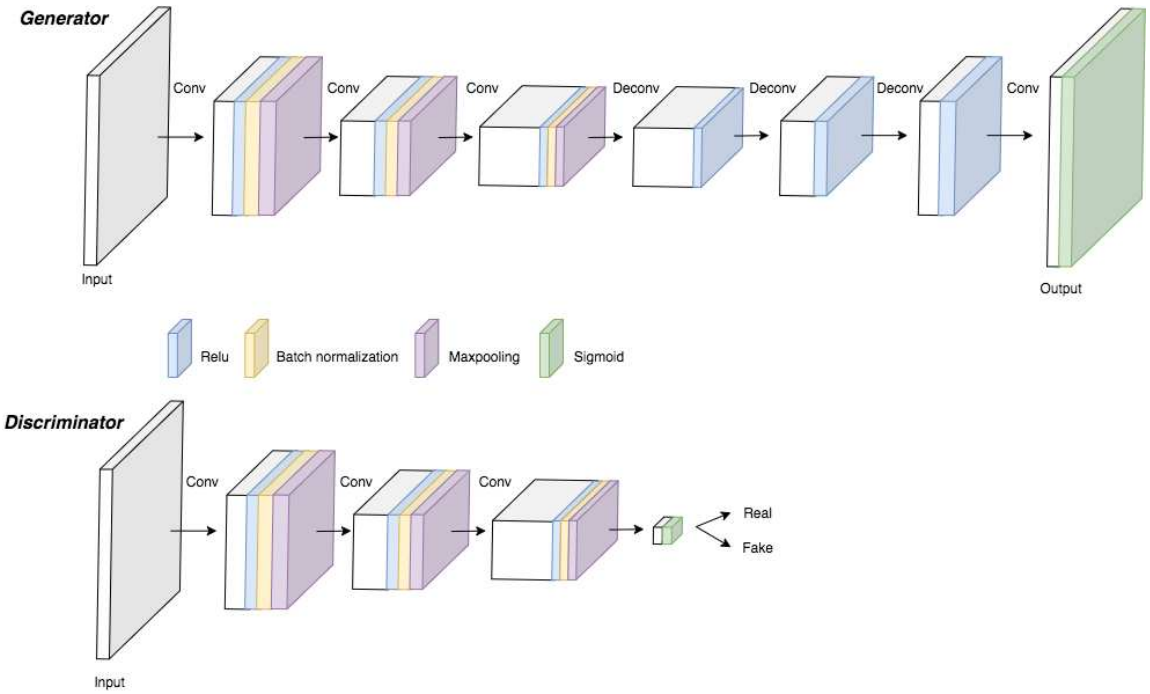


Figure 3.4: Example of architectures for Generator and Discriminator

Figure 3.4 depicts an example of the architectures of Generator (top), with the goal to produce examples which are indistinguishable from real data, and Discriminator (bottom), with the goal to estimate the probability that the data come from the real dataset rather than was produced by Generator.

3.2.4 Neural Architecture Search

The creation and selection of features in many tasks are done manually in general; this critical phase for some conventional machine learning algorithms might be time-consuming and costly. Neural Networls address this challenge by learning feature extractors in an end-to-end manner. These feature extractors, on the other hand, rely on architectures that are still manually constructed, and with the rapid development of the field, designing an appropriate NN model has become onerous in many cases.

This problem has recently been addressed by a new field of research called (`NAS`) [Elsken et al., 2019]. In a variety of applications, such as image segmentation and classification, Neural Networks with automatically found architectures have already outperformed "conventional" NN models with hand-crafted structures.

Different types of existing methods of search are described below.

Studies on the subject of `NAS` have gained significant interest in the last few years. In the literature, there are various of techniques based on Reinforcement Learning (RL) [Zoph and Le, 2016], evolutionary algorithm [Real et al., 2019] or Bayesian Optimization (BO) [Jin et al., 2019]. Recently, new gradient-based methods became increasingly popular. One of the first methods using this technique is called DARTS [Liu et al., 2018], in which a relaxation is used to simultaneously optimize the struc-

ture of a *cell*, and the weight of the operations relative to each *cell*. At the end, cells are manually stacked to form a neural network. Based on DARTS, more complex methods have emerged such as AutoDeepLab [Liu et al., 2019] in which a network is optimized at 3 levels : (i) the parameters of the operations, (ii) the cell structure and (iii) the macro-structure of the network that is stacked manually. Despite a complex representation leading to powerful architectures, this technique has certain drawbacks, such as the fact that the generated architecture is single-path, which means it does not fully exploit the representation's capabilities. Moreover, as the search phase is over a fixed architecture, it might not be the same between different runs, thus it is complicated to use transfer learning and the impact of training from scratch can be significant. To overcome these limitations, one technique is to use *Dynamic Routing* (`DR`) as proposed in [Li et al., 2020]. This approach is different from the traditional gradient based methods proposed for `NAS` in the sense that it does not look for a specific fixed architecture but generates a dynamic path in a mesh of cells on the fly *without searching*.

In the topic of signal strength map reconstruction there was no studies (for our knowledge) with application of `NAS` to the field measurements. There is no huge amount of data in the sense of amount of available base stations with sufficient number of measurements, so we will focus on the application of `NAS` to the single given image, which will be covered in the Chapter 4.1, and its adaptation for the generalization case given small amount of available base stations with corresponding measurements, which will be described in the Chapter 5.

3.3 Semi-Supervised Learning

The constitution of coherent and consistent labeled collections are often done manually. This necessitates tremendous effort, which is generally time consuming and, in some situations, unrealistic. The learning community has been looking at the concept of semi-supervised learning for discrimination and modeling tasks since the end of the 1990s, based on the observation that labeled data is expensive while unlabeled data is plentiful and contains information on the problem we are trying to solve.

Framework and definitions In this case, the labeled examples are generally assumed to be too few to obtain a good estimate of the association sought between the input space and the output space and the aim is to use unlabeled examples in order to obtain a better estimate. For this, we will assume available a set of labeled training examples $S = \{(x_i, y_i) \mid i = 1, \ldots, m\} \in (\mathcal{X} \times \mathcal{Y})^m$ supposed to be generated i.i.d. from an underlying distribution $\mathcal{D}$; and a set of unlabeled examples $X_u = \{x_i \mid i = m+1, \ldots, m+u\}$ that are drawn i.i.d. from the marginal distribution $\mathbb{P}(x)$. If X_u is empty, we fall back on the problem of supervised learning. If S is empty, we deal with an unsupervised learning problem. During learning, semi-supervised algorithms estimate labels for unlabeled examples. We note $\tilde{y}$ the pseudo-label of an unlabeled example $x \in X_u$ estimated by these algorithms. The interest of semi-supervised learning arises when $u = |X_u| \gg m = |S|$ and the goal

is that the knowledge one gains about the marginal distribution, $\mathbb{P}(x)$, through the unlabeled examples can provide information useful in inferring $\mathbb{P}(y \mid x)$. If this goal is not achieved, semi-supervised learning will be less efficient than supervised learning and it may even happen that the use of unlabeled data degrades the performance of the learned prediction function [Zhang and Oles, 2000, Cozman and Cohen, 2002]. It is then necessary to formulate working hypotheses for taking unlabeled data into account in the supervised learning of a prediction function.

Inductive vs Transductive Learning Before presenting these hypotheses, we note that semi-supervised learning can be formulated in two different possible settings, namely *transductive* and *inductive* learning. In the previous Section, we covered the inductive case where the learning objective is to minimize the generalization risk with respect to the distribution $\mathcal{D}$, by training a model over a finite number of training samples. This setting is also the most common in semi-supervised learning.

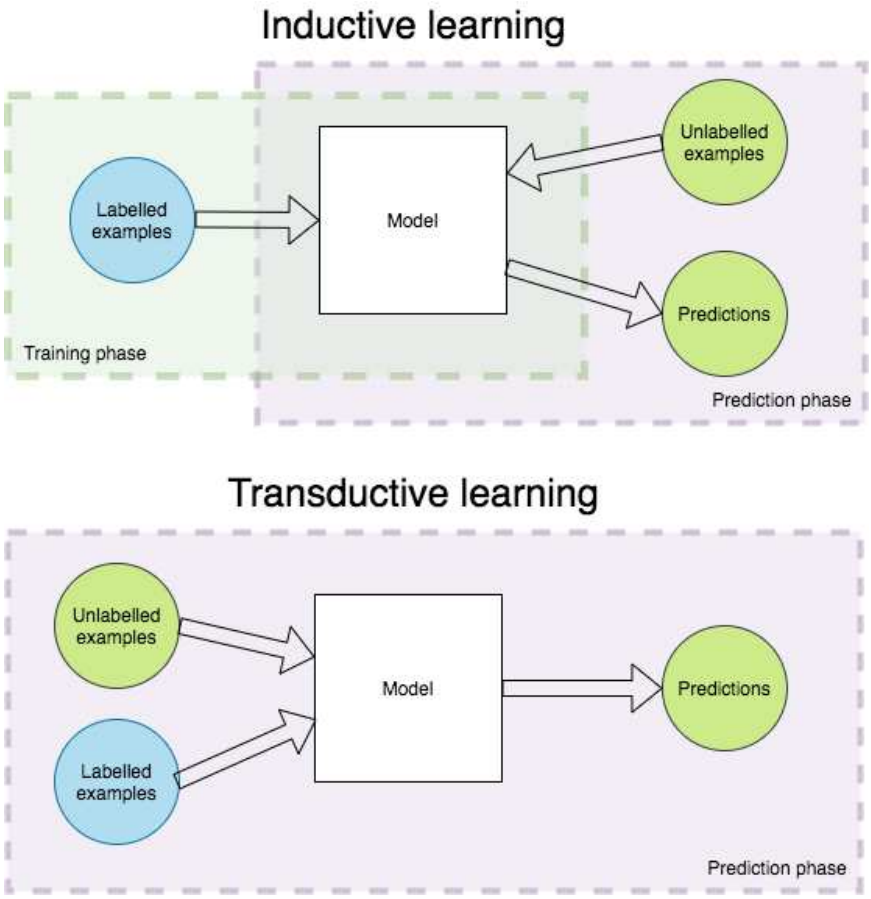


Figure 3.5: Inductive and transductive learning. In inductive learning there are two separate stages of training and prediction, while in transductive learning there is only one and which is prediction.

Despite this, accurate predictions for only the unlabeled cases in X_u are more crucial in some applications than finding a more general rule for all existing examples

drawn i.i.d. with respect to $\mathcal{D}$. It was shown in [Vapnik, 1982, Vapnik, 1998] that in this scenario, it would be preferable to ignore the more general problem and concentrate on an intermediary problem known as *transductive learning*. As a result, rather than looking for a general rule first (as in inductive learning), the goal of learning in this situation would be to predict the class labels of unlabeled cases by having the smallest average error (called the transductive error). These two settings are depicted in Figure 3.5.

3.3.1 Central assumptions

The three major hypotheses in SSL that are: *smoothness* assumption, *cluster* assumption and *manifold assumption*.

The basic assumption in semi-supervised learning, called the smoothness assumption states that:

Hypothesis 1. *If two examples x_1 and x_2 are close in a high density region, then their class labels y_1 and y_2 should be similar.*

This assumption implies that if two points belong to the same group, then their output label is likely to be the same. If, on the other hand, they were separated by a low density region, then their outputs would be different.

Now suppose that the examples of the same class form a partition. The unlabeled data could then help find the boundary of each partition more efficiently than if only the labeled examples were used. So one way to use the unlabeled data would be to find the partitions with a mixture pattern and then assign class labels to the partitions using the labeled data they contain. The underlying hypothesis of the latter, called the cluster assumption, can be formulated by:

Hypothesis 2. *If two examples x_1 and x_2 are in the same group, then they are likely to belong to the same class y.*

This hypothesis could be understood as follows: if there is a group formed by a dense set of examples, then it is unlikely that they can belong to different classes. This is not equivalent to saying that a class is formed by a single group of examples, but that it is unlikely to find two examples belonging to different classes in the same group. According to the previous continuity hypothesis, if we consider the partitions of examples as regions of high density, another formulation of the partition hypothesis is that the decision boundary passes through regions of low density. This assumption is the basis of generative and discriminative methods for semi-supervised learning.

For high-dimensional problems, these two hypotheses may not be accurate since the search for densities is often based on a notion of distance which loses its meaning in these cases. A third hypothesis, called the manifold assumption, on which some semi-supervised models are based, then stipulates that:

Hypothesis 3. *For high-dimensional problems, the examples are on locally Euclidean topological spaces (or geometric manifolds) of low dimension.*

In the following, we will present some classic models of the three families of semi-supervised methods resulting from the previous hypotheses.

3.3.2 Main approaches

There are three main families of SSL approaches that have been developed according to the above assumptions.

Generative Methods Semi-supervised learning with generative models involves estimating the conditional density $\mathbb{P}(x \mid y, \Theta)$ using a maximum liklihood technique to estimate the parameters Θ of the model. In this case, the hidden variables associated with the labeled examples are known in advance and correspond to the class of these examples. The basic hypothesis of these models is thus the cluster assumption (Hypothesis 2) since, in this case, each partition of unlabeled examples corresponds to a class [Seeger, 2001]. We can thus interpret semi-supervised learning with generative models (a) as a supervised classification where we have additional information on the probability density $\mathbb{P}(x)$ of the data, or (b) as a partition with additional information on the class labels of a subset of examples [Basu et al., 2002, Maximov et al., 2018]. If the hypothesis generating the data is known, generative models can become very powerful [Zhang and Oles, 2000].

Discriminant Methods The disadvantage of generative models is that, in the case where the distributional assumptions are no longer valid, their use will tend to deteriorate their performance compared to the case where only labeled examples are employed to learn a model [Cohen et al., 2004]. This finding has motivated many works to overcome this situation. The first works were based on the so-called directed decision technique (or self-training) proposed in the context of the adaptive processing of the signal and which consists in using the current predictions of the model for unlabeled examples in order to assign them pseudo-labels and use the pseudo-labeled examples in the training process. This process of pseudo-labeling and learning is repeated until no more unlabeled examples are pseudo-labeled. In the case where class pseudo-labels are assigned to unlabeled examples, by thresholding the outputs of the classifier corresponding to these examples, it can be shown that the self-learning algorithm works according to the clustering assumption [Amini et al., 2022].

Graph-based Methods Generative and discriminant methods proposed in semi-supervised learning exploit the geometry of the data through density estimation techniques or based on the predictions of a learned model. The last family of semi-supervised method uses an empirical graph $G = (V, E)$ built on the labeled and unlabeled examples to express their geometry. The nodes $V = [1, \ldots, m + u]$ of this graph represent the training examples and the edges E translate the similarities between the examples. These similarities are usually given by a positive symmetric matrix $\boldsymbol{W} = [W_{ij}]_{i,j} \in \mathbb{R}^{(m+u)\times(m+u)}$, where the weight W_{ij} is non-zero if and only if te examples indices i and j are connected, or equivalently, if $(i, j) \in E \times E$ is an edge of the graph G. The two examples of similarity matrices used in the literature are [Zhu and Ghahramani, 2002]:

- the binary matrix of the k-nearest neighbors:

$$\forall (i,j) \in \{1,\ldots,m+u\}^2; W_{ij} = 1 \text{ if and only if examples } x_i \text{ is among } k - \text{nearest neighbours example } x_j$$

- the Gaussian similarity matrix with parameter σ:

$$\forall (i,j) \in \{1,\ldots,m+u\}^2; W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

By convention, $W_{ii} = 0$. In the case of binary classification where $\mathcal{Y} = \{-1, +1\}$, a simple idea to take advantage of this construction is to propagate the labels of the examples through the graph. The nodes, $1,\ldots,m$ associated with the labeled examples are assigned class labels, $+1$ or -1, of these examples; and the nodes associated with the unlabeled examples are assigned the label 0. The proposed algorithms following this framework, called *label spreading* algorithms, are quite similar and they propagate the label of each node of the graph to its neighbors [Zhu and Ghahramani, 2002, Zhu et al., 2003, Zhou et al., 2004]. The objective of these algorithms is that the labels found $\tilde{Y} = (\tilde{Y}_m, \tilde{Y}_u)$ are consistent with the class labels of the labeled examples $\tilde{Y}_m = (y_1,\ldots,y_m)$, and also with the geometry of the data induced by the structure if the graph G and expressed by the matrix $\boldsymbol{W}$.

3.4 Conclusion

In this chapter, we presented a brief overview of supervised and semi-supervised learning frameworks by demonstrating their basic principles and discussing the main approaches developed under each. Among the many different approaches developed under these settings, we particularly presented Neural Network models that we would employ in the subsequent chapters that make up our contributions.

Part II

Contributions

Chapter 4

Self-Learning for Received Signal Strength Map Reconstruction with Neural Architecture Search

This chapter is based on submitted papers:

- [Malkova et al., 2021a]: "Self-Learning for Received Signal Strength Map Reconstruction with Neural Architecture Search" – Aleksandra Malkova, Loic Pauletto, Christophe Villien, Benoit Denis, Massih-Reza Amini. Artificial Neural Networks and Machine Learning – ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V;

- [Malkova et al., 2021b]: "Techniques d'Auto-Apprentissage et Recherche Automatique d'Architecture Neuronale pour la Reconstruction de Cartes de Puissance Radio" – Aleksandra Malkova, Loic Pauletto, Christophe Villien, Benoit Denis, Massih-Reza Amini. Conférence sur l'apprentissage Automatique (CAp), Jun 2021, Saint Etienne (en ligne), France.

In this chapter we will cover the first application of Machine Learning, more specifically Deep Learning, to the task of RSSI Map Reconstruction. This algorithm – Self-Learning for Received Signal Strength Map Reconstruction with Neural Architecture Search ($\texttt{SL}_{\texttt{NAS}}$) – works in the Transductive learning setting (see Section 3.3), where we try to assign the labels (meaning measurements) to the unseen set of points and then use these predictions to re-evaluate the output of the neural network. The advantage of this method is that we do not need the big dataset to train on. We use the single snapshot for each BS and do the reconstruction out of only given sparse measurements in the zone of interest around the base station location.

In this chapter, we consider `RSSI` map reconstruction in a constrained low-cost and low-complexity `IoT` context, where one can rely only on few ground-truth (i.e., `GPS`-tagged) single-snapshot field measurements and for which data-augmentation techniques based on side deterministic simulations cannot be applied, due to their prohibitive computational cost and/or to *a prior* unknown environment physical

characteristics. This problem of map interpolation is similar to the task of image restoration for which, NN based models with fixed architectures have been already proposed [Ulyanov et al., 2017]. In the case where there are few observed pixels in an image these approaches fail to capture its underlying structure that is often complex. To tackle this point we propose a first NN model based on Neural Architecture Search (NAS) for the design of the most appropriate model given a RSSI map with a small number of ground truth measurements. For this purpose, we develop two strategies based on genetic algorithms and dynamic routing for the search phase. We show that with the latter approach, it is possible to learn the model parameters while simultaneously searching the architecture. Ultimately, in order to enhance the model's predictions, the proposed approach uses also some extra data of the map with the predictions of the optimized NN in non-visited positions together with the initial set of ground-truth measurements for learning a final model. The proposed technique thus aims at finding practical trade-offs between agnostic learning interpolation techniques and data-augmented learning approaches based on deterministic prediction tools that generally require a very detailed physical characterization of the operating environment. Experimental results on five large-scale RSSI maps show that our approach outperforms non-learning based interpolation state-of-the-art techniques and NN models with a given fixed architecture.

4.1 Application of NAS to the Stated RSS Map Reconstruction Problem

In this section, we first introduce our notations and setting, and then present our main approach, denoted as $\texttt{SL}_{\texttt{NAS}}$ in the following.

4.1.1 Notations and Setting

For a given base station X, let $Y \in \mathbb{R}^{H \times W}$ be the whole matrix of ground-truth signal measurements , where $H \times W$ is the size of the (discretized) area of interest. We suppose to have access to only some ground truth measurements Y_m in Y, that is $Y_m = Y \odot M$, where $M \in \{0,1\}^{H \times W}$ is a binary mask indicating whether each pixel includes one available measurement or not, and $\odot$ is the Hadamard's product. Here we suppose that the number of non-null elements in Y_m is much lower than $H \times W$. We further decompose the measurements set Y_m into three parts Y_ℓ (for *training*), Y_v (for *validation*) and Y_t (for *test*), such that $Y_\ell \oplus Y_v \oplus Y_t = Y_m$, where $\oplus$ is the matrix addition operation. Let X_ℓ, X_v, X_t, X_m be the associated 2D node locations (or equivalently, the cell/pixel coordinates) with respect to base station X and X_u be the set of 2D locations for which no measurements are available.

Our approach is based on three main phases *i*) *architecture search phase* - the search of an optimal architecture of a Neural Network model; *ii*) *data-augmentation phase* - the assignment of pseudo-labels to randomly chosen unlabeled data using the predictions of the found NN model trained over Y_ℓ; and *iii*) *self-learning phase* - the training of a second NN model with the same architecture over the set of initial

Algorithm 1: $\texttt{SL}_{\texttt{NAS}}$

Input: A training set: (X_ℓ, Y_ℓ);
a validation set: (X_v, Y_v)
and a set of $2D$ locations without measurements: X_u;

Init: Using $(X_\ell, Y_\ell) \cup (X_v, Y_v)$, find interpolated measurements $\tilde{Y}_u$ over X_u using the `RBF` interpolation method;

Step 1: Search the optimal `NN` architecture using $(X_\ell, Y_\ell) \cup (X_u, \tilde{Y}_u)$;

Step 2: Find the parameters $\theta_1^\star$ of the `NN` model f_θ :

$$\theta_1^\star = \text{argmin}_\theta \, \mathcal{L}(X_\ell, Y_\ell, \theta) \quad \# \text{ (Eq. 4.1)};$$

Step 3: Choose $X_u^{(k)}$ randomly from X_u and find the new parameters $\theta_2^\star$ of the `NN` model f_θ :

$$\theta_2^\star = \underset{\theta}{\text{argmin}} \, \mathcal{L}(X_\ell \cup X_u^{(k)}, Y_\ell \cup f_{\theta_1^\star}(X_u^{(k)}), \theta);$$

Output: $f_{\theta_2^\star}$,$\tilde{Y}_u$

ground truth measurements and the pseudo-labeled examples. In the next sections, we present these phases in more detail. These phases are resumed in Algorithm 1.

4.1.2 Architecture Search phase

Here, we consider a first reference `RSSI` map as an input image, where unknown measurements in X_u are obtained with a `RBF` using points in the train and validation sets; $(X_\ell, Y_\ell) \cup (X_v, Y_v)$. The latter was found the most effective among other state-of-the-art interpolation techniques [Choi et al., 2018]. Let $\tilde{Y}_u$ be the set of interpolated measurements given by `RBF` over X_u. For the search phase of the `NAS` we have employed two strategies described below.

4.1.2.1 Genetic Algorithm

(`GA`) From the set $(X_\ell, Y_\ell) \cup (X_u, \tilde{Y}_u)$, we use an evolutionary algorithm similar to [Real et al., 2019] for searching the most efficient architecture represented as a Direct Acyclic Graph (`DAG`). Here, the validation set (X_v, Y_v) is put aside for hyperparameter tuning. The edges of this DAG represent data flow with only one input for each node, which is a single operation chosen among a set of candidate operations. We consider usual operations in the image processing field, that are a mixture of convolutional and pooling layers. We also consider three variants of 2D convolutional layers as in [Ulyanov et al., 2017] with kernels of size 3, 5 and 7, and two types of pooling layers that compute either the average or the maximum on the filter of size 4. Candidate architectures are then built from randomly sampled operations and the corresponding

NN models are trained. The 30 resulting architectures are then ranked according to a pixel-wise Mean Absolute Error (MAE) criterion between the interpolated result of the network and the set of interpolated measurements given by RBF $\tilde{Y}_u$. The most performing one is finally selected for mutation and placed in the trained population. The oldest architecture is removed in order to keep the size of the population equal to 20 as in [Real et al., 2019]. Table 4.1 illustrates such an optimized architecture with 18 nodes, which was found for the RSSI Map of the city of Grenoble used in our experiments (Section 2.3). The interconnection of these nodes is depicted in Figure 4.1.

Table 4.1: Description of the Neural network architecture structure by layers found by the Architecture Search phase for the RSSI Map of the city of Grenoble used in our experiments.

Layer	Operation	Input layer	Size
1	(Conv2D+BatchNorm+LReLu)×2+MaxPool		(368,368,32)
1a	Conv2D	1	(46,46,8)
1b	SpaceToDepth+Conv2D	1	(184,184,64)
2	(Conv2D+BatchNorm+LReLu)×2+MaxPool	1	(92,92,16)
2a	DepthToSpace+Conv2D	2	(184,184,8)
3	(Conv2D+BatchNorm+LReLu)×2+MaxPool	2	(92,92,8)
2a	DepthToSpace+Conv2D	2	(184,184,8)
4	Concatenation	3+1b	(46,46,16)
5	(Conv2D+BatchNorm+LReLu)×2+Upsampling	4	(92,92,4)
6	Concatenation	5+3a	(92,92,8)
7	(Conv2D+BatchNorm+LReLu)×2+Upsampling	6	(184,184,8)
8	Concatenation	1a+2a+7	(184,184,80)
9	(Conv2D+BatchNorm+LReLu)×2+Upsampling	8	(368,368,32)
10	(Conv2D+BatchNorm+LReLu)×2	9	(368,368,32)
11	Conv2D	10	(368,368,1)

4.1.2.2 Dynamic Routing

(DR) For the training phase, we employ the same structure and routing process as those proposed in [Li et al., 2020] (Figure 4.2). The structure is composed of 4 down-sampling level, where the size of the features map is divided by 2 at each level, but the depth of the latter is multiplied by 2 using a 1×1 2D-convolution. In our experiments we use a networks of 9 layers, which correspond to 33 cells in total (in yellow on Fig.4.2). The structure also contains an *"upsampling aggregation"* module at the end (red part on Fig.4.2). The goal of this module is to combine the features maps from all levels and reconstruct a map of the size of the input. Different from [Li et al., 2020], here, each cells contains three *transforming* operations (i.e. 2D-convolution with a kernel size of 3, 5 or 7) to have a good point of comparison with the method described above. However, due to the structure of the network we decided not to use pooling operations, as this could have been potentially redundant. In addition, we have left

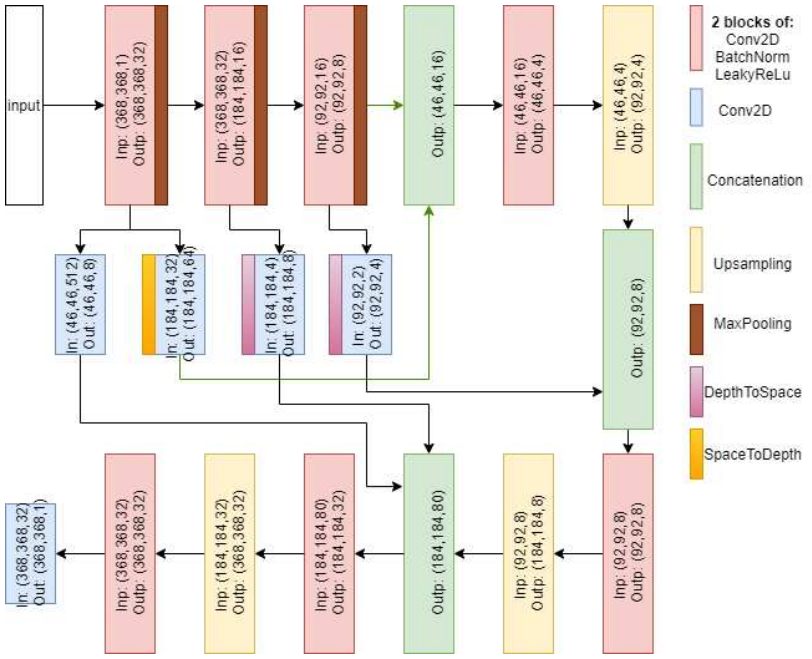


Figure 4.1: Interconnection between the nodes of the Neural network found by the Architecture Search phase.

the possibility of creating residual connections by adding operation identity in each cells. Moreover, we did not use the first two convolutions, originally used to reduce the size of the input, in order to keep as much information as possible. Instead, we used a 1×1 2D-convolution (in purple on Fig.4.2).

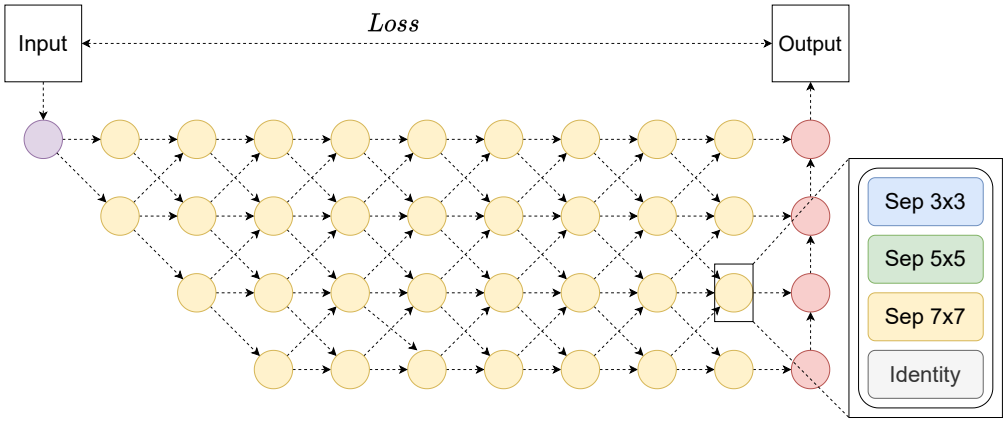


Figure 4.2: Diagram of the architecture used in our experiments. The purple, yellow and red dots represents respectively the "stem" convolution, the cells and the "upsampling aggregation" module. The arrows represent the data flow.

4.1.3 Data-augmentation and Self-Learning phases

After the search phase, the found NN model with parameters θ, f_θ is trained on (X_ℓ, Y_ℓ) by minimizing the following loss :

$$\mathcal{L}(X_\ell, Y_\ell, \theta) = \ell(f_\theta(X_\ell), Y_\ell) + \lambda\|\theta\|_2^2 + \mu\Omega(f_\theta(X_\ell)) \tag{4.1}$$

where $\ell(.)$ is the Mean Absolute Error (MAE), and $\Omega(f_\theta(X_\ell))$ is the total variation function defined as:

$$\Omega(Z) = \sum_{i,j} \mid z_{i+1,j} - z_{i,j} \mid + \mid z_{i,j+1} - z_{i,j} \mid, \tag{4.2}$$

with $z_{i,j}$ the measurement value of a point of coordinates i, j in some signal distribution map Z. This function estimates the local amplitude variations of points in Z that is minimized in order to ensure that neighbour points will have fairly close predicted measurements (i.e., preserving signal continuity/smoothness). Here, λ and μ are hyperparameters for respectively the regularization and the total variation terms and they are found by cross-validation.

With Dynamic routing used in the search phase, we optimize the network structure and the learning of parameters minimizing (Eq. 4.1) at the same time. Referring to Algorithm 1, the step 1 and 2 are combined in this case.

Let $\theta_1^\star$ be the parameters of the optimized NN model found by minimizing the loss (Eq. 4.1) on ground truth measurements (X_ℓ, Y_ℓ). This model is then applied to randomly chosen points, $X_u^{(k)}$, in X_u and pseudo-RSSI measurements $\tilde{Y}_u^{(k)}$ are obtained from the predictions of the optimized NN model $f_{\theta_1^\star}$: $\tilde{Y}_u^{(k)} = f_{\theta_1^\star}(X_u^{(k)})$. The number of ranfomly added points (k) will be also studied and depends on the initial amount of given sparse measurements.

With the same NN architecture, a second model $f_{\theta_2^\star}$ is obtained by minimizing the loss (Eq. 4.1) over the augmented training set $(X_\ell, Y_\ell) \cup (X_u^{(k)}, \tilde{Y}_u^{(k)})$.

4.2 Evaluation Settings

In all experiments, we considered maps of size 368×368 cells and tested our algorithm on field data from two distinct urban environments, namely the cities of Grenoble (France) and Antwerp (The Netherlands). We aggregated and averaged the given measurements in cells/pixels of size 10 meters x 10 meters. The Antwerp dataset is described in detail in Section 2.3.2 on which we considered three base stations, BS_1', BS_2' and BS_3', with respectively 5969, 6450 and 7118 ground-truth measurements. For the Grenoble dataset, we collected GPS-tagged LoRa RSSI measurements with respect to 2 base stations located in different sites BS_1 and BS_2 with respectively 16577 and 7078 ground truth measurements; the dataset is desribed in Section 2.3.3.

To perform in-cell data aggregation, we measured the distances based on local East, North, Up (ENU) coordinates. Then in each cell, we also computed the mean received power over all in-cell measurements (once converted into RSSI values), before

feeding our algorithm and the averaged RSSI values have been normalized between 0 and 1.

For each base station, 8% of the pixels with ground-truth measurements were chosen for training (X_ℓ, Y_ℓ), 2% for validation (X_v, Y_v) and the remaining 90% for testing (X_t, Y_t).The unlabeled data used in **Step 3** of Algorithm 1 were selected at random from the remaining 4% of each map's cells with no ground truth measurements. Results are evaluated over the test set using the MAE, dB, estimated after re-scaling the normalized values to the natural received signal strength ones. The reported errors are averaged over 20 random sets (training/validation/test) of the initial ground-truth data and unlabeled data were randomly chosen for each experiment.

Considered baselines and their short description. We compare several state-of-the-art interpolation techniques with the proposed SL_{NAS} approach. First is Total Variation (TV) in-painting [Rudin et al., 1992] (Eg. 4.2), where in our case the result is obtained by solving the optimization problem

$$\begin{aligned} \underset{Z}{\text{argmin}} \quad & \Omega(Z) \\ \text{s.t.} \quad & Y_l = Z \odot M_l \end{aligned} \tag{4.3}$$

where Z is some signal strength distribution map and the goal is to minimize the total variation function while conserving the measurements in a given points on the map (Y_l).

Another baseline is Radial basis functions (RBF) [Bishop, 2006, Powell, 1987]. Initially radial basis functions were introduced for the interpolation of exact function values. In other words, we want to keep the same values in the points where they were given. In our case, we can formulate the condition in the same way as in Eq. 4.3: $Y_l = Z \odot M_l$ where Z is the output signal distribution map. Suppose that we have $|X_l|$ points with known measurements values. Then the final interpolation function for each point z will be computed as

$$h(z) = \sum_{i=1}^{|X_l|} w_i \psi(d(z, X_l^i)) \tag{4.4}$$

where X_l^i is the data point with known measurement from the set X_l, z is some point without known measurement, $\psi(\cdot)$ is some radial basis function, $d(\cdot, \cdot)$ is some distance function (usually the Euclidean one is considered), $\{w_i\}$ is the set of weights (or coefficients) which is computed by least squares. The number of constraints (namely the number of given measurement point to be passed by the function) is the same as the amount of weight, so the final function will pass through these points exactly. There are different possible kernel functions that are widely used in various types of applications. In our case we considered linear kernel (or radial basis) function that was found the most performant, but there exist different ones, like Gaussian, cubuc, multiquadric, etc.

Next baseline is kriging (KRIG) [Oliver and Webster, 1990], or also which is called Gaussian process regression. First the log-normal path loss model trend from Eq. 2.2

	Grenoble		Antwerp		
	BS_1	BS_2	BS'_1	BS'_2	BS'_3
RBF [Bishop, 2006]	$5.03^{\downarrow}$	$3.16^{\downarrow}$	$3.58^{\downarrow}$	3.35	3.90
KRIG [Oliver and Webster, 1990]	$5.68^{\downarrow}$	$4.21^{\downarrow}$	$3.69^{\downarrow}$	$4.39^{\downarrow}$	$4.91^{\downarrow}$
NS [Bertalmio et al., 2001]	$5.11^{\downarrow}$	$3.14^{\downarrow}$	$4.28^{\downarrow}$	3.45	3.87
TV [Rudin et al., 1992]	$5.13^{\downarrow}$	2.89	3.76	3.51	3.83
DIP [Ulyanov et al., 2017]	$5.14^{\downarrow}$	$3.22^{\downarrow}$	3.53	3.41	3.92
SL_{NAS}-DR	4.82	2.82	3.48	3.42	3.81
SL_{NAS}-GA($f_{\theta_1^\star}$)	4.79	2.81	3.39	**3.27**	3.75
SL_{NAS}-GA($f_{\theta_2^\star}$)	**4.76**	**2.79**	**3.33**	**3.27**	**3.74**

Table 4.2: Average values of the MAE, dB of different approaches on all base stations.

(the parameters are computed by ordinary least squares according to the given data) is substracted from the collected data and then the ordinary kriging procedure is thus applied to predict the shadowing on the image.

The last one is Navier-Stocks (NS) [Bertalmio et al., 2001], which is based on the equations from fluid dynamics and applied to the image inpainting problem by solving locally these equations in the zones where there is no measurements. The authors proposed analogies for the fluid dynamics quantities to quantities in the inpainting method to preserve the smoothness and image intensity.

For the proposed SL_{NAS} approach, we employ both search phase methods based on Genetic Algorithm (GA) and Dynamic Routing (DR) and respectively referred to as SL_{NAS}-GA and SL_{NAS}-DR. For SL_{NAS}-GA we also evaluate the impact of the self-training step (**Step 3**) (called SL_{NAS}-GA($f_{\theta_2^\star}$)) by comparing it with the NN model found at **Step 1** (called SL_{NAS}-GA($f_{\theta_1^\star}$)). The evolutionary algorithm in the architecture search phase (Section 4.1.2) was implemented using the NAS-DIP [Ho et al., 2020] package[1]. The latter was developed over the Deep Image Prior (DIP) method (see Chapter 3.2.3) which is a fixed convolutional NN model proposed for image reconstruction. By considering RSSI maps as corrupted images with partially observed pixels (ground-truth measurements), we also compare with this technique by training a NN model having the same architecture than the one presented in [Ulyanov et al., 2017] and referred to as DIP in the following. All experiments were run on Tesla V100 GPU.

4.3 Experimental Results

Table 4.2 summarizes results obtained on the five considered RSSI maps. We use boldface to indicate the lowest errors. The symbol $^{\downarrow}$ indicates that the error is significantly higher than the best result with respect to Wilcoxon rank sum test used at a p-value threshold of 0.01 [Wilcoxon, 1945]. In all cases, SL_{NAS}-GA and SL_{NAS}-DR perform better than other state-of-the-art models even without the data-augmentation and self-training steps (SL_{NAS}-GA($f_{\theta_1^\star}$)). We notice that DIP which is also a NN based

[1] https://github.com/Pol22/NAS_DIP

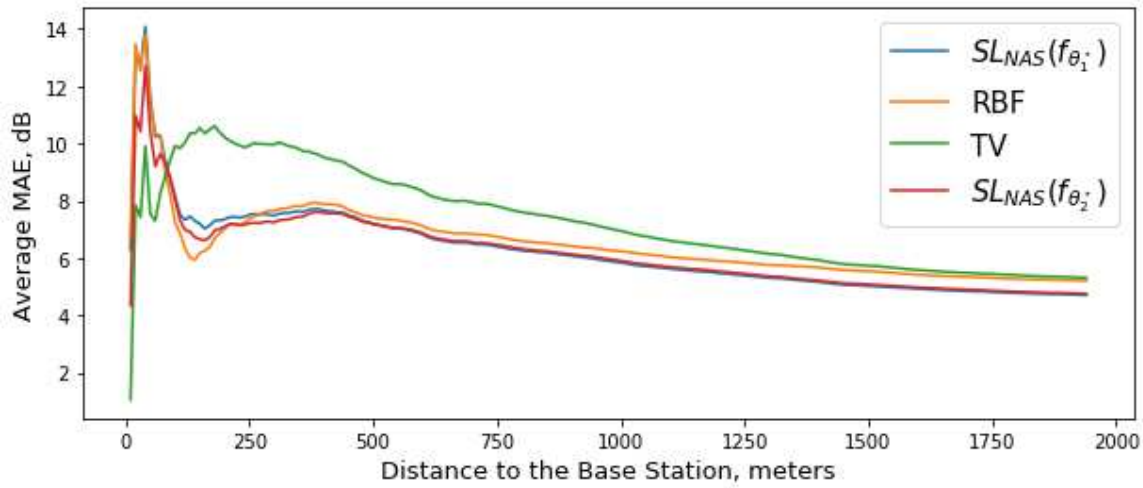


Figure 4.3: MAE, dB with respect to the distance to the base station, BS_1.

model but with a fixed architecture has similar results than RBF. These results show that the search of an optimized NN model is effective for RSSI map reconstruction in a constrained low-cost and low-complexity IoT context.

Figure 4.3 depicts the average MAE in dB with respect to the distance to the Base Station BS_1 for the city of Grenoble. For a distance above 250m, $\texttt{SL}_{\texttt{NAS}}\texttt{-GA}(f_{\theta_2^\star})$ provides uniformly better predictions in terms of MAE. These findings point to new investigations into how the model predicts the signal dynamics in regions where the signal is more irregular and where the dynamics are strong (for example near the base stations), especially in the cases where extra contextual knowledge about the physical environment may be included into the learning process (e.g., typically as a side information channel or the city map) that we tackle in the next chapter.

Figure 4.4 displays the MAE, dB boxplots of DIP, RBF and $\texttt{SL}_{\texttt{NAS}}\texttt{-GA}$ $(f_{\theta_2^\star})$ on BS_1 for different percentages of unlabeled data used in the self-learning phase (Section 4.1.3). We notice that by increasing the size of unlabeled examples, the variance of MAE for $\texttt{SL}_{\texttt{NAS}}\texttt{-GA}(f_{\theta_2^\star})$ increases mostly due to the increase of noisy predicted signal values by $f_{\theta_1^\star}$. This is mostly related to learning with imperfect supervisor that has been studied in semi-supervised learning [Amini et al., 2009, Krithara et al., 2008]. As future work, we plan to incorporate a probabilistic label-noise model in **step 3** of algorithm 1 and to learn simultaneously the parameters of the NN and the label-noise models.

Figure 4.5 there is a visual comparison for one of the base stations in Grenoble, France. The interpolations have either very local effects (cases 4.5a, 4.5c), follow the law of signal propagation (4.5b) or tries to smooth the regions while keeping the input points (4.5d). Our proposed algorithm has both of the properties: trying to follow the distribution (meaning keeping the input values) and having the smoothness for the signal distribution.

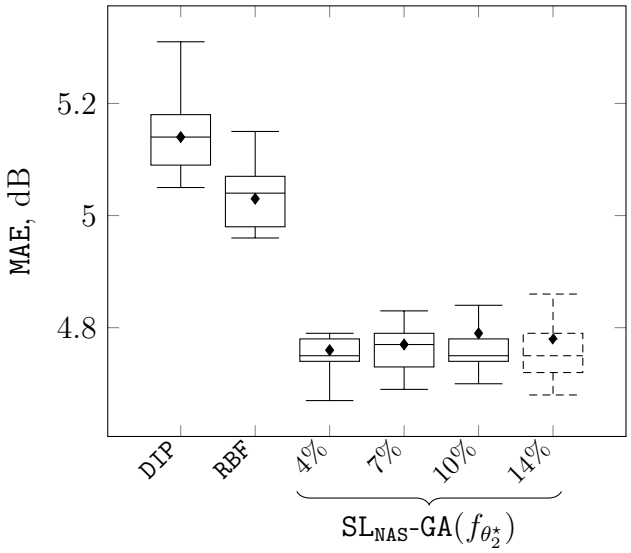
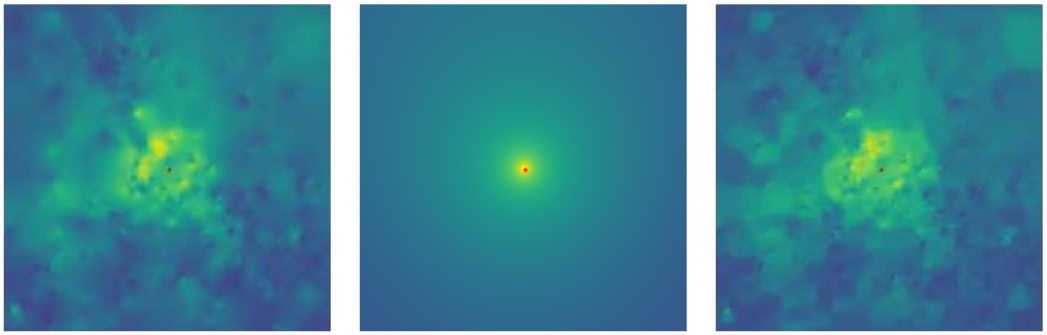
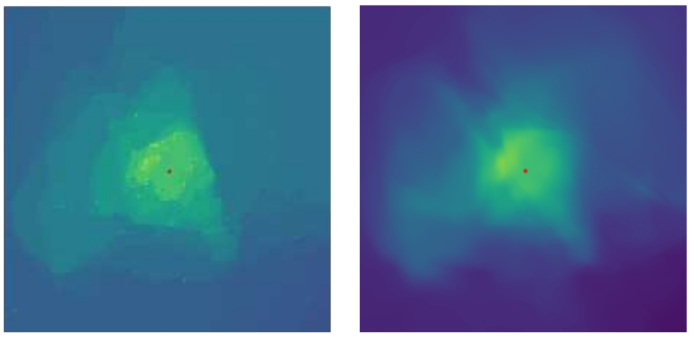


Figure 4.4: Boxplots showing the MAE, dB distributions of DIP, RBF and SL_{NAS}-GA ($f_{\theta_2^\star}$) on BS_1 for different percentage of unlabeled data $\{4, 7, 10, 14\}$ used in the self-learning phase.



(a) RBF interpolation (b) KRIG interpolation (c) NS interpolation

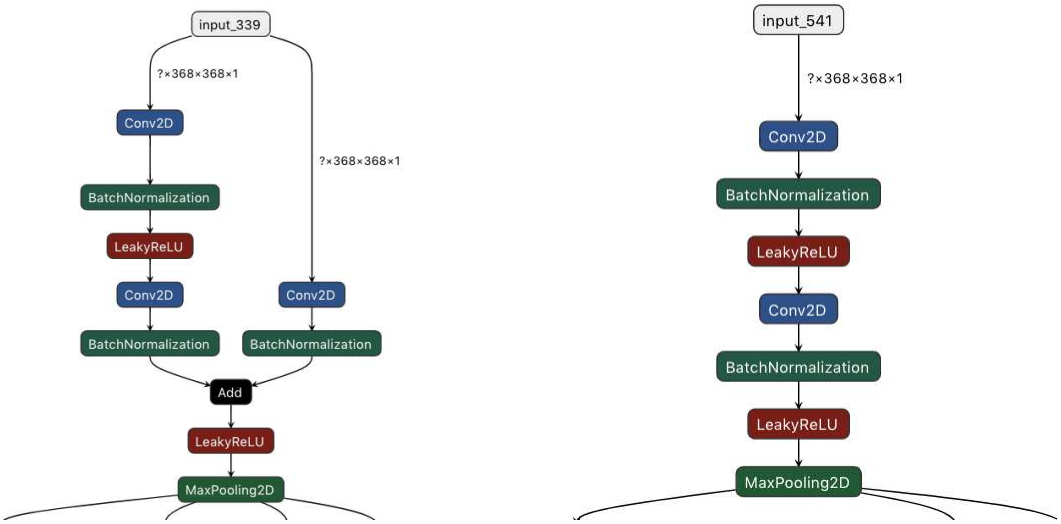


(d) TV interpolation (e) SL_{NAS}-GA($f_{\theta_1^\star}$) output

Figure 4.5: Visual comparison of the output for different techniques for BS_1, BS position is in the center (red point), close zone

Comparison of the obtained models. When we look at the structure of the obtained best models for each considered base station, it is possible to find similar

patterns in the layers order and construction. For example, best models for BS'_2, BS'_3 and BS_2 have similar order of the first two convolutional layers (without additional skip connections), see Figure 4.6b, but all of them have different kernel sizes (thus these models are different), while for the BS'_1 and BS_1 they have skip connections and additional convolutional layer in parallel, see Figure 4.6a. The same similarities in the layer sequence could be found in the last layers, while in the middle it is hard to find the same patterns. Thereby, as all the networks overall have different construction and different internal layer sizes, we could suppose that to extract local features and to find their internal relations for each base station with its own distribution without having access to any other type of information with this algorithm it is quite hard to obtain the generalized version of the architecture.



(a) Example of parallel convolutional layer with connection

(b) Example of sequentional layers constructions

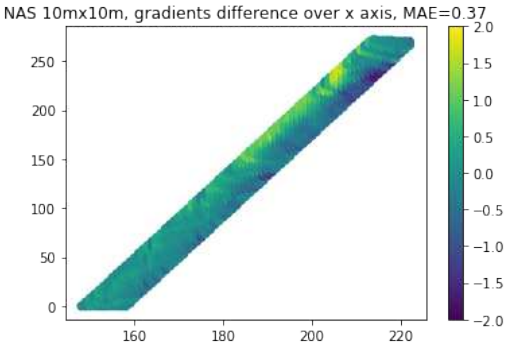
Figure 4.6: Difference in the first layers between the best received architectures

Comparison of the gradients behaviour for the Line-of-sight street for base station located in Grenoble. In the LOS conditions we can model the signal behaviour by doing the interpolation over the given points as there are almost no obstacles which can dramatically affect the signal. So one possible direction of comparison lies in the looking at the closeness of the gradients in the model and the output of some algorithm. Comparing the different interpolation techniques it has been shown that the $\mathtt{SL}_{\mathtt{NAS}}$ interpolation has the smoothest gradients in the output and the smallest range of it for both x and y axes (Figure 4.7) compared to the other methods when we consider the road in the LOS in one of the base station located in Grenoble. Also in this figure we can see the gradients difference for the baselines like RBF interpolation, Navier-Stokes interpolation, Total Variation interpolation, and for them gradients variations have more local effect than for the proposed algorithm, as all the baselines tends to pass directly through the given points without possible

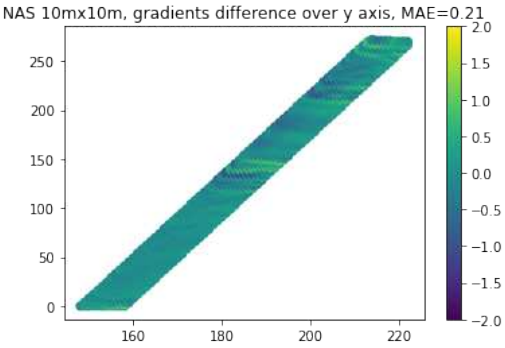
consideration of the presenting of the noise and so having the big gradients between the given measurements points and interpolated ones. Moreover, gradients over the $\texttt{SL}_{\texttt{NAS}}$ output are the closest by Mean absolute error (MAE) to the values, computed by the model in Eq. A.1 over given points in this LOS road.

4.4 Conclusion

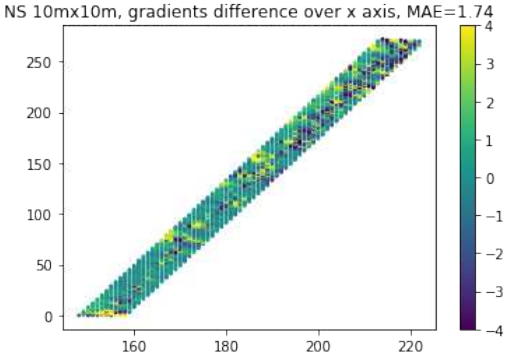
In this chapter, we presented a Neural Network model based on NAS and self-learning for RSS map reconstruction from sparse single-snapshot input measurements in the absence of data augmentation via side deterministic simulations. We presented two variants for the search phase of NAS based on Genetic algorithm and Dynamic routing. Experimental results on five large-scale maps of RSS measurements reveal that our approach outperforms non-learning based interpolation state-of-the-art techniques and NN with manually designed architecture.



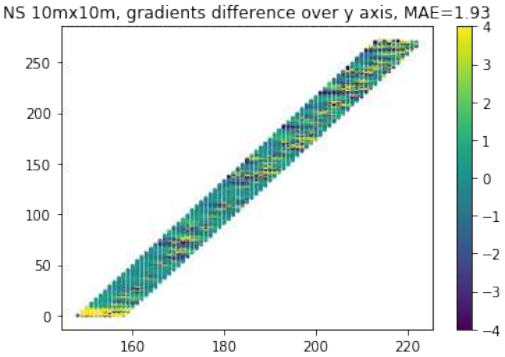
(a) NAS, gradients over x axis



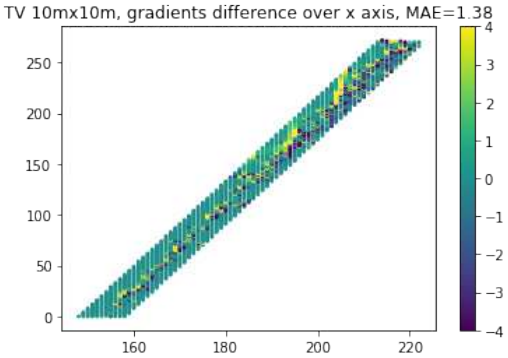
(b) NAS, gradients over y axis



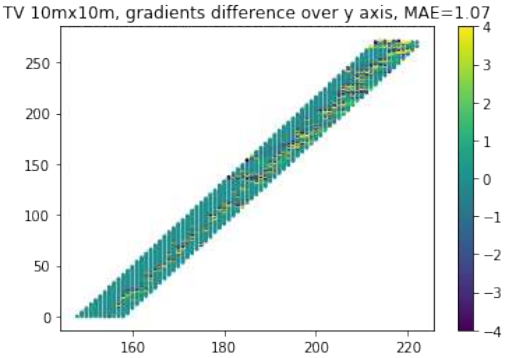
(c) NS, gradients over x axis



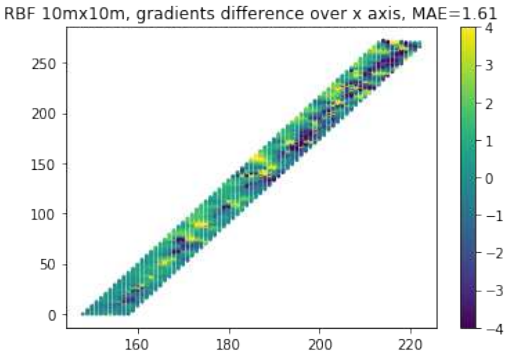
(d) NS, gradients over y axis



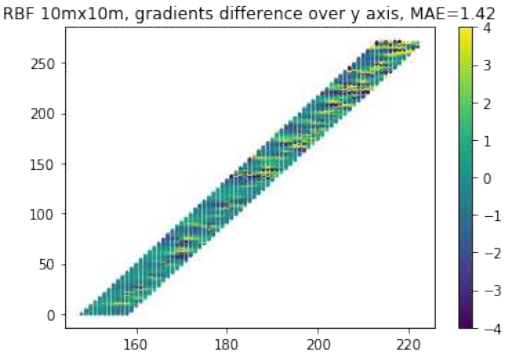
(e) TV, gradients over x axis



(f) TV, gradients over y axis



(g) RBF, gradients over x axis



(h) RBF, gradients over y axis

Figure 4.7: Difference between different methods gradients and signal distribution model gradients over LOS street

Chapter 5

Use of Heterogeneous Side Information

This chapter is based on submitted paper:
"Radio Map Reconstruction with Deep Neural Networks in a Weakly Labeled Learning Context with use of Heterogeneous Side Information" – Aleksandra Malkova, Massih-Reza Amini, Benoit Denis and Christophe Villien. 4th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2022), 19-21 October, Corfu, Greece.

We demonstrated in the last chapter that an effective transductive learning model can be found using interpolated RBF measurements on unlabeled locations corresponding to base stations that served to find the model. Until now, it was thought that models developed from data on a single base station could not be applied to additional, unknown base stations.

For that, we covered the case of map reconstruction for, at the same time, a small amount of input measurements (in our case less than 10% of available pixels after aggregation) and not big enough set of available base stations (at most 6 for one city). Although the achieved map reconstruction accuracy is greater than that of other state-of-the-art approaches, the suggested strategy is transductive, and its fundamental disadvantage is that independent of past map learning, a new network architecture must be constructed for a new arriving base station. Another caveat, is that there was no significant improvement for areas within a certain radius of the base station which for some applications, may be the most important. The learning time for finding an appropriate architecture for RSSI map reconstruction and learning the parameters of the obtained model is also a strong constraint of using the proposed approach in Chapter 4 for RSSI map reconstruction.

In this chapter, we overcome these limitations by proposing an inductive approach that is able to make use of extra information during the learning process for this task by validating the assumption that interpolated measurements on certain provided base stations are crucial to learning an efficient general model with efficient predictions on points corresponding to unseen base stations. In particular, we will show how these side information affect the final reconstruction for new base stations that were not

identified during the learning process.

5.1 Application to the Stated RSS Map Reconstruction Problem

Additional information could be represented in different manners, and they could be included into the algorithm in a variety of ways, such as independent channels, parallel channels inputs, directly in the learning goal, or in the ranking metric during model selection. We adapted the proposed algorithm presented in Chapter 4 for multi-channel input by combining additional context information with the data in the model's input; and we assessed the model's performance on unseen base stations that were not utilized in the learning process.

Here, we suppose to have a small set of n available base stations $(X^j)_{1 \leqslant j \leqslant n}$. For each given matrix of base station $X^j; j \in \{1, \ldots, n\}$, let $Y^j \in \mathbb{R}^{H \times W}$ be its corresponding 2D matrix of signal strength values measurements, where $H \times W$ is the size (in number of elements in a grid) of the zone of interest. In practice, we have access only to some ground truth measurements Y_m^j, meaning that $Y_m^j = Y^j \odot M^j$, with $M^j \in \{0,1\}^{H \times W}$ a binary mask of available measurements, and $\odot$ is the Hadamard's

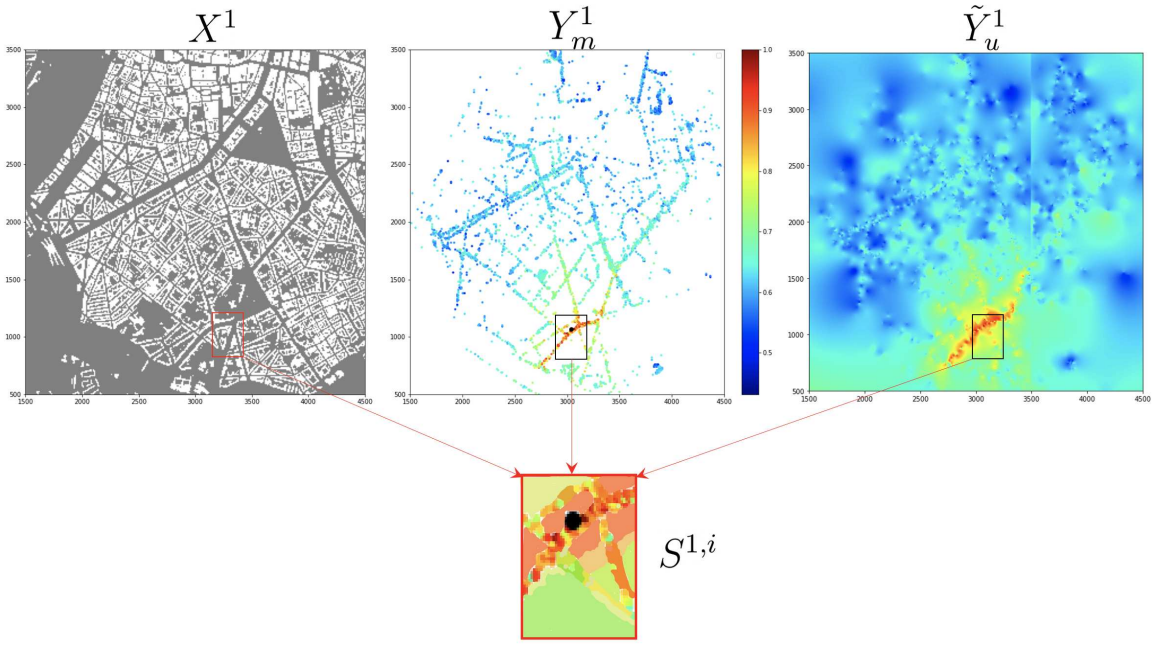


Figure 5.1: An example of constituting the training sets for one base station. X^1 corresponds to 2D node locations, buildings are shown in white. Y_m^1 is RSSI map (true measurements); the base station is shown by a black circle,and $\tilde{Y}^1$ corresponds to interpolated points found by RBF. Colors depict the strength of the signal from dark red (highest) to deep blue (lowest) RSSI values. $S^{1,i} = S_\ell^{j,i} \cup S_v^{j,i} \cup S_u^{j,i}$ is one sub-matrix of partially labeled training data found from $Y_m^1 \cup \tilde{Y}^1$.

product. Here we suppose sparsity meaning that the number of non-null elements in Y_m^j is much lower than the overall size $H \times W$. For each base station X^j we estimate unknown measurements $\tilde{Y}_u^j$ in Y^j with a RBF interpolation given (X_m^j, Y_m^j), so that we have a new subset $(X_u^j, \tilde{Y}_u^j)$, where $X_m^j = X^j \odot M^j$ is the associated 2D node locations of Y_m^j in X^j, and the values in $\tilde{Y}_u^j$ are initially given by RBF predictions on X_u^j corresponding to the associated 2D node locations (or equivalently, the cell/pixel coordinates) with respect to the base station X^j which do not have measurements. In our semi-supervised setting, the values for unknown measurements in $\tilde{Y}_u^j$ will evolve by using the predictions of the current NN model during the learning process.

We further decompose the measurements set Y_m^j into two parts: Y_ℓ^j (for *training*), Y_v^j (for *validation*), such that $Y_\ell^j \oplus Y_v^j = Y_m^j$, where $\oplus$ is the matrix addition operation. Let X_ℓ^j, X_v^j be the associated 2D node locations pf Y_ℓ^j and Y_v^j in X^j.

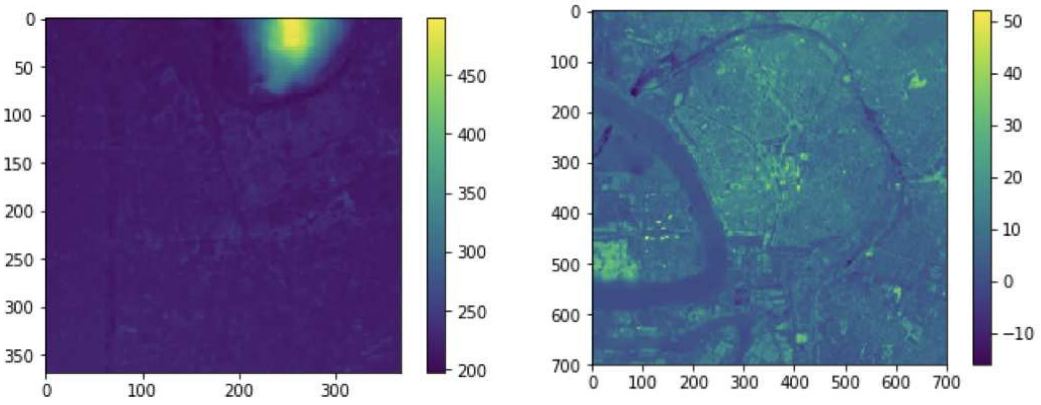
In our experiments the number of base stations n is small, so in order to increase the size of labeled and pseudo-labeled training samples, we cut the initial measurements maps $(Y_m^j \oplus \tilde{Y}_u^j)_{1 \leqslant j \leqslant n}$ into smaller matrices which resulted into the sets $(S^{j,i})_{\substack{1 \leqslant j \leqslant n \\ 1 \leqslant i \leqslant m_j}}$ where the sets $S^{j,i} \subseteq Y_m^j \oplus \tilde{Y}_u^j; \forall i \in \{1, \ldots, m_j\}$ are shifted with overlapping of the points. Each submatrix $S^{j,i}$ is hence divided into labeled, $S_\ell^{j,i} \cup S_v^{j,i}$, and pseudo-labeled (first interpolated points using RBF and then using the predictions of the current NN model) $S_{\aleph}^{j,i}$. To each submatrix $S^{j,i}$ corresponds a 2D location $X^{j,i} \subset X$. Figure 5.1 gives a pictorial representation of the notations.

5.2 NAS with Genetic Algorithm for RSSI map reconstruction using side information

`UNet` and GANs (Chapter 3, Section 3.2.3) are the two primary Neural Network models that can handle multiple channels and hence consider side-information as well as the RSSI map on their input. As additional context (or side) information, we have considered:

- information about buildings presence, which was taken from the open-source OpenStreetMap dataset [Boeing, 2017] – matrix of binary 0-1 values, denoted as "buildings map" further (Figure 5.1 left);
- amount of crossed buildings by signal from base station to each point of the map. By analogy to the data representation in the indoor localisation and map reconstruction, with the amount of crossed walls by signal – matrix of non-negative integer values, denoted as "buildings count map" further;
- information about distance from the base station. By the log-normal path loss model and corresponding RSSI (from the Section 2.1, Eq. 2.2) we can take the $-\log_{10}(distance)$ transformation to emphasize the zones closest from the base station – matrix of continuous values, denoted as "distance map" further;

- information about the relief represented by DSM (digital surface model): terrain elevation summed with artificial features of the environment (buildings, vegetation..), see Figure 5.2. This information was taken from the open-source dataset[1] provided by Japan Aerospace Exploration Agency with 30m accuracy – matrix of integer values, denoted as "elevation map" further.



(a) Elevations map, Grenoble (b) Elevations map, Antwerp

Figure 5.2: Elevation maps for two different cities

Our objective is to find the optimal architecture for GAN and `UNet` using these side-information and study the generalization ability of obtained models for RSSI map reconstruction.

NAS is performed with a Genetic algorithm is similar to one presented in Section 4.2as it gave better performance result in terms of obtained accuracy.

From the sets $(S^{j,i})_{\substack{1\leqslant j\leqslant n\\ 1\leqslant i\leqslant m_j}}$ we use an evolutionary algorithm similar to [Real et al., 2019] for searching the most efficient architecture represented as a Direct Acyclic Graph (`DAG`). Here, the validation sets $(S_v^{j,i})_{\substack{1\leqslant j\leqslant n\\ 1\leqslant i\leqslant m_j}}$ are put aside for hyperparameter tuning. The edges of this DAG represent data flow with only one input for each node, which is a single operation chosen among a set of candidate operations. We consider usual operations in the image processing field, that are a mixture of convolutional and pooling layers. We also consider three variants of 2D convolutional layers with kernels of size 3, 5 and 7, and two types of pooling layers that compute either the average or the maximum on the filter of size 4.

Candidate architectures are built from randomly selected operations and the corresponding `NN` models are trained over the set $(S_\ell^{j,i})_{\substack{1\leqslant j\leqslant n\\ 1\leqslant i\leqslant m_j}}$ and its (possible) combinations with side information. The resulted architectures are then ranked according to pixel-wise error between the interpolated result of the outputs over $(S_v^{j,i})_{\substack{1\leqslant j\leqslant n\\ 1\leqslant i\leqslant m_j}}$ and interpolated measurements given by RBF interpolation by filtering out the buildings. As error functions, we have considered the Mean Absolute Error (`MAE`) or its Normalized version (`NMAE`) where we additionally weight the pixel error according to the

[1] `https://doi.org/10.5069/G94M92HB`

Algorithm 2: $\texttt{SL}_{\texttt{NAS}}{}^{ind}$

Input: A labeled training set with given measurements: $(X_m^j, Y_m^j)_{1\leqslant j\leqslant n}$ and an unlabeled set $(X_u^j)_{1\leqslant j\leqslant n}$

Init: Using $(X_m^j, Y_m^j)_{1\leqslant j\leqslant n}$, find interpolated measurements $(\tilde{Y}_u^j)_{1\leqslant j\leqslant n}$ over $(X_u^j)_{1\leqslant j\leqslant n}$ using the RBF interpolation method;

Step 1: Cut the initial measurements maps $(Y_m^j \oplus \tilde{Y}_u^j)_{1\leqslant j\leqslant n}$ into smaller matrices: $(S^{j,i})_{\substack{1\leqslant j\leqslant n \\ 1\leqslant i\leqslant m_j}}$.

Step 2: Search the optimal NN architecture using $(S^{j,i})_{\substack{1\leqslant j\leqslant n \\ 1\leqslant i\leqslant m_j}}$;

Scenario 1: Find the parameters $\theta_1^\star$ of the NN model f_θ :

$$\theta_1^\star = \text{argmin}_\theta \, \mathcal{L}(f_\theta, S_\ell \cup S_{\not u}^{j,i}) \quad \# \text{ (Eq. 5.1)};$$

Scenario 2: Apply $f_{\theta_1^\star}$ on unlabeled data and obtain new pseudo-labeled measurements $S_{\not u}^{j,i}$ and find the new parameters $\theta_2^\star$ of the NN model f_θ :

$$\theta_2^\star = \underset{\theta}{\text{argmin}} \, \mathcal{L}(f_\theta, S_\ell \cup S_{\not u}^{j,i})$$

Output: $f_{\theta_1^\star}$ for scenario 1 or $f_{\theta_2^\star}$ for scenario 2.

distance matrix value. Best ranked model is then selected for mutation and placed in the trained population. The oldest and worst in the rank are then removed to keep the population size equal to 20 models.

Once the NN model with the optimized parameters are found by NAS, f_θ, we consider the following two scenarios for learning its corresponding parameters θ by minimizing

$$\mathcal{L}(f_\theta, S_\ell \cup S_{\not u}^{j,i}) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{m_j}\sum_{i=1}^{m_j}\left[\frac{1}{|S_\ell^{i,j}|}\sum_{(x,y)\in S_\ell^{i,j}}\ell(y, f_\theta(x)) + \frac{1}{|S_{\not u}^{j,i}|}\sum_{(x,\tilde{y})\in S_{\not u}^{j,i}}\ell(\tilde{y}, f_\theta(x))\right] \tag{5.1}$$

These two scenarios relate to obtaining model parameters on *labeled and pseudo-labeled* measurements using just RBF interpolated data (scenario 1) or predictions from a first model learnt on these data (scenario 2). The overall learning process is depicted in Algorithm 2.

5.3 Evaluation Setup

We have considered three case studies from Paris, Antwerp (The Netherlands) and Grenoble.

Datasets. The size of maps are 500 × 500 pixels for the generated dataset from Paris, 700 × 700 pixels for the dataset collected in Antwerp and 368 × 368 for

city	name	all points	train points	validation points	status
Grenoble	BS_1	6264	5591	673	train
	BS_2	2728	2448	280	train
	BS_3	7266	6516	750	train
	BS_4	6836	6096	740	test
Paris	BS''_1 - BS''_5	250000	7495	242505	train
	BS''_6	250000	7495	242505	test
Antwerp	BS'_1	6060	5440	620	train
	BS'_2	5606	5034	572	train
	BS'_3	7548	6785	763	train
	BS'_4	2539	2276	263	train
	BS'_5	2957	2667	290	train
	BS'_6	4940	4453	487	train
	BS'_7	3154	2829	325	train
	BS'_8	8277	7455	822	train
	BS'_9	4335	3888	447	test

Table 5.1: Summary over the settings for different cities: total amount of available measurements, points used as an input to the models, validation (test) points that were used also in the computation of the loss (during the evaluation)

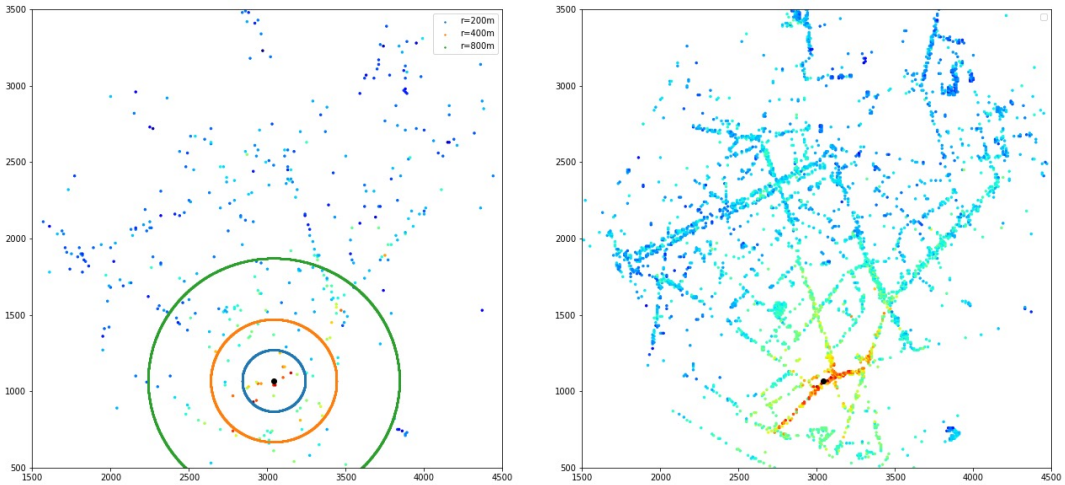
Grenoble (See Chapter 2.3). For that, we aggregated and averaged the power of collected measurements in cells/pixels of size 10 meters $\times$ 10 meters by the measured distance from base station location based on local ENU coordinates for Grenoble and Antwerp, while for Paris dataset the cell size is 2 meters $\times$ 2 meters. As we also consider the generalization task, the algorithm should learn from all the available base stations data simultaneously.

In our settings, we only have access to several base stations lacking several orders of magnitude in size compared to aforementioned datasets. To artificially overcome this drawback, we created submatrices of the original images by cutting them into smaller ones (we tested over 96 by 96 pixels size because of memory issues during learning of the neural network for the storing of the model weights). We also added the flipped and mirrored images and we also did a shift in 20 pixels meaning that in our dataset there were overlapping between the images. Moreover, if the amount of pixels with measurements in the initial cutted image was high enough (more than 3% of the presented pixels) then we masked out the randomly sampled rectangle of presented measurements similar to the cutout regularization ([Devries and Taylor, 2017]). By doing this we force the algorithm to do the reconstructions in the zones without measurements (not only locally) and be more robust to the amount of input data.

Matrices of the side information were used in the models as additional channels concatenated with measurements map. Before feeding the data into the algorithm,

all the values have been normalized between 0 and 1 in each channel separately before cutting them into smaller sizes to feed into the models.

Evaluation of the results over held out base stations To evaluate the result we left one base station out of the initial set of each city to compare further the models performances with baselines, namely test Antwerp and test Grenoble. To do this, all the points were divided into two parts, namely *train* and *test* points for 90% and 10% respectively. This will be used throughout all the following sections. Moreover, to highlight the importance of the zones close to base station (as it was mentioned in Chapter 2) we compare the performance of the algorithms over different considered circles around the base station location, namely 200 meters, 400 meters and 800 meters radiuses (see Figure 5.3).



(a) Test points and corresponding circular zones

(b) Train points for the same area for the same base station

Figure 5.3: Points division for the test base station, Antwerp; Base station location shown as a black point, coordinates are in meters

Information about base stations for each city, amount of all points, points, that were used in the validation/test process, and training set are given in Table 5.1.

As in the previous chapter, we considered state-of-the-art interpolation approaches which are: Total Variation (`TV`) in-painting by solving the optimization problem (Eg. 4.2), Radial basis functions (`RBF`) [Bishop, 2006] with linear kernel that were found the most efficient. The evolutionary algorithm in the model search phase was implemented using the `NAS-DIP` [Ho et al., 2020] package[2]. All experiments were run on NVIDIA GTX 1080 Ti 11GB GPU.

[2] `https://github.com/Pol22/NAS_DIP`

5.4 Experimental Results

In our experiments, we are primarily interested in addressing the following two questions: (a) does the use of side contextual information aid in the more accurate reconstruction of RSSI maps?; and; (b) to what extent is the search for an optimum NN design effective in the two scenarios considered (Section 5.2)?

Regarding the first point, we consider the following learning settings:

1. given only the measurements (no side information),
2. given both measurements and distance maps,
3. given measurements, distances and elevation maps,
4. given measurements, distances maps and map of amount of buildings on the way from base station to corresponding point in the map (or, in other words, buildings count).

From the standpoint of application, accurate interpolation in all regions where the signal varies the most is critical. We will compare the cumulative mistakes across held-out pixels for each of the zones that are close enough to the test base station for the LoRa signal (by considering the fixed radius of 1 km).

Initially, we attempted to optimize the Generative adversarial network (GAN) utilizing side information as a multi-channel input, but we discovered that the model is incredibly difficult to optimize (see appendix A.2 for more information). In the following, we will present our findings using the UNet model.

5.4.1 Generalization ability of UNet

We first study the learnability of the UNet model (Section 3.2.3) for RSSI map reconstruction without the use of unlabeled data. In order to see if there is an effect of using side information we have just considered distance maps as additional context information and considered the model with a hand-crafted classical architecture shown in Figure 5.4 used for in-painting.

The goal of this early experiment is to validate the usage of UNet for this task and investigate what effects the side information and labeled measurements have. For this we consider the simplest Paris data case where we keep only the points on the roads (as the points could be collected over the street by the vehicle drivers or pedestrians). The difference with Grenoble and Antwrep datasets is in the sampling procedure, as in reality it is very hard to obtain the collected data sampled uniformly in all the regions while in Paris dataset this is the case. All the RSSI measurements also exist in Paris dataset, this allows to see the importance of labeled information in the predictions by varying the percentage of labeled measurements in the training set.

Figure 5.5 depicts the evolution of MAE in measurements with respect to the distance to the test base station getting lower in comparison with the RBF interpolation,

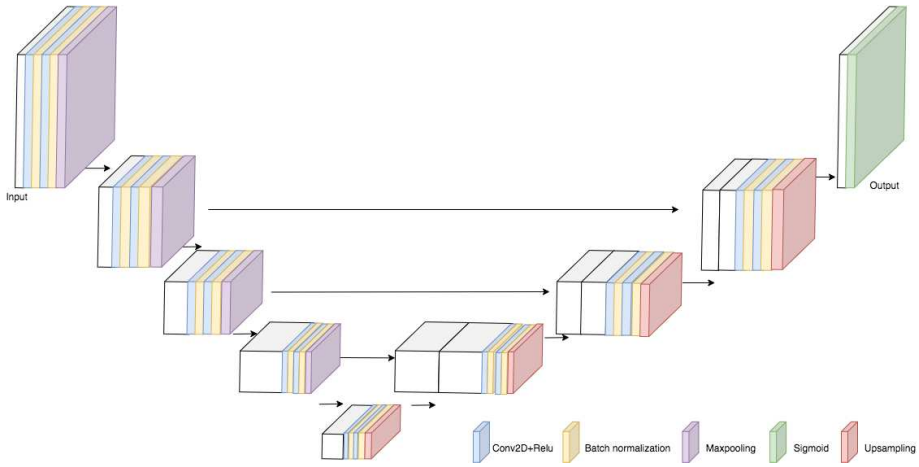


Figure 5.4: UNet model with hand-crafted architecture proposed for in-painting.

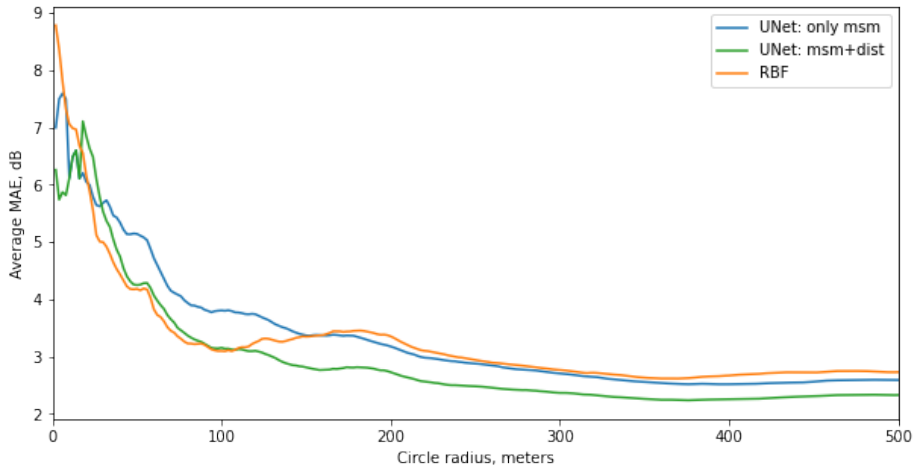


Figure 5.5: MAE for the distance from the base station for the RBF and fixed UNet outputs over Paris dataset with validation set compliment to the train one, measurements and distances or only measurements input

as well as the overall error becomes smaller (BS_6'' - table 5.1), of RBF, `UNet` using only measurements (UNet only msm) and UNet using measurements and distance maps (UNet msm+dist). From these results, it comes that RBF outperforms UNet using only measurements in a circle zone of less than 150 m radius around the base station. However, when distance maps are added to the model's second channel, the situation is reversed.

With the inclusion of side-information, we see that `UNet` performs around 1 point better in MAE than RBF. This situation is illustrated on the map reconstruction ability of both models around the test base station BS_6'' in Figure 5.6. As can be observed, the projected signal levels are more discernible on the roads, which are actually the zones of interest where the signal is sought, as predicted by the `UNet` model.

As a result, these findings show that the `UNet` model can effectively account for

side-information. We examined the RBF and `UNet` models for the influence of labeled measurements on the predictions by altering the percentage of labeled data utilized by RBF for discovering the interpolation and by the `UNet` model for learning the parameters. With regard to this proportion, Figure 5.7 displays the average MAE 200 meters (left) and 400 meters (right) away from the test base station. The test error of the RBF model on unseen test data remains constant as the quantity of labeled training data increases, but the test error of the UNet model decreases as this number increases.

5.4.2 The use of unlabeled by taking into account side information with NAS

We now expand our research to real-world data sets from Grenoble and Antwerp, taking into account more side-information and investigating the impact of neural architecture search on the creation of a better NN model. As in this case, the labeled training measurements are scarce we examine the usage of unlabeled data in addition to the labeled measurements as described in the previous section.

We begin by envisaging scenario 1 of Algorithm 2 (Section 5.2) and investigating the impact of side information on the performance of the optimized NN model discovered by NAS. The architecture of the NN model, $f_{\theta_1^\star}$, found for the city of Antwerp is shown in Figure 5.8.

Figure 5.9 shows the evolution of MAE inside a circular zone of varied radius for $f_{\theta_1^\star}$ trained just on measurements and measurements with distance maps (left) and $f_{\theta_1^\star}$ trained on measurements, distance maps, and building counts or elevation (right) on the test base stations of Antwerp BS_9'. The use of distance maps to supplement measurements improves predictions, which is consistent with our earlier findings. When the third side-information is included, such as height or building counts, we find that the elevation yields better signal estimations than the latter.

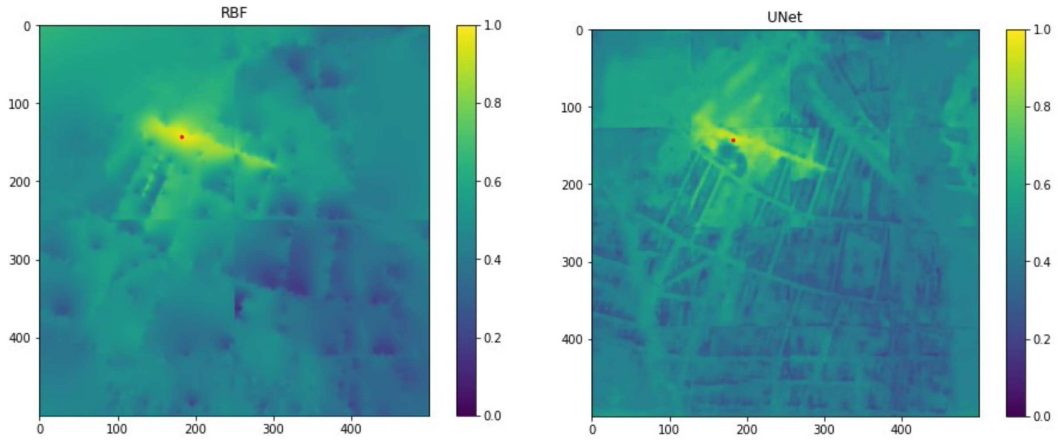
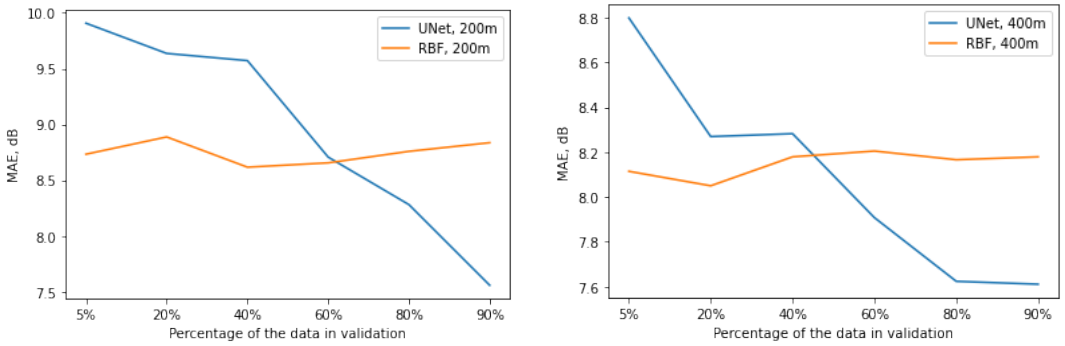


Figure 5.6: Map reconstruction of RBF (left) and `UNet` (right) over the test base station BS_6'' in Paris shown by a red dot.



(a) 200 meters from the base station (b) 400 meters from the base station

Figure 5.7: MAE with respect to different percentage of labeled data in the training set 200 meters (left) and 400 meters (right) away from the test base station.

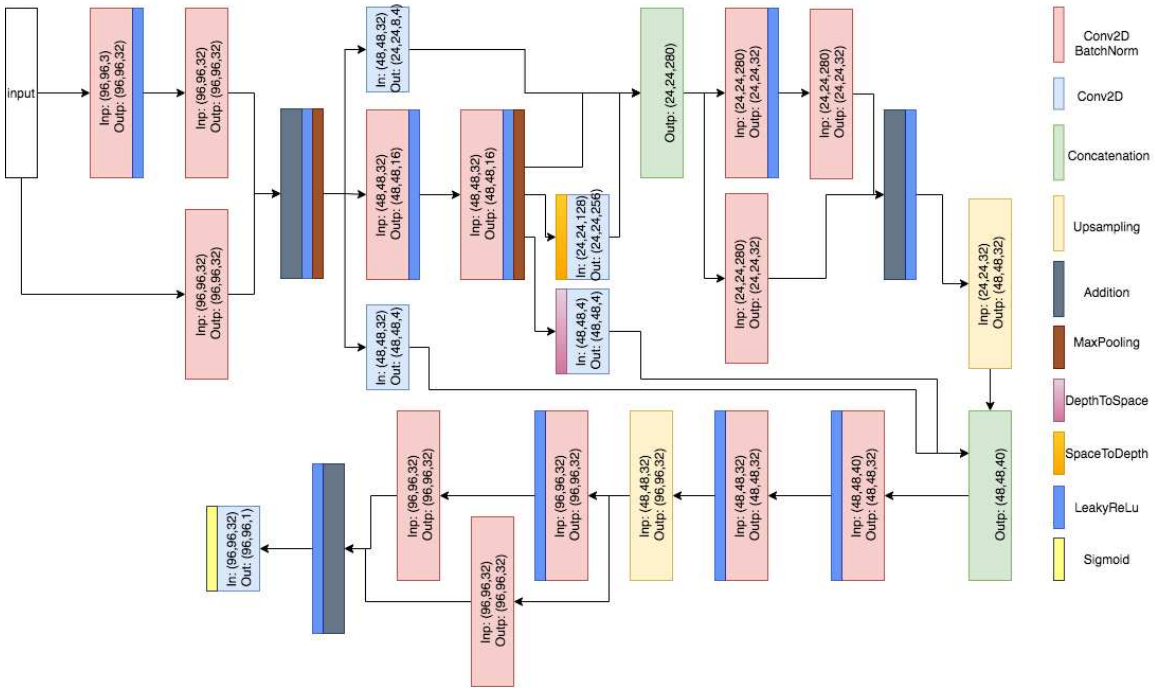


Figure 5.8: Example of the Neural network architecture found by the NAS for the RSSI Map of the city of Antwerp used in our experiments, $f_{\theta_1^\star}$.

This is understandable because signal transmission can be severely slowed by building heights.

As a best model obtained by Algorithm 2, scenario 1 we consider the case with three input channels: measurements, distances and elevations and present comparative results with other baselines in Table 5.2. The lowest errors are shown in boldface. The symbol $^\downarrow$ denotes that the error is significantly greater than the best result using the Wilcoxon rank sum test with a p-value threshold of 0.01. According to these findings, $f_{\theta_1^\star}$ outperforms other state-of-the-art models as well as the UNet model with a handcrafted architecture. These results suggest that the search of an opti-

Model	MAE, dB, 200m	MAE, dB, 400m
RBF	$8.34^{\downarrow}$	$7.04^{\downarrow}$
kNN	$7.98^{\downarrow}$	$7.08^{\downarrow}$
TV	$7.50^{\downarrow}$	$6.97^{\downarrow}$
UNet	$6.37^{\downarrow}$	$6.81^{\downarrow}$
$f_{\theta_1^\star}$	**5.88**	**6.37**

Table 5.2: Comparisons between baselines in terms of MAE with respect to the three distances to Antwerp' test base station BS'_9. Best results are shown in bold.

mal NN model with side-information has strong generalization ability for RSSI map reconstruction.

Figure 5.10 depicts the average MAE in dB of all models as well as the NN model $f_{\theta_2^\star}$ corresponding to scenario 2 of Algorithm 2, with respect to the distance to the test base Station BS'_9 for the city of Antwerp. For distances between 200 and 400 meters, $f_{\theta_2^\star}$ consistently outperforms in terms of MAE. As in chapter 4, these findings imply that self-training constitutes a promising future direction for RSSI map reconstruction.

Figure 5.11 presents the average MAE in dB of all models with respect to the distance to the test base Station BS_4 for the city of Grenoble. These results are consistent with those obtained over the city of Antwerp.

The general conclusion that we can draw is that knowing about local patterns (even if from different locations/distributions/base stations) allows us to use this information in signal strength map reconstruction for application to unseen measurements from different base stations, demonstrating the ability to generalize output in the same area.

In order to get a finer granularity look at the estimations of the suggested technique, $f_{\theta_2^\star}$, Figure 5.12 depicts the errors heatmaps on circular zones of radius 200m and 400m surrounding the test base stations for Antwerp and Grenoble. Each point reflects the difference between the real and predicted signal values. For both cities,

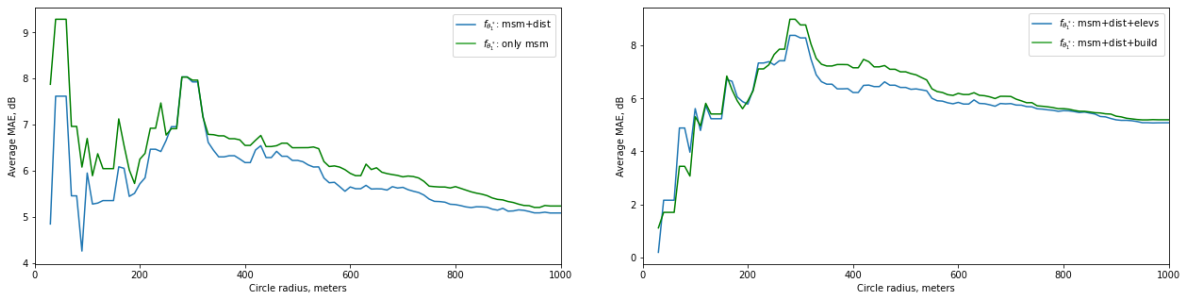


Figure 5.9: Cumulative MAE distribution of $f_{\theta_1^\star}$ (scenario 1 of Algorithm 2) according to the distance to the test base station for the city of Antwerp BS'_9; with measurements and measurements with distance maps (left) and measurements, distance maps, and building counts or elevation (right).

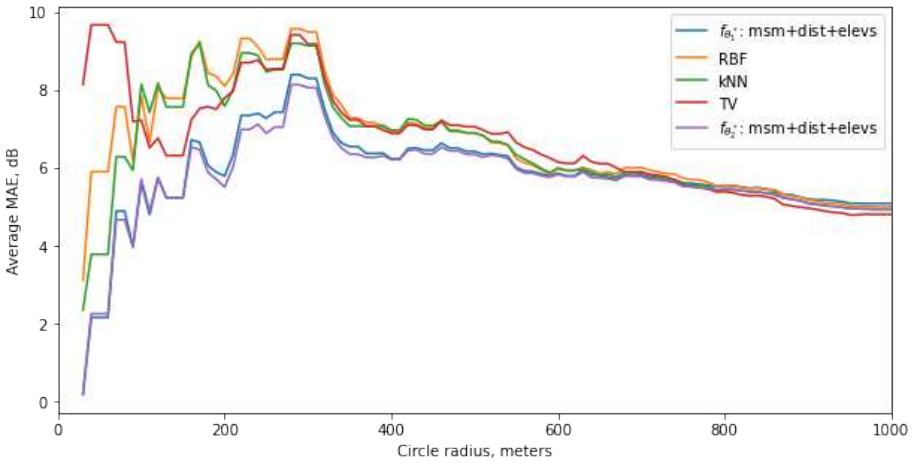
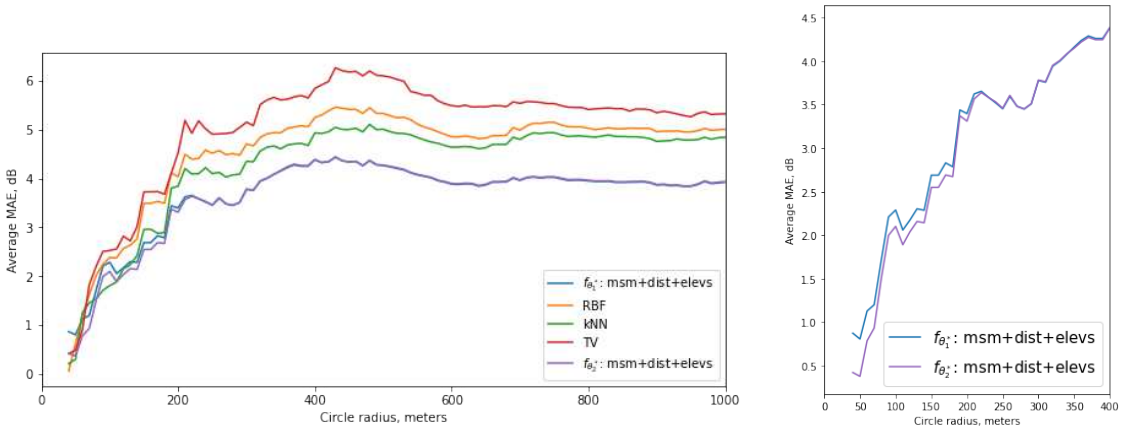


Figure 5.10: Average MAE in dB of all models with respect to the distance to the test base Station BS'_9 for the city of Antwerp.



(a) Comparison between different approaches (b) Zoom distance <400m

Figure 5.11: Average MAE in dB of all models with respect to the distance to the test base Station BS_4 for the city of Grenoble.

we notice that there is

- an overestimation of the signal (higher predicted values than the true ones) within the zone of radius less than 200 meters where the values of the true signal are high. In absolute value, the average MAE in dB are respectively 3.6 for Grenoble and 6.3 for Antwerp.
- an underestimation of the signal (lower predicted values than the true ones) within the zones of radius between 200 and 400 meters where the values of the true signal are low. In absolute value, the average MAE in dB are respectively 4.9 for Grenoble and 6.2 for Antwerp.

To better understand the aforementioned results, we provide the empirical cumulative distribution function (eCDF) of different techniques in a 200-meter zone around

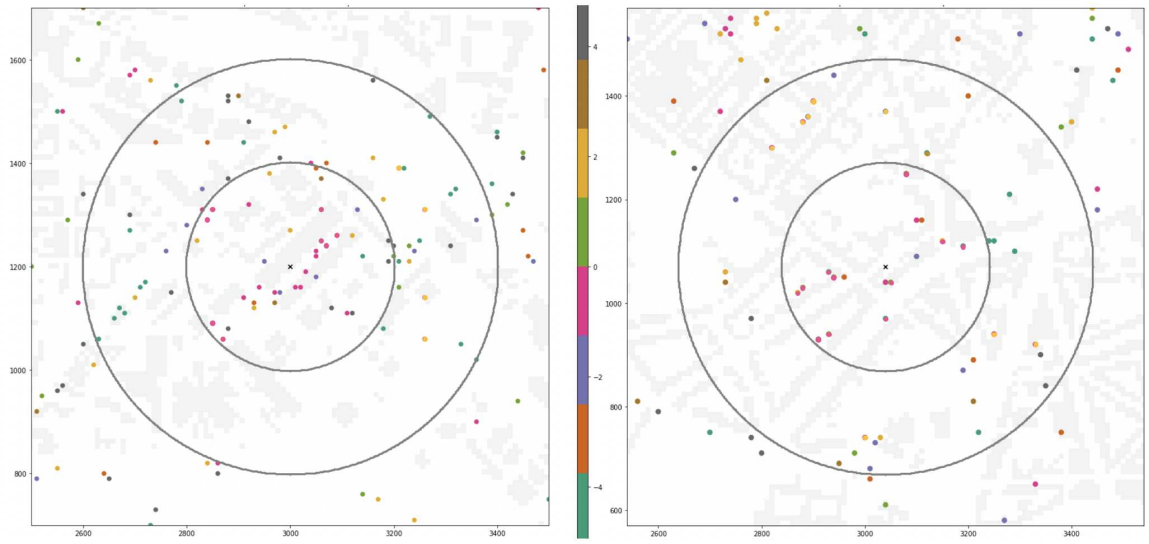
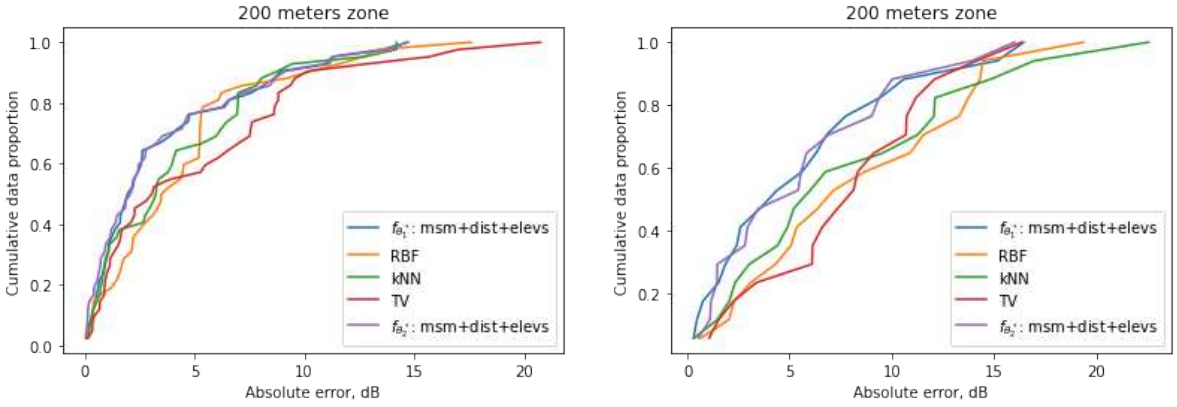


Figure 5.12: Heatmap of the errors, dB, between true and predicted values of, $f_{\theta_2^\star}$, over test base stations in Grenoble (left) and Antwerp (right).

the test base stations in Grenoble (Figure 5.13, left) and Antwerp (Figure 5.13 right). From these results, it comes that the probabilities of having less absolute dB error is higher for both $f_{\theta_1^\star}$ and $f_{\theta_2^\star}$ than the other approaches.

In Appendix A.4 we provide more eCDF curves for other zone areas around the test base stations of both cities.

The primary takeaway from these findings is that searching a Neural Network model with generalization capabilities might be useful for RSSI map reconstruction. To further investigate in this direction, we considered Scenario 1 of Algorithm 2 in which the training points of both cities are combined, with the goal of evaluating the model's ability to produce predictions for one of the cities. The average MAE in db



(a) (b)

Figure 5.13: Empirical cumulative distribution function of different techniques in a 200-meter zone around the test base stations in Grenoble (left) and Antwerp (right).

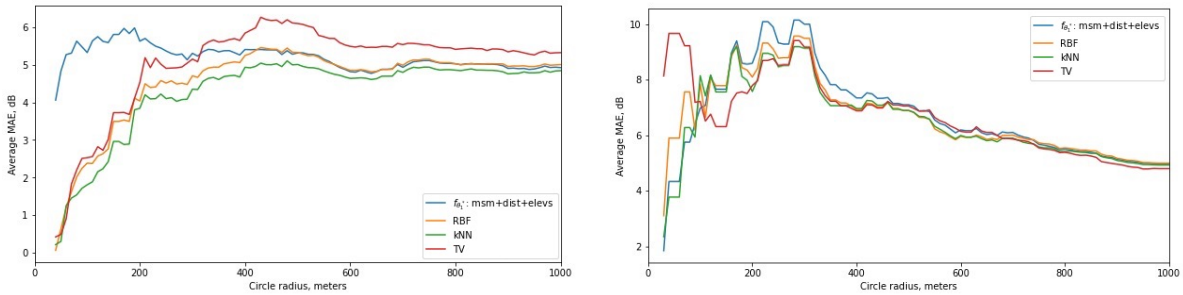


Figure 5.14: MAE over the distance to the base station evaluated over unseen base stations for Grenoble (left) and Antwerp (right), $f_{\theta_1^\star}$ is trained over mixed dataset Grenoble+Antwerp.

with respect to the distance to the base stations for different approaches are shown in Figure 5.14.

According to these findings, the inclusion of signal data from another city disrupts the search for an efficient NN model and learning its parameters. This is most likely owing to the fact that the data distributions in these cities differ, and it would be interesting to study over alignment strategies, such as those proposed for domain adaptation [Kumar et al., 2018], in order to narrow the gap between these distributions in future work.

5.5 Conclusion

In this chapter we studied the importance of the use of additional side-information for the search of an optimized NN architecture for RSSI map reconstruction over three different datasets. We have shown that the addition of distance and elevation of buildings to the measurements allow to significantly reduce the mean absolute error in dB of the obtained NN model with an optimized found architecture. Our proposed approach tends to outperform agnostic techniques especially in the close zone near to the test base stations. We have also shown that our NN based approach has good generalization ability. However, in situations where there exists a distribution shift between two maps, the prediction confidence given by the training model may be highly biased towards, and thus may be not reliable. In reality, a significant difference between two different RSSI maps could lead to complete degradation of the model's performance due to the large error in pseudo-labels. In practice, approaches like confidence regularization [Zou et al., 2019] may reduce the number of wrong pseudo-labels, but theoretically, studying the semi-supervised learning under a distribution shift is an important direction for future work.

Chapter 6

Conclusions and future work

This dissertation has studied the possibility of application of deep learning to the signal strength map reconstruction problem, which has become a main task for the fingerprinting localization technique in order to avoid use of GPS which consumes a lot of energy. In previous works, to solve this task the neural network models trained in the supervised manner mostly over the generated data were shown, while we propose to focus on the real data – collected measurements.

Summary of contributions

In Chapter 4 we proposed the algorithm $\texttt{SL}_{\texttt{NAS}}$ which allows to do the RSSI map reconstruction out of only given measurements and only one Base Station. We compared two different techniques of automatic neural network search, and for all the cases genetic algorithm has shown better results that dynamic routing, but, simultaneously, the learning time for the last one was lower. The proposed algorithm outperformed all the considered baselines, but for the new coming base station, we have to run the algorithm from scratch (which is time consuming) and we do not take into account the examples of other base stations. Additionally, we concluded that we should put more attention in the region close to the base station as here signal varies most.

Consequently, in Chapter 5 we proposed an algorithm which not only solves the generalization issue over one city region, but also takes the advantage of having additional information (like elevations map, distance to the base station, buildings..) thus increasing the accuracy of the output map especially in the close zone near the base station. But before running the search algorithm we have tested the fixed architecture over both generated and real data to see if it is possible to learn the model having only small amount of sparse measurements, which comes from the measurements collection process, and if it is possible to to the generalization with use of additional information.

In both above mentioned chapters we compared two different LoRa datasets of real data collected in urban environments, where one of the dataset is publicly available, while the second one was collected during this PhD and was analyzed in Chapter 2. Additionally, with the extracted path-loss model parameters, we compared the

Cramer-Rao Lower Bound for both conventional model-based parametric model and for fingerprinting positioning technique.

Future perspectives

Regarding the work described in Chapter 4, we plan to incorporate a probabilistic label-noise model in self-learning phase (while choosing additional random points that are further incorporated as the pseudo-labelled examples) and to learn simultaneously the parameters of the Neural Network and the label-noise models.

As for the future work regarding Chapter 5, we want to investigate the sources of artefacts in the output images, and further consider domain adaptation techniques to leverage the shift between the shifts of data distributions. Some of the existing works motivate the rationality of idea, but still they were mostly done with generated or full known data [Zhu et al., 2020, Gill et al., 2021]. In [Li et al., 2019] they have tried quite similar technique for different type of signal, but data sparsity is much lower than in our case (they have tested the case of 30-40% of available data per sample, while in our case we have less than 10% of available pixels per image sample). Particular research is underway, even though without sufficient results (it could be found in Appendix A.2), but we have some ideas how to overcome the existing problems (as this method was studied before application of `NAS` technique).

Furthermore, we want to explore the task of transfer learning from one city to another to solve more wide problem of generalization.

Appendix A

Appendices

A.1 Calculation of Positioning CRLB based on a Link-wise Parametric Path Loss Model

According to Eq. 2.1 and 2.2, we assume the average received power model for each base station i to be:

$$P_i^{dBm}(\theta) = P_{0,i}^{dBm} - 10n \log_{10} \frac{d_i(\theta)}{d_{0,i}} + w_i, \tag{A.1}$$

where $d_i(\theta) = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ is the distance from a base station i of 2D Cartesian coordinates (x_i, y_i) and some point on the map with coordinates $\theta = (x, y)$, $P_{0,i}^{dBm}$ is a free-space average received power at a reference distance $d_{0,i}$, $i \in \overline{1, N}$, $w_i \sim \mathcal{N}(0, \sigma_i^2)$. For simplicity, we denote $P_i^{dBm}(\theta) = P_i$, $d_i(\theta) = d_i$ and assume $d_{0,i} = 1$ in the following.

Assuming the average received power measurements from the N base stations to be independent, their joint probability density function is given by:

$$f(\mathbf{P}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\Big(-\frac{(P_i - (P_0 - \frac{10n_i \ln d_i}{\ln 10}))^2}{2\sigma_i^2}\Big) \tag{A.2}$$

Let us now denote $a_i = P_i - P_0$, $b_i = \frac{10n_i}{\ln 10}$, then it yelds the following log-likelihood function:

$$\mathcal{L} = \log f(\mathbf{P}) = \log \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\Big(-\frac{(a_i + b_i \ln d_i)^2}{2\sigma_i^2}\Big) = C_1 + \sum_{i=1}^{N} \Big(-\frac{(a_i + b_i \ln d_i)^2}{2\sigma_i^2}\Big), \tag{A.3}$$

where C_1 is a constant wrt θ.

The CRLB is defined as the inverse of the Fisher Information Matrix (FIM) [Kay, 1993]:

$$\mathbf{F} = -\mathbb{E}[\nabla_\theta (\nabla_\theta l(\theta))^T]$$

In our case, the FIM consists of four elements, as follows:

$$\mathbf{F} = \begin{pmatrix} F_{xx} & F_{xy} \\ F_{yx} & F_{yy} \end{pmatrix} \tag{A.4}$$

where $F_{xx} = -\mathbb{E}\frac{\partial^2 \mathcal{L}}{\partial x^2}$, $F_{yy} = -\mathbb{E}\frac{\partial^2 \mathcal{L}}{\partial y^2}$, $F_{xy} = F_{yx} = -\mathbb{E}\frac{\partial \mathcal{L}}{\partial x}\frac{\partial \mathcal{L}}{\partial y}$.

We thus first compute the partial first and second derivatives of the log-likelihood function in A.3 with respet to x, as follows:

$$\frac{\partial \mathcal{L}}{\partial x} = \sum_{i=1}^{N} -\frac{b_i}{2\sigma_i^2}(2(a_i + b_i \ln d_i)\frac{1}{d_i}\frac{1}{2}\frac{2(x - x_i)}{d_i} = \sum_{i=1}^{N} -\frac{b_i}{2\sigma_i^2}\frac{a_i + b_i \ln d_i}{d_i^2}(x - x_i) \quad \text{(A.5)}$$

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial x^2} &= -\sum_{i=1}^{N} \frac{b_i}{\sigma_i^2}\Big[b_i \frac{1}{d_i}\frac{(x - x_i)}{d_i}\frac{1}{d_i^2}(x - x_i) + (a_i + b_i \ln d_i)(-2)\frac{1}{d_i^4}(x - x_i)^2 + (a_i + b_i \ln d_i)\frac{1}{d_i^2}\Big] \\ &= -\sum_{i=1}^{N} \frac{b_i^2}{\sigma_i^2}\Big[\frac{(x - x_i)^2}{d_i^4}(1 - 2\ln d_i) + \frac{(x - x_i)^2}{d_i^4}\frac{(-2)a_i}{b_i} + \frac{a_i}{b_i d_i^2} + \frac{\ln d_i}{d_i^2}\Big]\end{aligned} \quad \text{(A.6)}$$

Then, we take the expectation of A.6, which also requires the expectation of a_i:

$$\begin{aligned}\mathbb{E}[a_i] &= \int_{-\infty}^{+\infty} (P_i - P_0) f(\xi)\, d\xi = \int_{-\infty}^{+\infty} (\frac{-10 n_i \ln d_i}{\ln 10} + \xi) f(\xi)\, d\xi \\ &= \frac{-10 n_i \ln d_i}{\ln 10} \underbrace{\int_{-\infty}^{+\infty} f(\xi)\, d\xi}_{\mathbf{=1}} + \underbrace{\int_{-\infty}^{+\infty} \xi f(\xi)\, d\xi}_{\textbf{mean value, =0}} = -b_i \ln d_i\end{aligned} \quad \text{(A.7)}$$

Taking the expectation of Eq. A.6 and given Eq. A.7, it comes:

$$\begin{aligned}F_{xx} = -\mathbb{E}\frac{\partial^2 \mathcal{L}}{\partial x^2} &= -\sum_{i=1}^{N} \frac{b_i^2}{\sigma_i^2}\Big[\frac{(x - x_i)^2}{d_i^4}(1 - 2\ln d_i) + \frac{(x - x_i)^2}{d_i^4}\frac{2 b_i \ln d_i}{b_i} - \frac{b_i \ln d_i}{b_i d_i^2} + \frac{\ln d_i}{d_i^2}\Big] \\ &= \sum_{i=1}^{N} \frac{b_i^2}{\sigma_i^2}\frac{(x - x_i)^2}{d_i^4}\end{aligned} \quad \text{(A.8)}$$

Similarly, the term F_{yy} is shown to be calculated as:

$$F_{yy} = \sum_{i=1}^{N} \frac{b_i^2}{\sigma_i^2}\frac{(y - y_i)^2}{d_i^4} \quad \text{(A.9)}$$

and the term F_{xy} (or F_{yx}) as:

$$\frac{\partial^2 \mathcal{L}}{\partial x \partial y} = -\sum_{i=1}^{N} \frac{b_i}{\sigma_i^2}\Big[\frac{b_i}{d_i}\frac{(y - y_i)}{d_i}\frac{(x - x_i)}{d_i^2} + (a_i + b_i \ln d_i)\frac{(-2)}{d_i^4}(x - x_i)(y - y_i)\Big] \quad \text{(A.10)}$$

$$F_{xy} = -\mathbb{E}\frac{\partial^2 \mathcal{L}}{\partial x \partial y} = \sum_{i=1}^{N} \frac{b_i^2}{\sigma_i^2}\frac{(x - x_i)(y - y_i)}{d_i^4} \quad \text{(A.11)}$$

Finally, to compute the Position Error Bound (PEB), which characterizes the best covariance for the estimation of $\theta = (x, y)$, we need take the trace of the inverse of the FIM in Eq. A.4, leading to:

$$var \geq PEB = \frac{F_{xx} + F_{yy}}{F_{xx}F_{yy} - F_{xy}^2} \tag{A.12}$$

Note that, in case $b_i = b$ and $\sigma_i = \sigma \ \forall i$, the final formulas are exactly similar to that in [Patwari and Hero, 2002], for the non-cooperative localization of one mobile tag.

A.2 Generative Adversarial Network for map reconstruction

We have tried the Conditional Generative Adversarial Network (GAN) version with multi-channel input with distances and/or buildings count (at that moment there was no available elevation maps). We defined the following instantaneous loss ℓ_G in (**??**) or (5.1) to train the generator G_{θ_G} with parameters θ_G:

$$\begin{aligned}\ell_G(y, G_{\theta_G}) = \alpha_1 \underbrace{[D(G_{\theta_G}(x_i)) - 1]^2}_{\textbf{least square term}} + \alpha_2 \underbrace{MAE(y, G_{\theta_G}(x))}_{\textbf{input measurements term}} + \alpha_3 \underbrace{\texttt{TV}(G_{\theta_G}(x))}_{\textbf{gradients smoothness}} \\ + \lambda_G \underbrace{||\theta_G||_2^2}_{\textbf{weights regularization}}\end{aligned} \tag{A.13}$$

where x_i is a signal strength value in either S_ℓ (for the supervised learning scenario) or $S_\ell \cup S_u^{j,i}$ (for the semi-supervised learning scenario). Different terms that intervene in (A.13) are:

- *least square term*, which forces the generator to produce the outputs as close to the given data so that the discriminator will do wrong prediction over the generated one;
- *input measurements term* which reduces the error between the input and output pixels as the ground truth data should be present in the final result with a minimal error;
- *gradients smoothness term*, or *total variation*, which allows to keep the smoothness between adjacent pixels;
- *weights regularization term*, which does not allow the model to overfit for the given training data.

$\alpha_1, \alpha_2, \alpha_3, \alpha_4$ and λ_G are the corresponding hyperparameters. Respectively, we defined the following instantaneous loss ℓ_D to train the discriminator D_{θ_D} with parameters θ_D

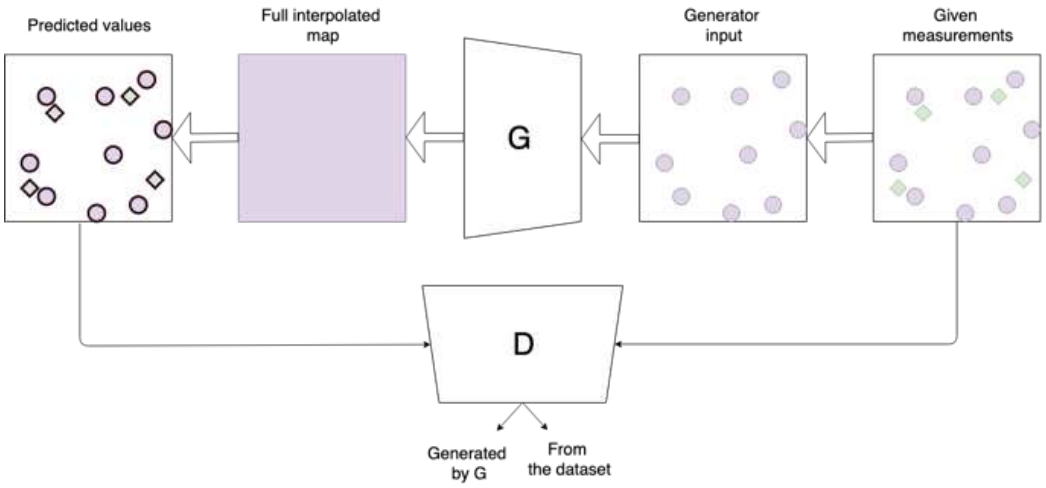


Figure A.1: Schema of learning procedure in GAN (specifically, measurements division). Training points from the masks M_l are given as an input to Generator (G) while in the Discriminator (D) we have both training and validation points from masks M_l and M_v

$$\ell_D(y, D_{\theta_D}) = \alpha_0[D_{\theta_D}(x) - 1]^2 + [D_{\theta_D}(G_{\theta_G}(x) - y]^2 + \lambda_D||\theta_D||_2^2 \tag{A.14}$$

Eq. A.14 consists of two main terms: also *least square term* as for the generator but it forces the discriminator network to distinguish correctly the real data images and generated ones, thus computing the error between the assigned 0 for generated and 1 for real labels, and *weights regularization term* to reduce the probability of overfitting. In the Figure A.1 there is an idea of learning procedure without using pseudo-labeling technique. In the loss function there only given measurements pixels were considered, as well as in the inputs to the generator and discriminator.

After training the model with two additional channels (distances and buildings count) and receiving the results over unseen base station (Figure A.2) we had a look over the gradients behaviour for both RBF and GAN outputs, see Figure A.3. Even though the amount of points with gradients that less than 0.001 for both x and y directions is smaller for GAN output than for RBF, and the computed MAE over test points gives slightly better result (GAN 4.02 dB, RBF 4.59 dB), the randomness of gradients does not correspond to the fact that we expect their smoothness in all the image, see Chapter 2.4. As it was found further, these artefacts come from the `Deconv2D` standard layer which creates checkerboard artefacts after returning to the input size by increasing the intermediate outputs. To avoid this, the combination of `Upsampling2D` and `Conv2D` has replaced the initial layer as it is the analogy of the operations that are done in the `Deconv2D` layer. But the parameters space has been increased and numerical results with similar accuracy were not achieved. Furthermore, inducing the side information is tricky in general (either to give it as an additional channel input or to proceed it separately to extract its local features).

From different experiments we have conducted and based on several tests we ran

by adjusting different variables in the losses of the Generator and the Discriminator, we found out that with the lack data, the optimization of the losses of both models is unstable. Our key finding is that learning GAN from sparse observations is problematic for RSSI map reconstruction. As a result, we concentrated our efforts on using UNet with an efficient architecture for this purpose.

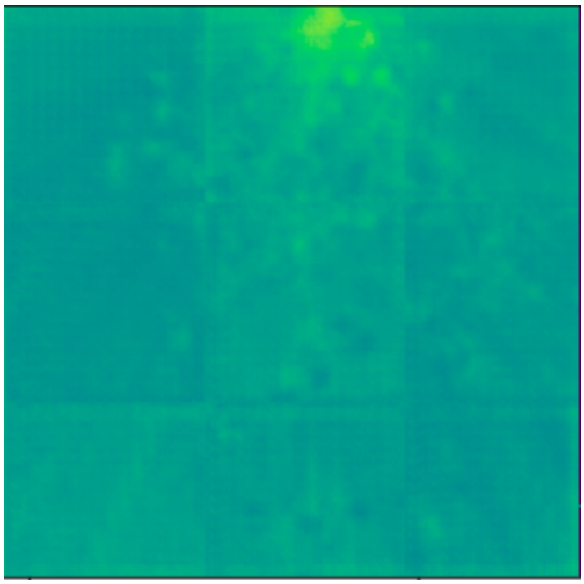


Figure A.2: Received output of the conditional GAN for the test base station. It seems to give visually good interpolation, but the output contains rectangular patterns on the top – checkerboard artefacts

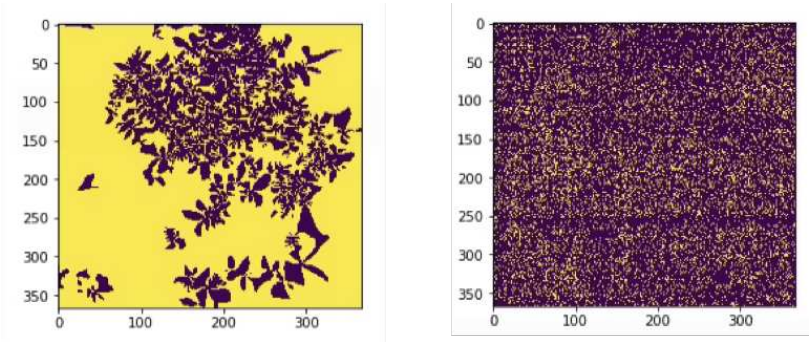


Figure A.3: Gradients behaviour over test base station. Binary mask of the points, where sum over x and y gradients is below some small value (for each axis less than 0.001; in yellow is small). *Left:* RBF interpolation, *Right:* GAN interpolation

A.3 Errors behaviour during learning process over Paris dataset

In the Figure A.4 there is an example of errors changing during the learning process in the area around the base station position. We can see that with the progress of learning the amount of underestimated regions is getting smaller (in the image there are less red and yellow regions), while still there are some regions where the predicted values are higher than in the ground truth data, especially close to the buildings positions, which could be correlated with the sharp generated data in this regions.

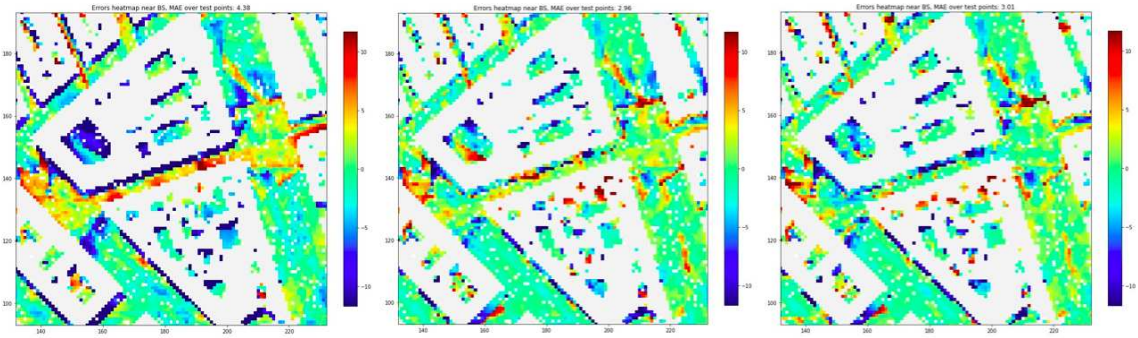


Figure A.4: Evolution of the errors distribution during learning process with the base station in the center of the figure (excluding the buildings area). *From left to right:* epoch 1, epoch 40, epoch 90

A.4 Additional experiments

UNet fixed. Results over the real dataset. In this part we moved to the real data case setting A, as there we have the biggest amount of available data. As it was mentioned before, we have considered two different settings with different scenarios respectively (Eq. 5.1), and one more goal in this section is to evaluate the result over the Antwerp dataset (setting A) and thus test unseen base station from this city. We also considered, as the additional information, distances to the base station location. During the training process, we left randomly chosen 90% of the measurements to the train and 10% of the measurements to the validation. The test base station BS'_9 was in the same conditions as we described in Section 5.3 – 90% of measurements were in the train, and 10% in the test.

To compare the results with model without side information, we have conducted the experiment with the same learning process as for the generated dataset but without information about the distances. In the Figure A.5 there is shown a MAE changes over the distance, the overall performance appeared to be worse than with use of information about the distances, which also validates the possibility of the side information to improve the final result.

Additionally, to check also influence of the loss function, we conducted following experiments. First, we considered the loss function over the validation pixels, as in

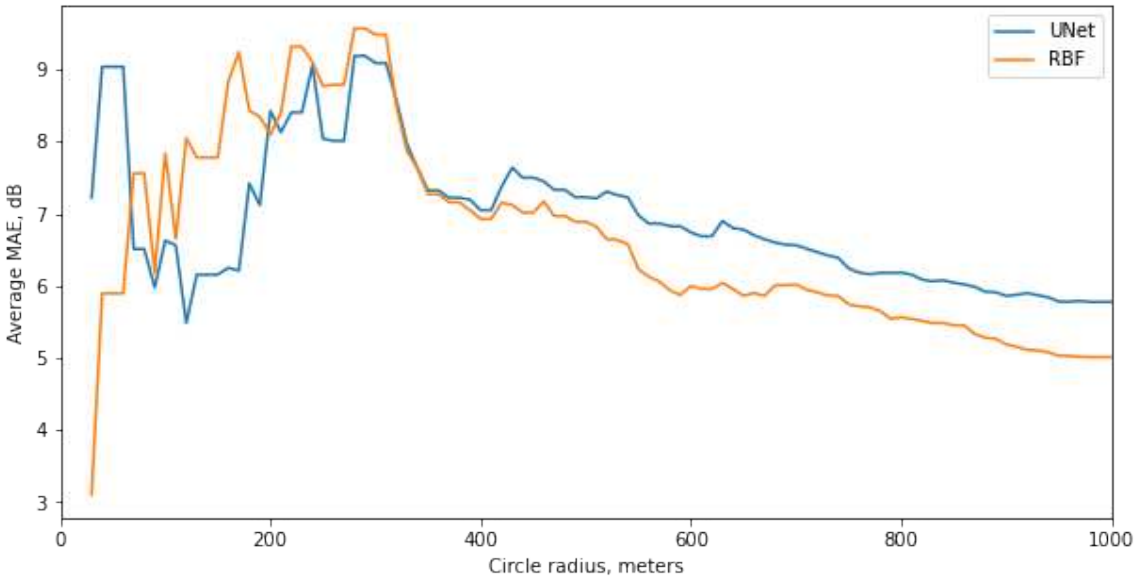


Figure A.5: Cumulative MAE distribution according to the distance to the base station. Comparison between baseline and UNet model trained only over given measurements map, Antwerp test base station

scenario 1. Because of high sparsity of measurements, the model based on the known measurements only as a final result does not give generalization properties as it tries to catch the behavior only over the point in the training dataset. Consequently, the result over the test base station is worse in comparison with the RBF interpolation, as it could be seen from the Figure A.6.

That is why loss from the Eq. 5.1, scenario 2 was further considered, where the RBF interpolation output (without taking into account pixels with buildings) was taken as a ground truth in the loss function. Moreover, the error over all the test points was lower, see Figure A.7.

Besides, use of additional channel in the input to the model (like buildings, elevations..) did not give an improvement both during the learning process and after testing over the unseen base station. This could be because of difficulty to interpret the additional information with this exact type of the network.

Comparison of the experiments above has shown the potential to use the baseline outputs as pseudo-labels for the learning process and motivated the use of neural architecture search algorithm to find (probably) better architecture in terms of accuracy in the region near the base station. These results will be covered in the next paragraphs.

Three-channel input: measurements, distances, elevation maps; setting A. As it was mentioned before, in the genetic algorithm of searching for a model there is a ranking metric which helps to select the best one. We have conducted two types of choosing the model and thus the type of metric: by `NMAE` or by `MAE` validation loss (by `NMAE` we mean the multiplication of the errors by the distances map thus

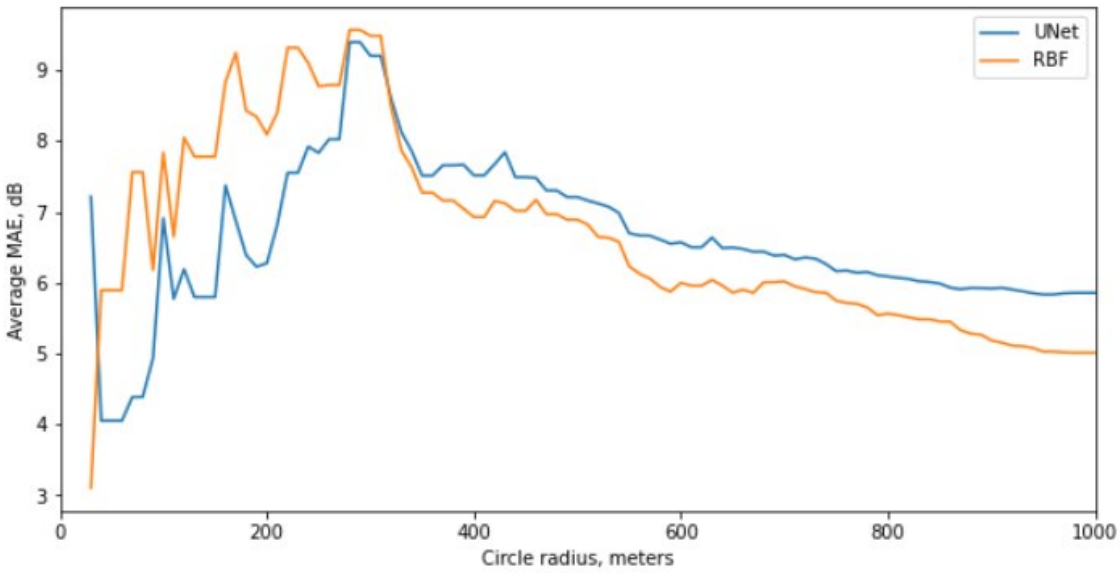


Figure A.6: Cumulative MAE distribution according to the distance to the base station location. Comparison between the baseline and UNet model trained over given measurements and distances and with loss from Eq. **??**, Antwerp test base station

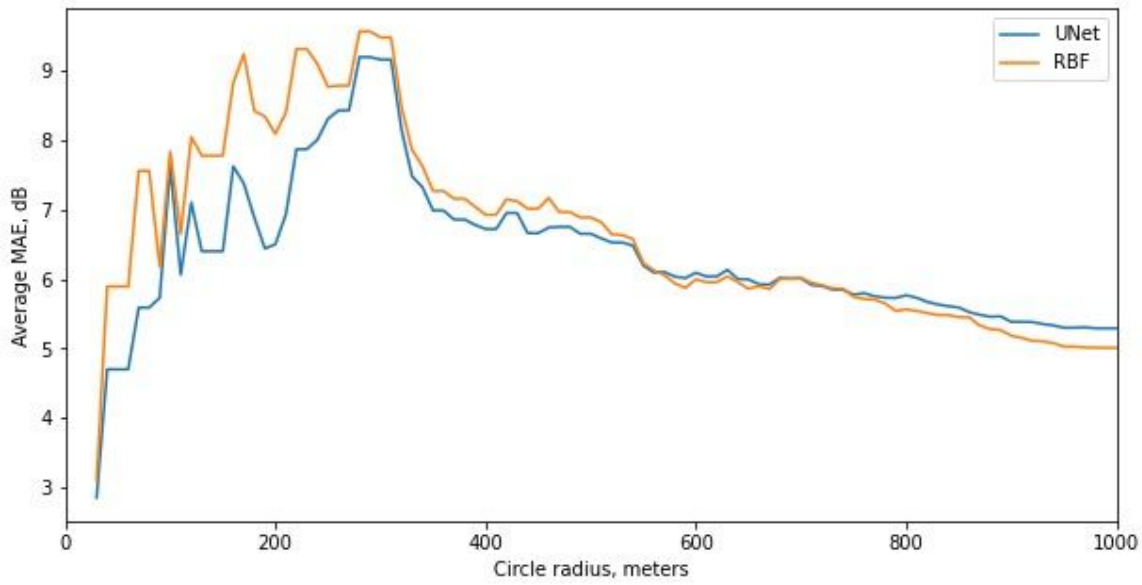


Figure A.7: Cumulative MAE distribution according to the distance to the base station. Comparison between baseline and UNet model trained over given measurements map and distances map input, loss from Eq.5.1, Antwerp test base station

weighting higher the points closer to the base station location). As a result, we have received two best models f_{θ_1} and f_{θ_2} for `NMAE` and `MAE` criteria respectively. In Figure A.8 we can see its comparison for the zone near the test base station. Ranking the

model by NMAE allows to have lower cumulative error in the close zone while the final metric over all points approximately the same. This approves the fact that weighting the higher values by using NMAE (which are closer to the base station) with a higher weight during the model ranking process over the unseen part of the dataset gives more accurate result in the zone close to the base station.

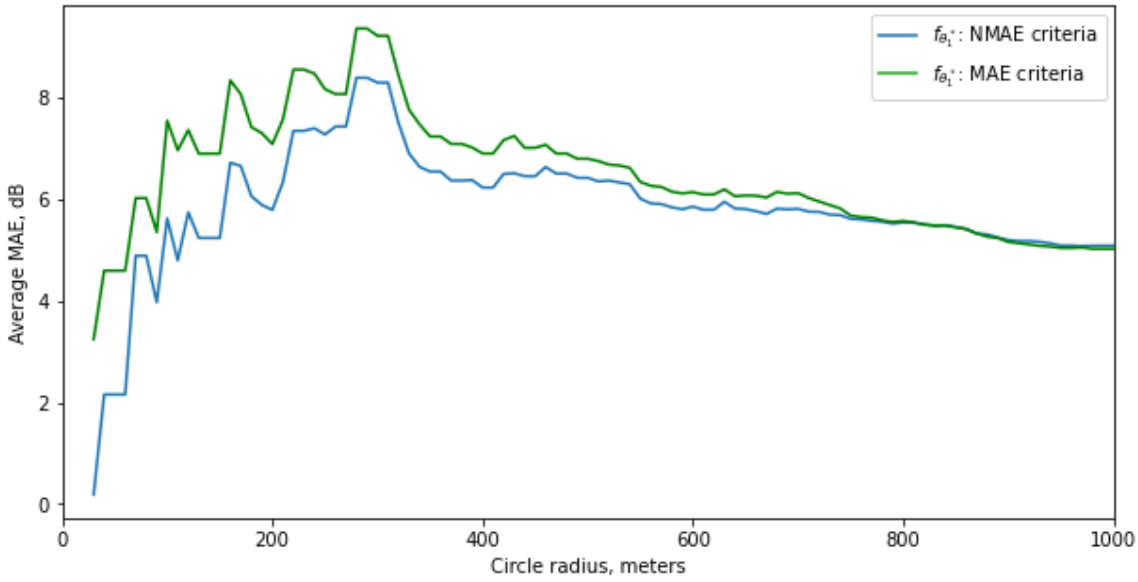
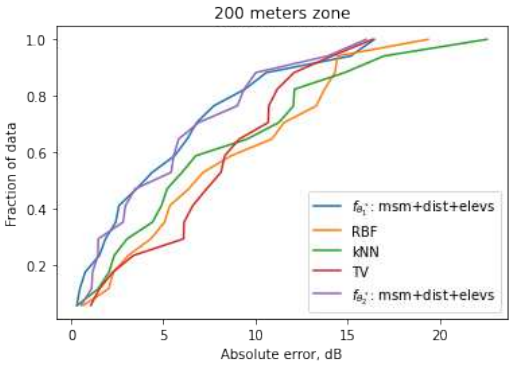


Figure A.8: Cumulative MAE distribution according to the distance to the test base station; comparison of two score metrics for two NAS algorithms, Antwerp test base station

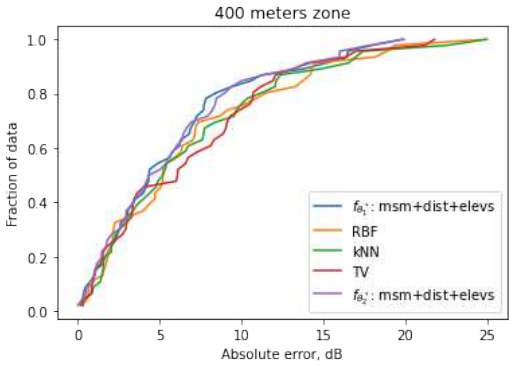
Algorithm	NMAE, 200m	NMAE, 400m	NMAE, 800m
RBF	5.33	3.90	2.51
kNN	5.08	3.88	2.46
TV	4.86	3.85	2.44
NAS, msm+dist+elev	3.72	3.43	2.43

Table A.1: Comparison of NMAEs over the points in the mentioned radius for different baselines, BS'_9

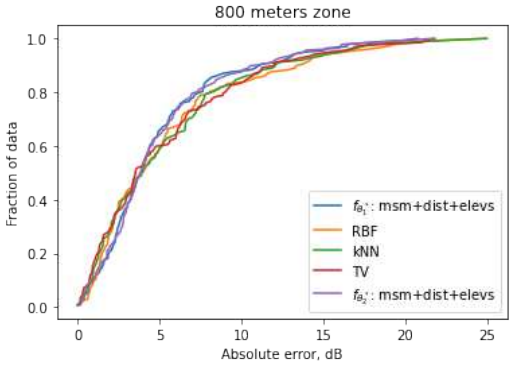
Final comparison of ECDF percentiles between the final model architectures and agnostic baselines. In the Table A.2 there is shown numerical comparison of the key percentiles of built empirical CDF over the test points for all the considered baselines as well as two last methods, namely the network received by NAS trained over the mixed data or the each city separately with a three channel input (measurements, distances, elevations), which has shown the improvement in comparison with model only over the measurements, measurements + distances. As it was also discussed above, the mixed data case is not showing good performance for all the



(a) 200 meters from the base station



(b) 400 meters from the base station



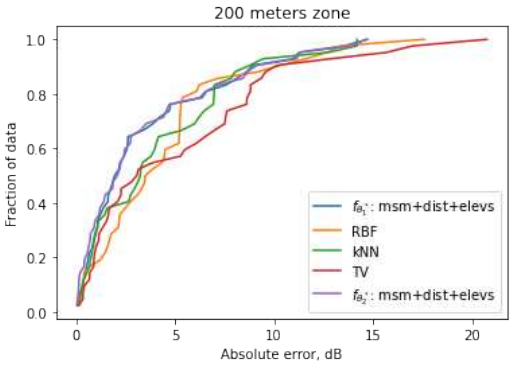
(c) 800 meters from the base station

Figure A.9: ECDF plots of errors for the key distances over test Antwerp dataset. Comparison between baselines and NAS model trained over Antwerp dataset, BS'_9

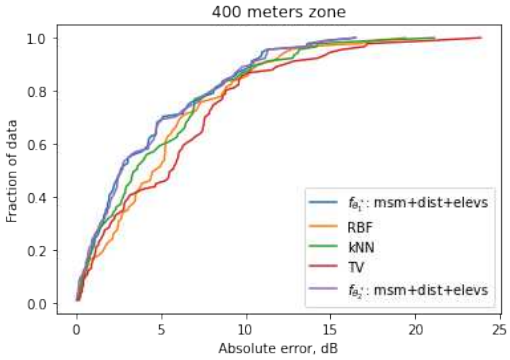
key distances, while for models trained at the city level the generalization works much better than the considered baselines. Thus we can see that by use of combination of input measurements and additional information in local area with common features can solve the generalization problem that was stated above, and adding the meta information can improve the result.

Mixed data In the Figures A.11 and A.12 there are shown ECDF plots for each of the considered close regions. Especially in very close zones (like 200m and 400m radiuses) the proportion of bigger errors is higher than for the other baselines, but with increase of considered radius it becomes indistinguishable, see Figures A.11c, A.12c.

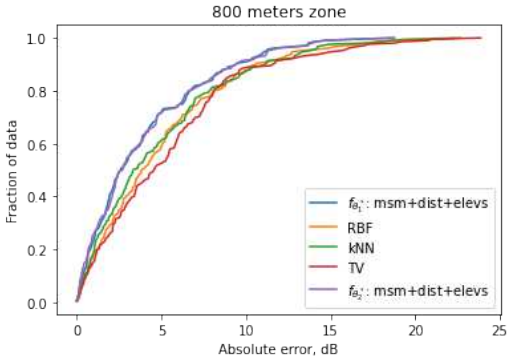
Final comparison of ECDF percentiles between the final model architectures and agnostic baselines. In the Table A.2 there is shown numerical comparison of the key percentiles of built empirical CDF over the test points for all the considered baselines as well as two last methods, namely the network received by `NAS` trained over the mixed data or the each city separately with a three channel input (measurements, distances, elevations), which has shown the improvement in compar-



(a) 200 meters from the base station



(b) 400 meters from the base station



(c) 800 meters from the base station

Figure A.10: ECDF plots of errors for the key distances over test Grenoble base station. Comparison between baselines and NAS model trained over Grenoble dataset, BS_4

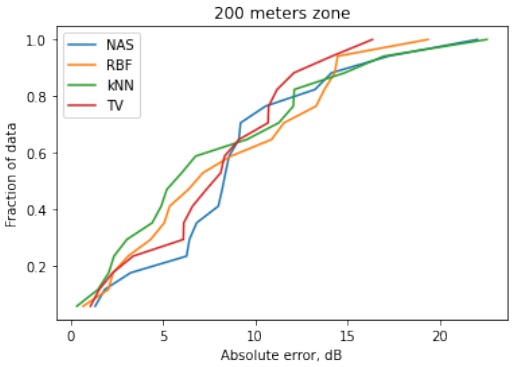
ison with model only over the measurements, measurements + distances. As it was also discussed above, the mixed data case is not showing good performance for all the key distances, while for models trained at the city level the generalization works much better than the considered baselines. Thus we can see that by use of combination of input measurements and additional information in local area with common features can solve the generalization problem that was stated above, and adding the meta information can improve the result.

In the Figures A.14, A.13 there are shown binary maps with points, where gradients are small. For `TV` algorithm as its goal is to minimize the gradients between adjusted points the area with small gradients is much bigger that for other algorithms. For the kNN output we have quite similar behaviour, but in the zones of division into clusters the gradients are higher. For the `RBF` interpolation the gradients are small where there was no measurements, bigger gradients are presented in the local areas. For the `NAS` interpolation the area with low gradients is smaller, but still it is close to the behaviour of `RBF` interpolation as we used this output as a benchmark during the training process. For the future work, there is a direction of exploration of gradients behaviour and how could it affect the final result.

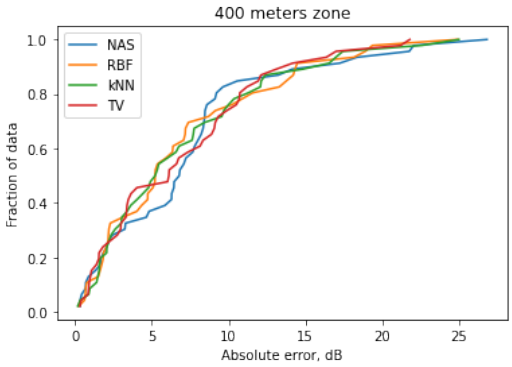
Algo	Percentile	Antwerp, BS'_9			Grenoble, BS_4		
		200m	400m	800m	200m	400m	800m
RBF	0.5	6.77	5.21	3.89	3.49	4.62	3.84
	0.8	13.57	11.43	8.06	**5.80**	8.76	8.17
TV	0.5	7.74	6.10	3.54	3.07	5.61	4.57
	0.8	11.01	10.72	8.99	8.74	8.82	8.08
kNN	0.5	5.61	5.21	3.94	3.24	3.48	3.34
	0.8	12.10	11.09	8.18	6.97	**7.99**	7.85
$f_{\theta_1^*}$	0.5	8.29	6.78	3.62	4.94	4.82	3.93
mixed data	0.8	12.17	9.19	8.38	7.75	8.55	8.27
$f_{\theta_1^*}$	0.5	**3.98**	**4.34**	3.85	**1.95**	**2.62**	**2.59**
same city	0.8	**8.75**	**8.36**	**7.50**	6.50	8.12	**6.66**
$f_{\theta_2^*}$	0.5	4.24	**4.35**	3.86	2.08	**2.69**	2.68
same city	0.8	9.24	8.90	**7.23**	6.43	**7.98**	**6.63**

Table A.2: Comparison of the key parameters of CDF for different cities and settings. In bold best result in the column for corresponding percentile

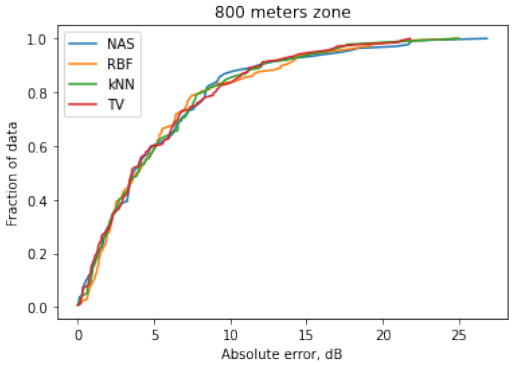
In the Figures A.15, A.16 there is an example of map interpolation in the zones around the base station. `RBF` output seems to have much more local effects, while for `NAS` interpolation it is smoother but still close to the benchmark. One more difference in the outputs lies in the division of the initial images into smaller ones for `NAS`, while for `RBF` it is possible to process full input. We can still see the borders between the cutted images, so there is a work for the future to find out how to reduce these effects in the final interpolation.



(a) 200 meters from the base station



(b) 400 meters from the base station

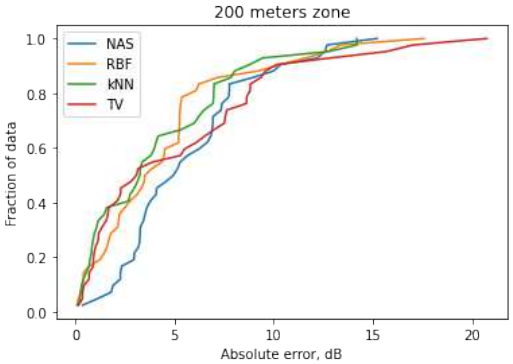


(c) 800 meters from the base station

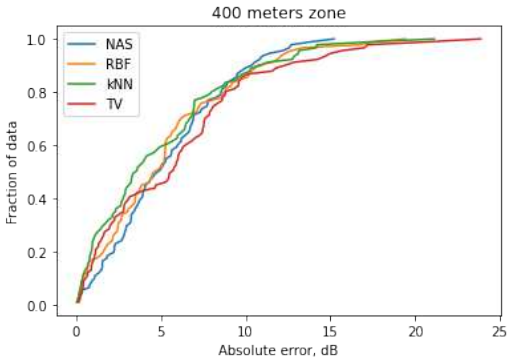
Figure A.11: ECDF plots of errors for the key distances over test Antwerp base station. Comparison between baselines and NAS model trained over mixture of two datasets: Antwerp and Grenoble

City	Method	200m	400m	800m
Grenoble BS_4	RBF	4.49	5.29	5.00
	kNN	4.20	4.92	4.84
	TV	5.19	5.91	5.42
	$f_{\theta_1^*}$	3.62	4.32	3.94
	$f_{\theta_1^*}$-mixed	5.69	5.39	5.00
	$f_{\theta_2^*}$	3.56	4.32	3.95
Antwerp BS'_9	RBF	8.40	6.93	5.52
	kNN	8.00	6.96	5.43
	TV	7.96	6.88	5.35
	$f_{\theta_1^*}$	6.33	6.22	5.53
	$f_{\theta_1^*}$-mixed	9.10	7.34	5.48
	$f_{\theta_2^*}$	6.01	6.20	5.45

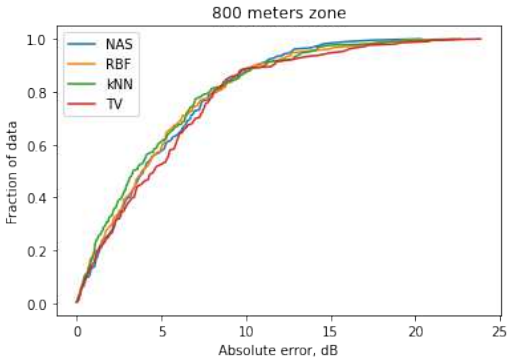
Table A.3: MAEs, dB, for each of the supposed distances over test base stations for different cities and methods



(a) 200 meters from the base station

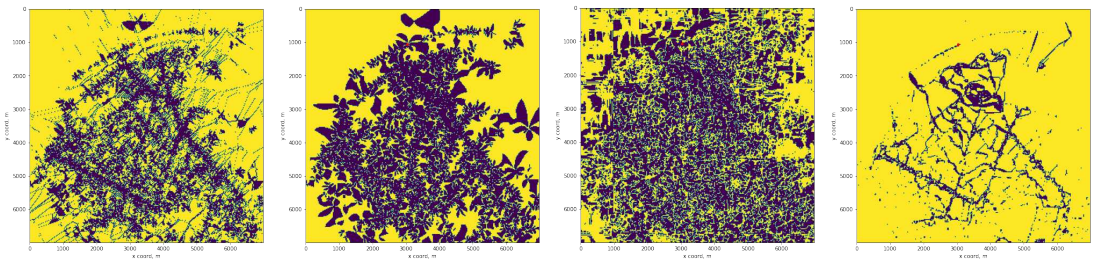


(b) 400 meters from the base station



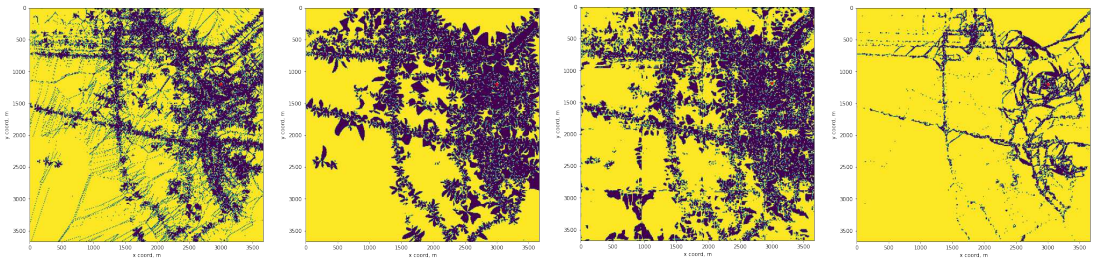
(c) 800 meters from the base station

Figure A.12: ECDF plots of errors for the key distances over test Grenoble base station. Comparison between baselines and NAS model trained over mixture of two datasets: Antwerp and Grenoble



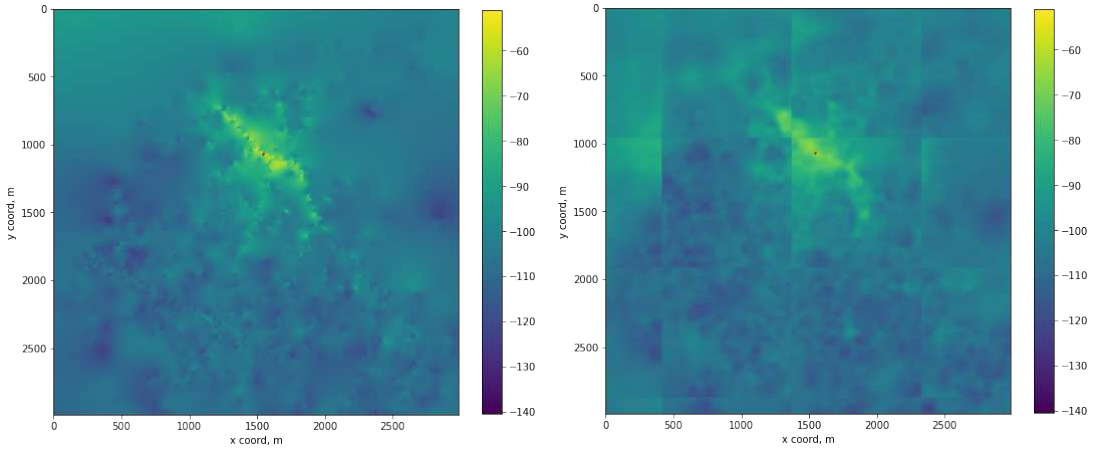
(a) kNN (b) RBF (c) $f_{\theta_1^\star}$ (d) TV

Figure A.13: Gradients behaviour over test base station. Binary mask of the points, where sum over x and y gradients is below some small value (for each axis less than 0.0001; in yellow is small), Antwerp test base station BS_9'



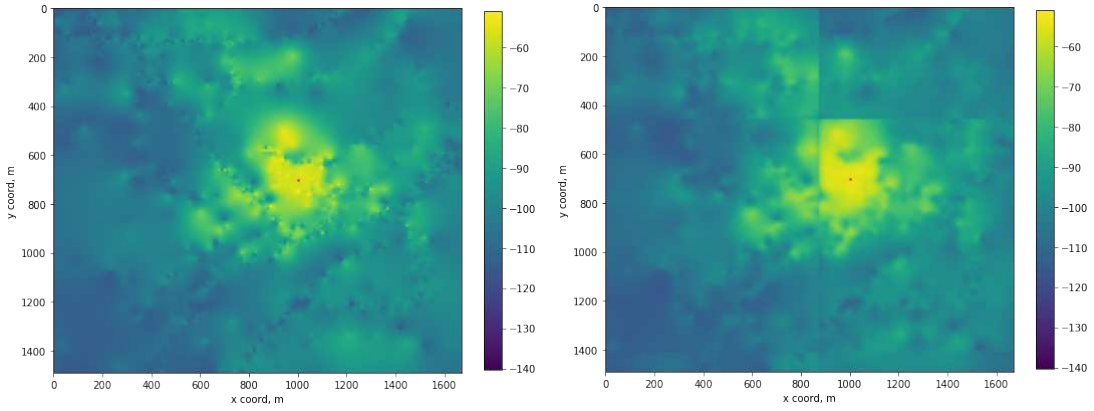
(a) kNN (b) RBF (c) $f_{\theta_1^\star}$ (d) TV

Figure A.14: Gradients behaviour over test base station. Binary mask of the points, where sum over x and y gradients is below some small value (for each axis less than 0.0001; in yellow is small), Grenoble test base station BS_4



(a) RBF (b) $f_{\theta_1^\star}$

Figure A.15: Interpolated zone near Antwerp base station BS'_9, signal strength in dBm



(a) RBF (b) $f_{\theta_1^\star}$

Figure A.16: Interpolated zone near Grenoble base station BS_4, signal strength in dBm

Bibliography

[SAR,] SARWS project. `https://bscw.celticnext.eu/pub/bscw.cgi/d31498/SARWS_leaflet-start_lq.pdf`. Accessed: 2020-10-08.

[Aernouts, 2022] Aernouts, M. (2022). *Localization with low power wide area networks*. PhD thesis, University of Antwerp.

[Aernouts et al., 2018] Aernouts, M., Berkvens, R., Vlaenderen, K. V., and Weyn, M. (2018). Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas. *Data*, 3(2).

[Amini et al., 2022] Amini, M., Feofanov, V., Pauletto, L., Devijver, E., and Maximov, Y. (2022). Self-training: A survey. *CoRR*, abs/2202.12040.

[Amini et al., 2009] Amini, M. R., Usunier, N., and Laviolette, F. (2009). A transductive bound for the voted classifier with an application to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 65–72.

[Bartlett and Mendelson, 2002] Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482.

[Basu et al., 2002] Basu, S., Banerjee, A., and Mooney, R. J. (2002). Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 27 – 34.

[Bertalmio et al., 2001] Bertalmio, M., Bertozzi, A. L., and Sapiro, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. In *CVPR*.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[Boeing, 2017] Boeing, G. (2017). Osmnx: A python package to work with graph-theoretic openstreetmap street networks. *Journal of Open Source Software*, 2(12):215.

[Brau et al., 2012] Brau, M., Corre, Y., and Lostanlen, Y. (2012). Assessment of 3d network coverage performance from dense small-cell lte. In *2012 IEEE International Conference on Communications (ICC)*, pages 6820–6824.

[Burghal et al., 2020] Burghal, D., Ravi, A. T., Rao, V., Alghafis, A. A., and Molisch, A. F. (2020). A comprehensive survey of machine learning based localization with wireless signals.

[Cheng et al., 2012] Cheng, L., Wu, C., Zhang, Y., Wu, H., Li, M., and Maple, C. (2012). A survey of localization in wireless sensor network. *International Journal of Distributed Sensor Networks*, 2012.

[Choi et al., 2018] Choi, W., Chang, Y.-S., Jung, Y., and Song, J. (2018). Low-power lora signal-based outdoor positioning using fingerprint algorithm. *ISPRS International Journal of Geo-Information*, 7(11).

[Cohen et al., 2004] Cohen, I., Cozman, F. G., Sebe, N., Cirelo, M. C., and Huang, T. S. (2004). Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1553–1567.

[Cozman and Cohen, 2002] Cozman, F. G. and Cohen, I. (2002). Unlabeled data can degrade classification performance of generative classifiers. In *Fifteenth International Florida Artificial Intelligence Society Conference*, pages 327–331.

[Dargie and Poellabauer, 2010] Dargie, W. and Poellabauer, C. (2010). *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & Sons.

[Devries and Taylor, 2017] Devries, T. and Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552.

[Elsken et al., 2019] Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21.

[Enrico and Redondi, 2018] Enrico, A. and Redondi, C. (2018). Radio Map Interpolation Using Graph Signal Processing. *IEEE Communications Letters*, 22(1):153–156.

[Fan et al., 2018] Fan, X., He, X., Xiang, C., Puthal, D., Gong, L., Nanda, P., and Fang, G. (2018). Towards system implementation and data analysis for crowdsensing based outdoor RSS maps. *IEEE Access*, 6:47535–47545.

[Gill et al., 2021] Gill, K. S., Nguyen, S., Thein, M. M., and Wyglinski, A. M. (2021). Three-way deep neural network for radio frequency map generation and source localization. *CoRR*, abs/2111.12175.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

[Han et al., 2020] Han et al., X. (2020). A Two-Phase Transfer Learning-Based Power Spectrum Maps Reconstruction Algorithm for Underlay Cognitive Radio Networks. *IEEE Access*, 8:81232–81245.

[Hayashi et al., 2020] Hayashi, T., Nagao, T., and Ito, S. (2020). A study on the variety and size of input data for radio propagation prediction using a deep neural network. In *2020 14th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5.

[Ho et al., 2020] Ho, K., Gilbert, A., Jin, H., and Collomosse, J. (2020). Neural architecture search for deep image prior.

[Inoue et al., 2020] Inoue, K., Ichige, K., Nagao, T., and Hayashi, T. (2020). Radio propagation prediction using deep neural network and building occupancy estimation. *IEICE Communications Express*, advpub.

[Jin et al., 2019] Jin, H., Song, Q., and Hu, X. (2019). Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD*, pages 1946–1956.

[Kamilaris and Ostermann, 2018] Kamilaris, A. and Ostermann, F. (2018). Geospatial Analysis and Internet of Things in Environmental Informatics.

[Kay, 1993] Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., USA.

[Khelifi et al., 2019] Khelifi, F., Bradai, A., Benslimane, A., Rawat, P., and Atri, M. (2019). A Survey of Localization Systems in Internet of Things. *Mobile Networks and Applications*, 24(3):761–785.

[Krisp, 2013] Krisp, J. (2013). *Progress in Location-Based Services*.

[Krithara et al., 2008] Krithara, A., Amini, M.-R., Renders, J.-M., and Goutte, C. (2008). Semi-supervised document classification with a mislabeling error model. In *30th European Conference on Information Retrieval*, pages 370–381, Glasgow.

[Kubota et al., 2013] Kubota, R., Tagashira, S., Arakawa, Y., Kitasuka, T., and Fukuda, A. (2013). Efficient survey database construction using location fingerprinting interpolation. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pages 469–476.

[Kumar et al., 2018] Kumar, A., Sattigeri, P., Wadhawan, K., Karlinsky, L., Feris, R. S., Freeman, B., and Wornell, G. W. (2018). Co-regularized alignment for unsupervised domain adaptation. pages 9367–9378.

[Laaraiedh et al., 2012] Laaraiedh, M., Uguen, B., Stephan, J., Corre, Y., Lostanlen, Y., Raspopoulos, M., and Stavrou, S. (2012). Ray tracing-based radio propagation modeling for indoor localization purposes. In *2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 276–280.

[Levie et al., 2020] Levie, R., Yapar, , Kutyniok, G., and Caire, G. (2020). Pathloss prediction using deep learning with applications to cellular optimization and efficient d2d link scheduling. In *ICASSP*, pages 8678–8682.

[Li and D.Heap, 2011] Li, J. and D.Heap, A. (2011). A review of comparative studies of spatial interpolation methods in environmental sciences: Performance and impact factors. *Ecological Informatics*, 6(3):228 – 241.

[Li et al., 2020] Li, Y., Song, L., Chen, Y., Li, Z., Zhang, X., Wang, X., and Sun, J. (2020). Learning dynamic routing for semantic segmentation.

[Li et al., 2019] Li, Z., Cao, J., Wang, H., and Zhao, M. (2019). Sparsely self-supervised generative adversarial nets for radio frequency estimation. *IEEE Journal on Selected Areas in Communications*, 37(11):2428–2442.

[Liao et al., 2019] Liao, J., Qi, Q., Sun, H., and Wang, J. (2019). Radio environment map construction by kriging algorithm based on mobile crowd sensing. *Wireless Communications and Mobile Computing*, 2019:1–12.

[Lin et al., 2020] Lin, Y., Dong, W., Gao, Y., and Gu, T. (2020). Sateloc: A virtual fingerprinting approach to outdoor lora localization using satellite images. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 13–24.

[Liu et al., 2019] Liu, C., Chen, L.-C., Schroff, F., Adam, H., Hua, W., Yuille, A. L., and Fei-Fei, L. (2019). Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of CVPR*, pages 82–92.

[Liu et al., 2018] Liu, H., Simonyan, K., and Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

[Luo, 2013] Luo, M. (2013). *Indoor radio propagation modeling for system performance prediction*. Theses, INSA de Lyon.

[Malkova et al., 2021a] Malkova, A., Pauletto, L., Villien, C., Denis, B., and Amini, M. (2021a). Self-learning for received signal strength map reconstruction with neural architecture search. Lecture Notes in Computer Science, pages 515–526. Springer International Publishing.

[Malkova et al., 2021b] Malkova, A., Pauletto, L., Villien, C., Denis, B., and Amini, M.-R. (2021b). Techniques d'Auto-Apprentissage et Recherche Automatique d'Architecture Neuronale pour la Reconstruction de Cartes de Puissance Radio. In *Conférence sur l'apprentissage Automatique (CAp)*, Saint-Etienne (en ligne), France.

[Maximov et al., 2018] Maximov, Y., Amini, M., and Harchaoui, Z. (2018). Rademacher complexity bounds for a penalized multi-class semi-supervised algorithm. *J. Artif. Intell. Res.*, 61:761–786.

[Nagao and Hayashi, 2020] Nagao, T. and Hayashi, T. (2020). Study on radio propagation prediction by machine learning using urban structure maps. In *2020 14th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5.

[Ning and et al., 2016] Ning, C. and et al. (2016). Outdoor location estimation using received signal strength-based fingerprinting. *Wireless Pers Commun*, 99:365–384.

[Oliver and Webster, 1990] Oliver, M. and Webster, R. (1990). Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information System*, 4(3):313–332.

[Patwari and Hero, 2002] Patwari, N. and Hero, A. (2002). Location estimation accuracy in wireless sensor networks. In *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002.*, volume 2, pages 1523–1527 vol.2.

[Powell, 1987] Powell, M. J. D. (1987). *Radial Basis Functions for Multivariable Interpolation: A Review*, page 143–167. Clarendon Press, USA.

[Rappaport, 2002] Rappaport, T. (2002). *Wireless communications: Principles and practice*. Prentice Hall communications engineering and emerging technologies series. Prentice Hall, 2nd edition. Includes bibliographical references and index.

[Raspopoulos and et al., 2012] Raspopoulos, M. and et al. (2012). Cross device fingerprint-based positioning using 3D Ray Tracing. In *Proc. Computing Conference (IWCMC'12*, pages 147–152.

[Raspopoulos et al., 2012] Raspopoulos, M., Laoudias, C., Kanaris, L., Kokkinis, A., Panayiotou, C. G., and Stavrou, S. (2012). 3d ray tracing for device-independent fingerprint-based positioning in wlans. In *2012 9th Workshop on Positioning, Navigation and Communication*, pages 109–113.

[Real et al., 2019] Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. (2019). Regularized evolution for image classifier architecture search. In *AAAI*, volume 33, pages 4780–4789.

[Redondi, 2018] Redondi, A. E. C. (2018). Radio map interpolation using graph signal processing. *IEEE Communications Letters*, 22(1):153–156.

[Ronneberger et al., 2015a] Ronneberger, O., Fischer, P., and Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[Ronneberger et al., 2015b] Ronneberger, O., P.Fischer, and Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer.

[Rudin et al., 1992] Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268.

[Rusu and Rusu, 2006] Rusu, V. and Rusu, C. (2006). Radial Basis Functions Versus Geostatistics in Spatial Interpolations. volume 217.

[Sati and Singh, 2014] Sati, G. and Singh, S. (2014). A review on outdoor propagation models in radio communication.

[Sato et al., 2019] Sato, K., Inage, K., and Fujii, T. (2019). On the performance of neural network residual kriging in radio environment mapping. *IEEE Access*, 7:94557–94568.

[Seeger, 2001] Seeger, M. (2001). Learning with labeled and unlabeled data. Technical report.

[Seybold, 2005] Seybold, J. (2005). *Introduction to RF propagation*. John Wiley & Sons.

[Sorour. and et al, 2012] Sorour., S. and et al (2012). RSS Based Indoor Localization with Limited Deployment Load. In *Proc. IEEE Global Communications Conference (GLOBECOM'12)*, pages 303–308.

[Sorour et al., 2015] Sorour, S., Lostanlen, Y., Valaee, S., and Majeed, K. (2015). Joint indoor localization and radio map construction with limited deployment load. *IEEE Transactions on Mobile Computing*, 14(5):1031–1043.

[Suto et al., 2021] Suto, K., Bannai, S., Sato, K., Inage, K., Adachi, K., and Fujii, T. (2021). Image-driven spatial interpolation with deep learning for radio map construction. *IEEE Wireless Communications Letters*, 10(6):1222–1226.

[Tahat et al., 2016] Tahat, A., Kaddoum, G., Yousefi, S., Valaee, S., and Gagnon, F. (2016). A look at the recent wireless positioning techniques with a focus on algorithms for moving receivers. *IEEE Access*, 4:6652–6680.

[Ulyanov et al., 2017] Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Deep image prior. *CoRR*, abs/1711.10925.

[Vapnik, 1982] Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

[Vo and De, 2016] Vo, Q. D. and De, P. (2016). A Survey of Fingerprint-Based Outdoor Localization. *IEEE Communications Surveys & Tutorials*, 18(1):491–506.

[Wilcoxon, 1945] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, 1(6):80–83.

[Yu et al., 2009] Yu, K., Sharp, I., and Guo, Y. (2009). *Ground-Based Wireless Positioning*. John Wiley & Sons, Ltd.

[Zhang and Oles, 2000] Zhang, T. and Oles, F. J. (2000). A probability analysis on the value of unlabeled data for classification problems. In *17th International Conference on Machine Learning*.

[Zhou et al., 2004] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in Neural Information Processing Systems (NIPS 16)*, pages 321–328. MIT Press.

[Zhu et al., 2020] Zhu, D., Cheng, X., Zhang, F., Yao, X., Gao, Y., and Liu, Y. (2020). Spatial interpolation using conditional generative adversarial neural networks. *International Journal of Geographical Information Science*, 34(4):735–758.

[Zhu and Ghahramani, 2002] Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University.

[Zhu et al., 2003] Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In 20^{th} *International Conference on Machine Learning*, pages 912–919.

[Zoph and Le, 2016] Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.

[Zou et al., 2019] Zou, Y., Yu, Z., Liu, X., Kumar, B., and Wang, J. (2019). Confidence regularized self-training. In *International Conference on Computer Vision - ICCV*, pages 5982–5991.