Neural architecture search for radio map reconstruction with partially labeled data

Alexandra Malkova, Massih-Reza Amini, Benoit Denis, Christophe Villien

To cite this version:

Alexandra Malkova, Massih-Reza Amini, Benoit Denis, Christophe Villien. Neural architecture search for radio map reconstruction with partially labeled data. Integrated Computer-Aided Engineering, 2024, 31 (3), pp.285-305. 10.3233/ICA-240732 . cea-04573260

Neural Architecture Search for Radio Map Reconstruction with Partially Labeled Data

Aleksandra Malkova [a], Massih-Reza Amini [a*], Benoît Denis [b] and Christophe Villien [b]
[a] *LIG-APTIKAL, Université Grenoble Alpes, 700 Av. Centrale, 38401 Saint-Martin-d'Hères, France*
E-mails: aleksandra.malkova@univ-grenoble-alpes.fr, Massih-Reza.Amini@univ-grenoble-alpes.fr
[b] *CEA-Leti, 17 Av. des Martyrs, Grenoble, France*
E-mails: benoit.denis@cea.fr, christophe.villien@cea.fr

Abstract.
In this paper, we tackle the challenging task of reconstructing Received Signal Strength (RSS) maps by harnessing location-dependent radio measurements and augmenting them with supplementary data related to the local environment. This side information includes city plans, terrain elevations, and the locations of gateways. The quantity of available supplementary data varies, necessitating the utilization of Neural Architecture Search (NAS) to tailor the neural network architecture to the specific characteristics of each setting.

Our approach takes advantage of NAS's adaptability, allowing it to automatically explore and pinpoint the optimal neural network architecture for each unique scenario. This adaptability ensures that the model is finely tuned to extract the most relevant features from the input data, thereby maximizing its ability to accurately reconstruct RSS maps. We demonstrate the effectiveness of our approach using three distinct datasets, each corresponding to a major city. Notably, we observe significant enhancements in areas near the gateways, where fluctuations in the mean received signal power are typically more pronounced. This underscores the importance of NAS-driven architectures in capturing subtle spatial variations. We also illustrate how NAS efficiently identifies the architecture of a Neural Network using both labeled and unlabeled data for Radio Map reconstruction. Our findings emphasize the potential of NAS as a potent tool for improving the precision and applicability of RSS map reconstruction techniques in urban environments.

Keywords: Neural Networks with optimized architecture, Radio Map Reconstruction, Learning with partially labeled data

1. Introduction

Retrieving the exact position of the connected objects has become an important feature of the Internet of Things (`IoT`). Such connected objects have indeed been widespread over the last few years thanks to the low cost of the radio integrated chips and sensors and their possibility of being embedded in plurality of the devices.

By this they can help in fast development of large-scale physical monitoring and crowdsensing systems (like smart cities, factories, transportation, etc.). For the location-dependent application and services these abilities to associate accurate location with physical data gives huge opportunities [28]. For example, the fine-grain and dynamic update of air pollution and/or weather maps could benefit from geo-referenced mobile sensing[1] (e.g., aboard taxis, buses, bicycles...), thus continuously complementing the data from static stations. One of the localization techniques is Global Positioning System (`GPS`) which has been widely used over the past decades. More recently, low-cost advanced `GPS` solutions were proposed (like `RTK`, Bi-band,..), but still they suffer from high energy consumption which is not suitable for `IoT` applications.

As an alternative, one can opportunistically measure location-dependent radio metrics, like Received Signal Strength (Indicator) (`RSSI`), Time (Difference) of Arrival, Angle of Arrival, etc., because these sensor

---

[1] https://sarws.eu/

nodes communicate with one/several gateways at the same time (e.g. while sending a packet to the infrastructure to store this data in the cloud/server). Based on these metrics, there exist several methods to determine the node position: trilateration, triangulation, proximity detection, or fingerprinting [11, 14, 67]. We will focus on the last approach – fingerprinting [12]– which requires (ideally) full map of mentioned above radio metrics, covering the zone of interest. However, collecting metrics in each point of the zone of interest is impractical and time costly in real-world scenarios, therefore most approaches rely on sparse and non-uniformly distributed measurements.

In this sense, classical map interpolation techniques such as `RBF` [18, 47, 54] or kriging [36, 45] are used. Although these methods are relatively fast, they are quite weak in retrieving and predicting the complex and heterogeneous spatial patterns that are usually observed in real life signals (e.g., sudden and/or highly localized transient variations in the received radio metric due to specific environmental, local or topological effects). Another approach consists on deterministic simulation such as Ray-Tracing tools [9, 32, 51, 61]. Given some real field measurements and then calibrated over them, these models predict the radio propagation while simulating electromagnetic interactions with the environment. These technologies, however, need a complete description of the environment (properties of the materials of the obstacles, buildings, shape,etc.). Moreover, they are computationally complex, and in case of minor changes in the local area, these simulations should be re-run again. Recently, studies have employed machine learning for this task by considering radio maps as images and adapting neural network models that have been proposed for image completion. These models are based on the fully generated dataset by Ray-Tracing tools for predicting the signal propagation given the buildings mask and position of the transmitter [33]; or predicting the received power value for the Long Term Evolution (`LTE`) of the signal with use of additional information and neural networks [24, 43] with handcrafted structures.

In this work we will focus on the received signal strength map reconstruction, where only small amount of ground truth `GPS`-tagged measurements are available preventing to use existing `NN` models with handcrafted architectures and for which Ray-Tracing models could not be applied due to the lack of information about physical properties of the environment or due to high computational complexity. Our approach is based on *Neural Architecture Search* (`NAS`) [17] which aims to find an optimized `NN` model for this task. We show that by employing the latter technique, it is feasible to learn model parameters while simultaneously exploring the architecture. In addition, we employ unlabeled data in conjunction with ground-truth measurements in the training phase, as well as side information that accounts for the existence of buildings, to obtain knowledge and improve the model's performance. We assess our technique using three `RSSI` Map reconstruction collections, including one we produced for the city of Grenoble in France. In the case of the latter, we thoroughly examine its properties. In particular, we show that unlabeled data can effectively be used to find an efficient optimized `NN` model and that the side information provides valuable knowledge for learning. The obtained model is shown to have generalization ability on base stations that were not used in the training phase. The contribution of this paper is twofold:

- We propose a unified framework with the use of side information, for which we study the generalization ability of a neural network model which architecture is optimized over labeled and unlabeled data using side-information. This is an extension of the work of [40, 41].
- Furthermore, we provide empirical evaluation over three large-scale `RSSI` collections showing that the proposed approach is highly competitive compared to the state-of-the-art models in terms of quality metrics.

2. Related State of the Art

Classical techniques such as radial basis functions (`RBF`) or kriging [12] are simple and fast, but they are poor at predicting the complex and heterogeneous spatial patterns commonly observed in real-world radio signals (e.g., sudden and/or highly localized transient variations in the received signal due to specific environmental or topological effects, such as specific building shapes, presence of public furniture, ultra-navigation, etc.). Furthermore, data augmentation approaches for artificially increasing the number of measurements in radio map reconstruction issues have been developed.

The goal is to use the synthetic data created as extra data to train complex map interpolation models. However, these techniques need a highly thorough description of the physical environment and are unable to predict dynamic changes in the environment over time. A key bottleneck is their high computational complexity.

In the following, we go through some more relevant work on `RSSI` map reconstruction, including interpolation and data-augmentation techniques, as well as machine learning approaches.

2.1. Interpolation and data-augmentation techniques

Kriging or Gaussian process regression [34] is a prominent technique for radio map reconstruction in the wireless setting that takes into consideration the distance information between supplied measured locations while attempting to uncover their underlying 2D dependency.

Radial basis functions (`RBF`) [12, 18, 54] are another approach that simply considers the dependent on the distance between observed locations. As a result, this method is more adaptive and has been found to be more tolerant to some uncertainty [57]. Furthermore, in order to compare the performance of the `RBF` with different kernel functions for the map reconstruction of signal strength of Long Range (`LoRa`) radio waves, [12] divided all of the points in a database of outdoor `RSSI` measurements into training and testing subsets, with the linear kernel showing the best accuracy in both standard deviation and considered metric. The two approaches stated above (which depend on kernel techniques and underlying spatial relationships of the input measurements) need a lot of input data to provide reliable interpolation results, making them sensitive to sparse training sets. These methods have consequently been considered in pair with crowdsensing, where, for example in [36], to improve the performance of basic kriging, one calls for measuring the radio metric in new points/cells where the predicted value is still presumably imprecise. A quite similar crowdsensing method has also been applied in [20] after considering the problem as a matrix completion problem using singular value thresholding, where it is possible to ask for additional measurements in some specific cells where the algorithm has a low confidence in the predicted result. In our case though, we assume that we can just rely on a `RSSI` map with few ground-truth initial `RSSI` measurements.

Another approach considered in the context of indoor wireless localization (with map reconstruction firstly) relies on both measured field data and an a priori path loss model that accounts for the effect of walls presence and attenuation between the transmitter and the receiver [30] by using the wall matrix, which counts the number of walls along the path from the access point to the mobile location and penalize value according to that number. In some outdoor settings, training points are divided into a number of clusters of measured neighbors having specific `RSSI` distributions, and local route loss models are applied in an effort to capture localized wireless topology effects in each cluster [44]. However as parametric path loss models are usually quite imprecise, these techniques have a limited generalization capabilities and require additional impractical in-site (self-)calibration. A quite similar approach, except the use of additional side information, is followed in [37], 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 prior 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 stipulated in the context of fingerprinting based positioning consists in relying on deterministic simulation approach, namely Ray-Tracing tools (e.g., [32, 51, 52, 60, 61]). This technique 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 (typically, 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 are notorious for requiring high and likely prohibitive computational complexity in real applications. 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).

2.2. `NN` based models trained after data augmentation

There is an increasing interest in applying machine and deep learning methods to the problem of `RSSI` map reconstruction. These approaches have shown an ability to capture unseen spatial patterns of local effects and unseen correlations. Until now, to the best of our knowledge, these algorithms were primarily trained over simulated datasets generated by data-augmentation approaches (that were mentioned above).

In [33], given a urban environment, city geography, transmitter (`Tx`) location, and optionally pathloss measurements and car positions the authors introduce a `UNet`-based neural network called RadioUNet in the supervised learning setting, which outputs radio path loss estimates trained on a large set of generated using the Dominant Path Model data [56].

The authors of [23] propose a two-phase transfer learning with Generative Adversarial Networks (`GAN`) , which comprises two stages, to estimate the power spectrum maps in the underlay cognitive radio networks. The domain projecting (`DP`) framework is used to first project the source domain onto a neighboring domain. The target domain's entire map is then rebuilt or reconstructed using the domain completing framework and the recovered features from the surrounding domain.

For training of the `DP`, fully known signal distribution maps have been used. In another contribution, to improve the kriging predictions the authors have used the feedforward neural network for path loss modelling [58], as conventional parametric path loss models has a small number of parameters and do not necessarily consider shadowing besides mean power attenuation.

Apart from wireless applications, similar problems of map reconstruction also exist in other domains. In [70] for instance, the goal is to create the full topographic maps of mountains area given sparse measurements of the altitudes values. For this purpose, they use a `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 given full true map. In other work [66] the authors estimate the sea surface temperature with use of `GAN` architecture in the unsupervised settings but having a sequence of corrupted observations (with different cloud coverage) and known mask distribution. Another close more general problem making extensive use of neural networks is the image inpainting problem, where one needs to recover missing pixels in a single partial 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. It has been shown in [41] the interpretation of collected measurements on a map as an image with some pixels gives overall better result. But the problem is in consumed time as it does not give the generalized model and do not use the additional information about the local environment. Usually, such image inpainting problems can be solved by minimizing a loss between true and estimated pixels, where the former are artificially and uniformly removed from the initial full image. This is, however, not possible and not realistic in our case, as only a few ground-truth field measurements collected on the map can be used to reconstruct the entire image.

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 collected 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 [24, 26, 43], `RSS` values are predicted in exact points, given meta information such as the radio characteristics (e.g., transmission specifications or relationship between the receiver (`Rx`) and the transmitter (`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 [26]). 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).

2.3. 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.

The three main families of Semi-supervised Learning (SSL) approaches are Generative, Graph-based and Discriminant methods [10].

Generative Methods Semi-supervised learning with generative models involves estimating the conditional density using a maximum likelihood technique to estimate the parameters 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 that if two examples are in the same group, then they are likely to belong to the same class [59]. We can thus interpret semi-supervised learning with generative models (a) as a supervised classification where we have additional information on the probability density of the data, or (b) as a partition with additional information on the class labels of a subset of examples [5, 42]. If the hypothesis generating the data is known, generative models can become very powerful [69].

Graph-based Methods Graph-based methods employ an empirical graph built on the labeled and unlabeled examples to express their geometry [6]. They are based on the hypothesis that if two examples are in the same group, then they are likely to belong to the same class. The nodes of this graph represent the training examples and the edges translate the similarities between the examples. These similarities are usually given by a positive symmetric matrix, where the weight between two nodes is non-zero if and only if the corresponding examples are connected.

Methods for Semi-Supervised Learning leverage generative and graph-based approaches to harness the underlying data geometry through density estimation techniques or distance-based similarity matrix construction. However, these methods have their limitations when the underlying distributional assumptions become invalid or the distance used to estimate the similarity matrix fails to accurately represent the true data topology. In such cases, their performance tends to degrade in comparison to scenarios where only labeled examples are used for model training. [13].

Discriminant Methods To address these shortcomings, researchers have been motivated to devise strategies for mitigating these issues. One such approach involves using predictions from a discriminant model on unlabeled data to assign pseudo-labels. These pseudo-labeled examples are then integrated into the training process in a technique known as self-training [4, 21]. This iterative self-training process involves the repeated steps of pseudo-labeling and training a new model until there are no more unlabeled examples left. Some approaches have tackled the problem of modelling noise in the pseudo-labeling phase [29]. furthermore, 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 [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 Networks 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 deep learning, 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`) [16, 17, 63, 65]. 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. In the last few years the research on the topic of `NAS` has been shown a huge interest in the different fields. Among various studies, there are different techniques that are based on divers methods like Reinforcement Learning [71], Evolutionary Algorithm [53] or Bayesian Optimization [27]. Recently gradient-based methods became more popular. For example, one of the first methods based on this technique was presented in [39] and is called Differentiable Architecture Search (`DARTS`), which is using relaxation to, at the same time, optimize the structure of a *cell*, and the weight of the operations relative to each *cell*. After finding the best combinations, blocks are stacked manually to produce a neural network. Based on `DARTS`, more complex methods have appeared such as AutoDeepLab [38] 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 the fact that a complex representation leads to powerful architectures, this technique has some 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 done over a fixed network 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 possible technique is to use *Dynamic Routing* as proposed in [35]. 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* by weighting the paths during training procedure.

In the context of signal strength map reconstruction with neural networks, the choice between using a Generic Algorithm and Dynamic Routing to automatically find the architecture of the neural network depends on the specific problem, dataset, and constraints. Genetic Algorithms are well-suited for exploring a large and complex search space of neural network architectures [68]. They work by evolving a population of potential solutions over multiple generation, allowing them to efficiently search for a wide range of architectures [64]. This is important in signal strength map reconstruction where the optimal architecture may not be known in advance and can vary based on the nature of the signals, the environment, and other factors. Genetic Algorithms maintain diversity within the population of neural network architectures, ensuring that a wide range of designs is considered. This can be beneficial in finding novel and effective solutions that might be missed by more deterministic methods like Dynamic Routing as it has been shown in [41].

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 with partially labeled data. From this perspective and following [41], we consider the genetic algorithm for `NAS`.

3. 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 [40] 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; $j\in\{1,\dots,n\}$, 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

$$Y_m^j = Y^j \odot M^j, \tag{1}$$

with $M^j \in \{0,1\}^{H\times W}$ a binary mask of available measurements, and $\odot$ is the element-wise product of two matrices of the same dimensions, also called the Hadamard's 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 does 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

$$Y_\ell^j \oplus Y_v^j = Y_m^j, \tag{2}$$

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.

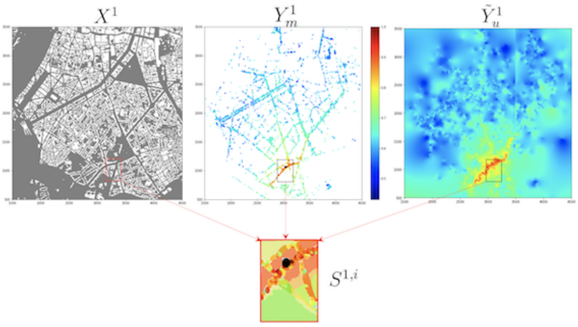


Fig. 1. An example of constituting the training sets for one base station. X^1 corresponds to 2D node locations, buildings are shown in white. Y_m^1 is `RSSI` map (true measurements); the base station is shown by a black circle,and $\tilde{Y}^1$ corresponds to interpolated points found by `RBF`. Colors 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 $Y_m^1 \cup \tilde{Y}^1$.

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 $(S^{j,i})_{\substack{1\leqslant j\leqslant n\\ 1\leqslant i\leqslant m_j}}$,i.e.

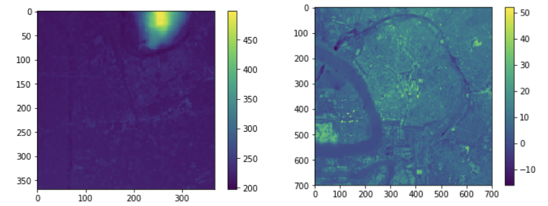
$$\forall j \in \{1,\dots,n\}, Y_m^j \oplus \tilde{Y}_u^j = \bigcup_{i=1}^{m_j} S^{j,i}, \tag{3}$$

where the sets $S^{j,i} \subseteq Y_m^j \oplus \tilde{Y}_u^j; \forall i \in \{1,\dots,m_j\}$ 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) $S_u^{j,i}$. To each submatrix $S^{j,i}$ corresponds a 2D location $X^{j,i} \subset X$. Fig. 1 gives a pictorial representation of the notations.

4. `NAS` with Genetic Algorithm for `RSSI` map reconstruction using side information

`UNet` [55] is one of the mostly used 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 in our experiments:

- information about the presence of buildings, which was taken from the open-source OpenStreetMap dataset [8] – matrix of binary 0-1 values, denoted as “buildings map” further (Fig. 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 localization 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` ([50]: the signal strength is proportional to $-10n\log_{10}(d)$ up to additive term, where n is a path loss exponent, d is a distance to base station) 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 Fig. 2. This information was taken from the open-source dataset[2] 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

Fig. 2. Elevation maps for two different cities

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

As stated in Section 2.4, 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 [53] for searching the most efficient architecture represented as a Direct Acyclic Graph[3]. Here, the validation sets

---

[2] https://doi.org/10.5069/G94M92HB

[3] https://colab.research.google.com/github/google-research/google-research/blob/master/evolution/regularized_evolution_algorithm/regularized_evolution.ipynb

$(S_v^{j,i})_{\substack{1\leqslant j\leqslant n\\ 1\leqslant i\leqslant m_j}}$ are put aside for hyperparameter tuning. The details for the Genetic algorithm are provided in Section 6.1. The edges of this acyclic graph 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 mean 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 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.

The entire learning process is outlined in Algorithm 1. We begin with a labeled training dataset and utilize the `RBF` interpolation method to estimate measurements for unlabeled training data. The true measurements from the labeled training set and the `RBF`-estimated measurements are subsequently partitioned into smaller matrices, which are used as inputs to the `NAS` module. Once the `NAS` module identifies the neural network model with optimized architecture, f_θ, we address two scenarios for learning its associated parameters θ by minimizing the following objective function:

$$\mathcal{L}(f_\theta, S_\ell \cup S_{\cancel{\ell}}^{j,i}) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{m_j}\sum_{i=1}^{m_j}\left[\frac{1}{|S_\ell^{i,j}|}\sum_{(x,y)\in S_\ell^{i,j}}\ell(y, f_\theta(x)) + \frac{1}{|S_{\cancel{\ell}}^{j,i}|}\sum_{(x,\tilde{y})\in S_{\cancel{\ell}}^{j,i}}\ell(\tilde{y}, f_\theta(x))\right]. \quad (4)$$

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

```
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 $f_\theta$ :
    $\theta_1^\star = \mathrm{argmin}_\theta\, \mathcal{L}(f_\theta, S_\ell \cup S_{\cancel{\ell}}^{j,i})$   # (Eq. 4);
  Scenario 2: Apply $f_{\theta_1^\star}$ on unlabeled data and obtain new pseudo-labeled measurements $S_{\cancel{\ell}}^{j,i}$ and find the new parameters $\theta_2^\star$ of the NN model $f_\theta$ :
    $\theta_2^\star = \underset{\theta}{\mathrm{argmin}}\, \mathcal{L}(f_\theta, S_\ell \cup S_{\cancel{\ell}}^{j,i})$
Output: $f_{\theta_1^\star}$ for scenario 1 or $f_{\theta_2^\star}$ for scenario 2.
```

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).

5. Evaluation Setup

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

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 [2]), consist of real GPS-tagged Long Range Wide Area Network (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 study.

5.1. Paris dataset

This first dataset is made of synthetic outdoor RSS measurements, which were simulated in a urban Long Term Evolution (LTE) cellular context with a ray-tracing propagation tool named VOLCANO (commercialized by SIRADEL). Those simulations were calibrated by means of side field measurements [9]. 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 2.1 regarding mostly computational complexity and prior information, we acknowledge a certain number of discrepancies or mismatches in comparison with 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] decibel-milliwatts (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. The overall scene is $1000m \times 1000m$, each pixel being $2m \times 2m$, thus forming a matrix of size 500×500. 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 Fig. 3.

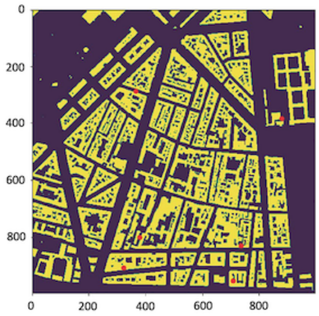


Fig. 3. Buildings map and corresponding Base Stations positions (in red) for Paris dataset. The x and y axes are in meters.

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 Fig. 4. Further details regarding the considered simulation settings can be found in [9].

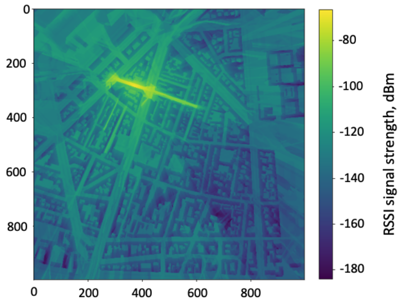


Fig. 4. Example of signal strength distribution (in dBm) generated through Ray-Tracing in the Paris dataset, with respect to one particular base station roughly located in (300m, 300m). The x and y axes are in meters.

5.2. Antwerp dataset

Measurement campaign and experimental settings The LoRaWAN dataset was collected in the urban are at the city centre of Antwerp from 17 November 2017 until 5 February 2018, [2], [1]. The dataset consists of 123,529 LoRaWAN messages with GPS coordinates on the map with RSSI measurements for 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 City of Things hardware. The latitude, longitude and Horizontal Dilution of Precision 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 from 30s to 5 min depending on the Spreading Factor used.

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 far from the collection zone having a flat signal. Finally we considered 9 base station – from Base-Station BS'_1 to Base-Station BS'_9 (see Fig. 5).

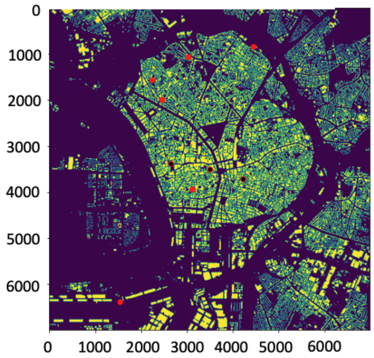


Fig. 5. Buildings map and corresponding considered 9 Base Stations positions (in red) for Antwerp dataset. The x and y axes are in meters.

The initial dataset with information about each Base-Station (BS) or gateway (`GW`), Receiving time of the message (RX time), Spreading Factor, Horizontal Dilution of Precision, Latitude, Longitude looks as following:

Dataset preprocessing and analysis. As an example, in this part we will explain the way the dataset was processed for the future application. We aggregated the received power into cells of the size 10 meters × 10 meters ($10m \times 10m$) and then averaged this power and translated into signal strength. To perform this aggregation, we measured the distance from the base station location based on local East, North, Up coordinates.

To compute the measurements density after data aggregation into cells of size $10m \times 10m$, we considered the close zone around the base station location of the size $3680m \times 3680m$. As an example, in the following Table 1 there is information about the first three considered base stations from this dataset.

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 1

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.

We consider the zone of full city of the size $7000m \times 7000m$ which covers the positions and most of the collected measurements in the city area, and we did the $10m \times 10m$ aggregation.

In the initial dataset if in the visited point on the map there was no captured signal, this point for corresponding base station was marked as -200 dBm, so in the Fig. 6 the informative range of the signal values lies in [-120; -60] dBm, where the left boundary correspond to the sensitivity of the device.

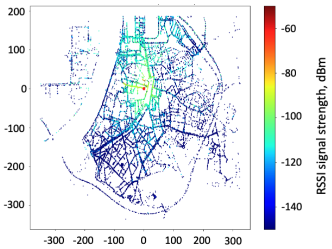


Fig. 6. RSSI values distribution, base station Antwerp, $10m \times 10m$ pixel resolution. The x and y axes are in meters.

5.3. Grenoble dataset

We have carried out an experimental campaign, and which is currently ongoing, aimed at gathering `LoRA` measurements within the actual urban setting of Grenoble city. This dataset holds significant value for future applications, primarily due to the scarcity of available data for research purposes in this domain. Furthermore, efforts are underway to expand this dataset further, which includes the installation of additional base stations and the collection of a larger volume of measurements, among other enhancements. In Appendix 8, we thoroughly present how these measurements were been collected.

Just like for the Antwerp 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 mean 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 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 Fig. 7.

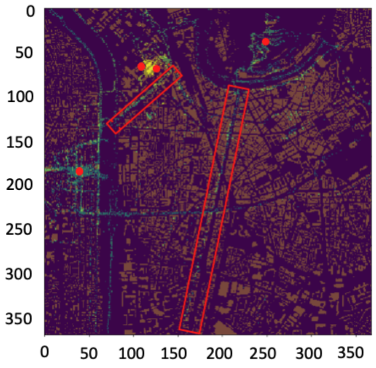
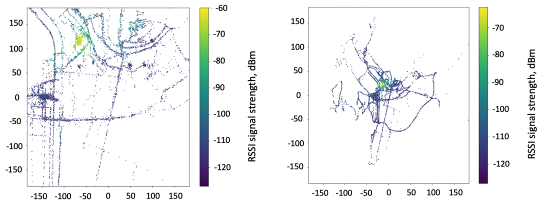


Fig. 7. Part of the Grenoble map, with a selection of the deployed base station positions (red circles) and two canonical streets in `LoS` (red rectangles. The x and y axes are in meters.

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 Fig. 8. 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 mean received power as a function of space but making available a larger amount of informative points for mapping.



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

Fig. 8. RSSI values distribution, base station BS_1 Grenoble, for different aggregation cell sizes. The x and y axes are in meters.

The amount of measurements for two base stations in the Grenoble-1 dataset for $10m \times 10m$ aggregation cells are shown in Table 2.

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

Table 2

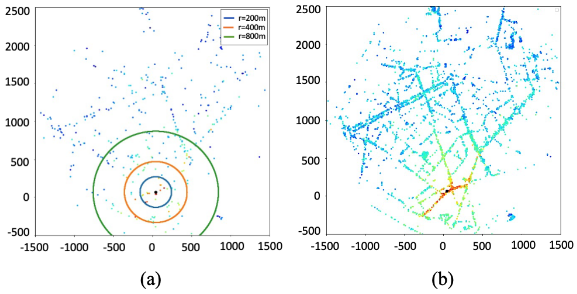
Amount of measurements for 2 base stations in the Grenoble-1 dataset (first version). Only the Base stations with the highest amount of points were selected.

Finally, 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 Grenoble. For that, as a reminder, we aggregated and averaged the power of collected measurements in cells/pixels of size 10 meters × 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 × 2 meters. As we also consider the generalization task, the algorithm should learn from all the available base stations data simultaneously.

In our settings, our available base station data is limited in scale, falling short by several orders of magnitude when compared to the datasets mentioned earlier. To address this inherent limitation, we devised a solution by artificially generating smaller submatrices from 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 ([15]). 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.



(a) (b)

Fig. 9. Illustration of (a) Test points across various circular zones and (b) Training points within the corresponding region of the Antwerp dataset. The location of the base station is denoted by a black point. Coordinates are in meters.

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 the Introduction) we compare the performance of the algorithms over different considered circles around the base station location, namely 200 meters, 400 meters and 800 meters radii (see Fig. 9).

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 4.

We considered state-of-the-art interpolation approaches which are: Total Variation (`TV`) in-painting by solving the optimization problem, Radial basis functions (`RBF`) [7] with linear kernel that was found the most efficient, and the k Nearest Neighbors (kNN) regression algorithm. The evolutionary algorithm in the model search phase was implemented using the `NAS-DIP` [25] package[4]. All experiments were run on NVIDIA GTX 1080 Ti 11GB GPU.

6. 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 4)?

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).

In the following, we will present our findings using the `UNet` model utilizing side information as a multi-channel input.

6.1. Generalization ability of `UNet`

We conducted an architectural optimization search for `UNet` using a Genetic Algorithm similar to one presented in [41] with the implementation of [25]. To configure the genetic algorithm, we defined a mutation rate of 5% to encourage diversity within the population and promote convergence. Moreover, we permitted the encoder-decoder networks to have asymmetric structures. The population size was determined to be 20 individuals, and the number of generations for the algorithm to iterate through was set at 30. The architecture is characterized by its "U" shape, where the encoder down-samples the input image to capture features, and the decoder upsamples to generate the final output. The number of layers in each component of the obtained U-Net architecture varied depending on the dataset used. More specifically:

- The encoder consists of multiple 2D-convolutional layers with Relu activation functions and having an increasing feature channels. The number of layers is determined based on experimentation and performance on the validation set. In our experiments, The convolutional layers found were 3 to 5 followed by a pooling or striding opera-

---

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

tion to reduce spatial dimensions. These configurations are fine-tuned after that the search of the architecture is completed.

- The bridge found, was a stack of 5 to 7 2D-convolutional layers with a smaller receptive field. It is usually thinner in terms of feature channels compared to the encoder and decoder which prevents overfitting.
- The decoder of the models contained also 3 to 5 layers, with transposed convolutions or upsampling layers to increase spatial dimensions.
- Skip connections were added between corresponding layers in the encoder and the decoder. There were as many skip connections as there were layers in the encoder and decoder.
- Finally, the output layer which is responsible for generating the output `RSSI` sub-metric is a single convolutional layer with a sigmoid activation function. We have found that the fine-tuning part at the end produces meaningful predictions specific to the dataset.

It is to be noted that the number of layers is determined based on experimentation and performance on the validation set. Furthermore, to mitigate overfitting; dropout and batch normalization regularization techniques are incorporated during the optimization process. These methods introduce randomness during training, preventing the model from relying too heavily on specific features of the training set. As an example, Table 3 outlines the neural network architecture layers discovered during the architecture search phase for the `RSSI` Map of the city of Grenoble.

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)

Table 3

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.

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 4

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)

We first study the learnability of the `UNet` model 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 used for in-painting [55]. The input was either only the measurements array or the measurements array stacked with the distances matrix.

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 Antwerp 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.

Fig. 10 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, as well as the overall error becomes smaller (BS''_6 - Table 4), 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 150m radius around the base station. However, when distance maps are added to the model's second channel, the situation is reversed.

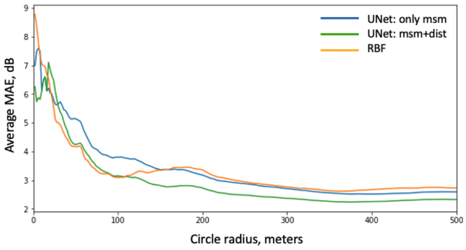


Fig. 10. MAE for the distance from the base station for the RBF and fixed UNet outputs over Paris dataset with validation set compliment to the train one, measurements and distances or only measurements input

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 Fig. 11. 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.

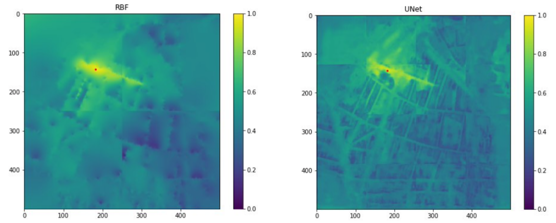
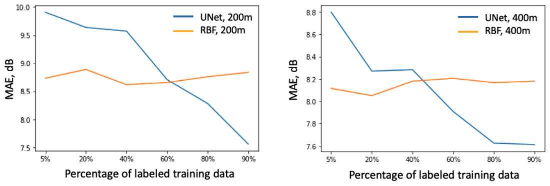


Fig. 11. Map reconstruction of RBF (left) and UNet (right) over the test base station BS_6'' in Paris shown by a red dot.

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, Fig. 12 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.



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

Fig. 12. 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.

6.2. The use of unlabeled data 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 1 (Section 4) and investigating the impact of side information on the performance of the optimized NN model discovered by NAS.

Fig. 13 illustrates the evolution of MAE within a variable-radius circular zone for $f_{\theta_1^\star}$ trained solely on measurements and measurements accompanied by distance maps (upper panel). In contrast, the lower panel depicts the performance of $f_{\theta_1^\star}$ trained on measurements, distance maps, elevation, and building count for the test base stations of Antwerp BS_9'. The integration of distance maps to complement measurements demonstrates enhanced prediction accuracy, consistent with our prior observations. When incorporating additional side information, such as building counts, the inclusion of elevation leads to superior signal estimations compared to scenarios where this information is omitted. This improvement can be attributed to the impact of building heights on signal transmission, highlighting the significance of considering elevation information in the training of the model.

As a best model obtained by Algorithm 1, scenario 1 we consider the case with three input channels: measurements, distances and elevations and present comparative results with other baselines in Table 5. The lowest errors are shown in boldface. The symbol $^\downarrow$ denotes that the error is significantly greater than the best

Model	`MAE`, dB, 200m	`MAE`, dB, 400m
`RBF`[7]	8.34↓	7.04↓
kNN	7.98↓	7.08↓
TV	7.50↓	6.97↓
`UNet` [55]	6.37↓	6.81↓
DIP [62]	6.55↓	6.63↓
$f_{\theta_1^\star}$	**5.88**	**6.37**

Model	`MAE`, dB, 200m	`MAE`, dB, 400m
`RBF` [7]	4.03↓	5.29↓
kNN	3.84↓	4.92↓
TV	4.53↓	5.91↓
DIP [62]	4.64↓	5.50↓
$f_{\theta_1^\star}$	**3.40**	**4.32**

Table 5

Comparisons between baselines in terms of `MAE` with respect to the two distances to Antwerp' test base station BS'_9 (left), and Grenoble test base station BS_4 (right). Best results are shown in bold.

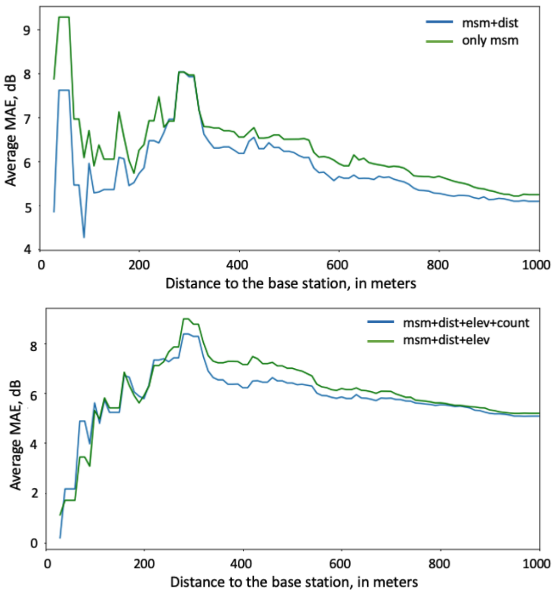


Fig. 13. Cumulative `MAE` distribution of $f_{\theta_1^\star}$ (scenario 1 of Algorithm 1) according to the distance to the test base station for the city of Antwerp BS'_9; with measurements and, measurements with distance maps (top) and measurements, distance maps, and elevation with and without building counts (down).

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

Fig. 14 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 1, 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 paper [41], these findings imply that self-training constitutes a promising future direction for `RSSI` map reconstruction.

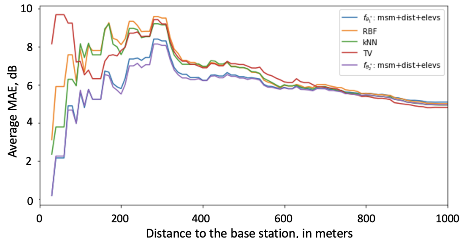
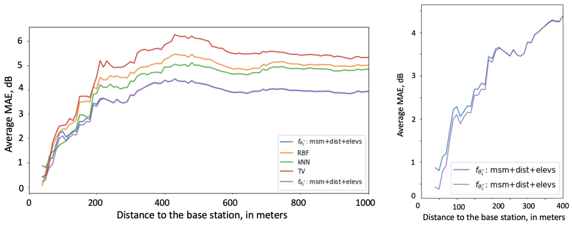


Fig. 14. Average `MAE` in dB of all models with respect to the distance to the test base Station BS'_9 for the city of Antwerp.

Fig. 15 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.



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

Fig. 15. Average `MAE` in dB of all models with respect to the distance to the test base Station BS_4 for the city of Grenoble.

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, $f_{\theta_2^\star}$, Fig. 16 depicts the errors heatmaps on circular zones of radius 200m and 400m surrounding the test base stations for Antwerp and Grenoble.

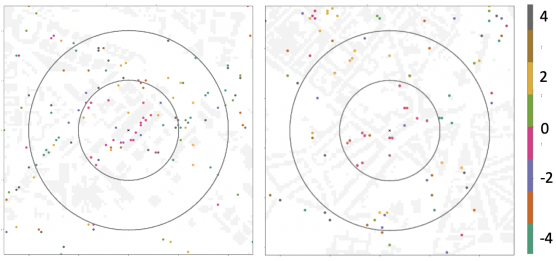
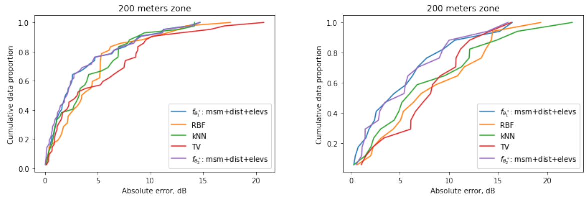


Fig. 16. Heatmap of the errors between true and predicted values of, $f_{\theta_2^\star}$, over test base stations in Grenoble (left) and Antwerp (right).

Each point reflects the difference between the real and predicted signal values. For both cities, 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 of different techniques in a 200-meter zone around the test base stations in Grenoble (Fig. 17, left) and Antwerp (Fig. 17 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.



(a) (b)

Fig. 17. Empirical cumulative distribution function of different techniques in a 200-meter zone around the test base stations in Grenoble (left) and Antwerp (right).

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 1 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 with respect to the distance to the base stations for different approaches are shown in Fig. 18.

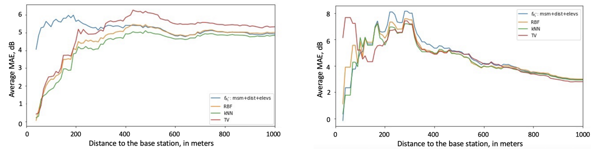


Fig. 18. `MAE` over the distance to the base station evaluated over unseen base stations for Grenoble (left) and Antwerp (right), $f_{\theta_1^\star}$ is trained over mixed dataset Grenoble+Antwerp.

Discussion According to these findings, the inclusion of signal data from another city and the resulting disruption in the search for an efficient `NN` model for Received Signal Strength Indicator (`RSSI`) measurements highlight an important consideration in the field of machine learning for signal strength mapping. This disruption is primarily due to the fact that the characteristics of radio signal propagation can vary significantly from one location to another. The environmental conditions, infrastructure, and geographical layout all influence the behavior of radio waves.

Indeed, different locations have unique characteristics that affect radio signal propagation. For example, urban environments with tall buildings may exhibit different signal behaviors compared to rural areas with open landscapes. The presence of obstacles, building materials, and even weather conditions can impact signal strength. As a result, `NN` models trained on data from one location may not generalize well to another. When building `NN` models for `RSSI` measurements, it is essential to focus on their ability to generalize to new, unseen data. The disruption caused by incorporating data from a different city suggests that models need to adapt to the specific characteristics of the location they are deployed in. This is often referred to as "model adaptation". It would be interesting to study over alignment strategies, such as those proposed for domain adaptation [31], in order to narrow the gap between these distributions in future work.

To address the challenges of location-specific characteristics, another approach is to develop an ensemble of models. Each model is trained on data from a specific location or under specific environmental conditions. The choice of which model to deploy can then be based on information about the incoming radio maps. For example, by using techniques like geolocation tagging to determine the source of the incoming data and selecting the corresponding model.

7. Conclusion and Future work

In this paper, we conducted an extensive investigation into the importance of incorporating additional side information for optimizing `NN` architectures in the context of map reconstruction across three diverse datasets. Our research underscores the pivotal role that auxiliary data, such as building distance and elevation, can play in substantially enhancing the performance of `NN` models with tailored architectures. Specifically, we found that by augmenting our training data with these contextual features, we achieved a significant reduction in the mean absolute error in dB, affirming the efficacy of our optimized architecture.

Our proposed approach demonstrates a notable advantage over agnostic techniques, particularly in proximity to the test base stations. The model's ability to accurately reconstruct signal strength maps, especially in the near vicinity of base stations, holds promising implications for practical applications such as urban planning and wireless network optimization. Importantly, our analysis reveals that our `NN`-based approach exhibits strong generalization capabilities, bolstering its utility in varying real-world scenarios.

However, it is essential to acknowledge that in situations where a distribution shift between two maps exists, the predictive confidence provided by the trained model may exhibit biases that render it less reliable. Substantial dissimilarities between two different Received Signal Strength Indicator (`RSSI`) maps can lead to a substantial degradation in the model's performance due to inaccuracies in pseudo-labels. While techniques like noise modelling may mitigate some of these issues in practice [22], it is vital to note that there remains a theoretical gap in our understanding. Specifically, exploring semi-supervised learning techniques under distribution shifts represents an important avenue for future research.

To address this challenge effectively, future work could focus on the development of Neural Dynamic Classification algorithms that adapt dynamically to distribution shifts [48]. These algorithms would enable the model to update its pseudo-labeling strategy based on evolving data distributions, enhancing its robustness in the face of changing environments.

Furthermore, the incorporation of Dynamic Ensemble Learning Algorithms could be explored to combine predictions from multiple models, each specialized for different distribution scenarios [46]. This ensemble approach may help mitigate the impact of distribution shifts by selecting the most appropriate model for the given context, further improving the reliability and accuracy of map reconstructions.

Additionally, self-supervised learning techniques offer a promising direction for enhancing the model's performance under distribution shifts [49]. By leveraging self-supervised learning, the model can learn useful representations from unlabeled data and adapt to new distribution patterns more effectively.

In summary, our findings underscore the potential of tailored `NN` architectures and the significance of auxiliary data in map reconstruction. However, addressing the challenges posed by distribution shifts remains an important direction for future research, with the potential for Neural Dynamic Classification algorithms, Dynamic Ensemble Learning Algorithms, and self-supervised learning to play a pivotal role in enhancing model reliability and accuracy in diverse and evolving scenarios.

References

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

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

[3] M.-R. Amini, V. Feofanov, L. Pauletto, L. Hadjadj, E. Devijver, and Y. Maximov. Self-training: A survey. *CoRR*, abs/2202.12040, 2022.

[4] M.-R. Amini, F. Laviolette, and N. Usunier. A transductive bound for the voted classifier with an application to semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS 21)*, pages 65–72, 2008.

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

[6] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56(1-3):209–239, 2004.

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

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

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

[10] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006.

[11] L. Cheng, C. Wu, Y. Zhang, H. Wu, M. Li, and C. Maple. A survey of localization in wireless sensor network. *International Journal of Distributed Sensor Networks*, 2012, 12 2012.

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

[13] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang. 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, 2004.

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

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

[16] X. Ding, W. Nie, X. Liu, X.Wang, and Q. Yuan. Compact convolutional neural network with multi-headed attention mechanism for seizure prediction. *Int. J. Neural Syst.*, 33(3):2350014:1–2350014:18, 2023.

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

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

[19] O. Esrafilian, R. Gangula, and D. Gesbert. Map reconstruction in uav networks via fusion of radio and depth measurements. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021.

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

[21] V. Feofanov, E. Devijver, and M.-R. Amini. Transductive bounds for the multi-class majority vote classifier. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3566–3573, 2019.

[22] L. Hadjadj, M.-R. Amini, and S. Louhichi. Generalization guarantees of self-training of halfspaces under label noise corruption. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3777–3785, 8 2023. Main Track.

[23] X. Han, L. Xue, Y. Xu, and Z. Liu. A Two-Phase Transfer Learning-Based Power Spectrum Maps Reconstruction Algorithm for Underlay Cognitive Radio Networks. *IEEE Access*, 8:81232–81245, 2020.

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

[25] K. Ho, A. Gilbert, H. Jin, and J. Collomosse. Neural architecture search for deep image prior. *Computers and Graphics*, 98, 2021.

[26] K. Inoue, K. Ichige, T. Nagao, and T. Hayashi. Radio propagation prediction using deep neural network and building occupancy estimation. *2020 International Symposium on Antennas and Propagation (ISAP)*, pages 315–316, 2021.

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

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

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

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

[31] A.k Kumar, P. Sattigeri, K. Wadhawan, L. Karlinsky, R. S. Feris, B. Freeman, and G. W. Wornell. Co-regularized alignment for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS 32)*, pages 9367–9378, 2018.

[32] M. Laaraiedh, B. Uguen, J. Stephan, Y. Corre, Y. Lostanlen, M. Raspopoulos, and S. Stavrou. 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, 2012.

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

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

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

[36] Jianxin Liao, Qi Qi, Haifeng Sun, and Jingyu Wang. Radio environment map construction by kriging algorithm based on mobile crowd sensing. *Wireless Communications and Mobile Computing*, 2019:1–12, 02 2019.

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

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

[39] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[40] A. Malkova, M.-R. Amini, B. Denis, and C. Villien. Radio map reconstruction with deep neural networks in a weakly labeled learning context with use of heterogeneous side information. In *4th International Conference on Advances in Signal Processing and Artificial Intelligence*, 2022.

[41] A. Malkova, L. Pauletto, C. Villien, B. Denis, and M.-R. Amini. Self-learning for received signal strength map recon-

struction with neural architecture search. In *31^{st} International Conference on Artificial Neural Networks*, pages 515–526, 2022.

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

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

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

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

[46] D. R. Pereira, M. A. Piteri, A. N. Souza, J.P. Papa, and H. Adeli. Fema: A finite element machine for fast learning. *Neural Comput. Appl.*, 32(10):6393–6404, 2020.

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

[48] M. H. Rafiei and H. Adeli. A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 28(12):3074–3083, 2017.

[49] M. H. Rafiei, L. V. Gauthier, H. Adeli, and D. Takabi. Self-supervised learning for electroencephalography. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2022.

[50] T.S. Rappaport. *Wireless communications: Principles and practice*. Prentice Hall, 2002.

[51] M. Raspopoulos and et al. Cross device fingerprint-based positioning using 3D Ray Tracing. In *Proc.*

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

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

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

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

[56] O. Ronneberger, P.Fischer, and T. Brox. 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, 2015.*

[57] V. Rusu and C. Rusu. Radial Basis Functions Versus Geostatistics in Spatial Interpolations. In International Federation for Information Processing Digital Library; Artificial Intelligence in Theory and Practice*, volume 217, 10 2006.*

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

[59] M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2001.

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

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

[62] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.*

[63] J. Urdiales, D. Martín, and J. María Armingol. An improved deep learning architecture for multi-object tracking systems. Integr. Comput. Aided Eng.*, 30(2):121–134, 2023.*

[64] Y. Xue, P. Jiang, F. Neri, and J. Liang. A multi-objective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks. Int. J. Neural Syst.*, 31(9):1–17, 2021.*

[65] Y. Xue, Y. Zhang, and F. Neri. A method based on evolutionary algorithms and channel attention mechanism to enhance cycle generative adversarial network performance for image translation. Int. J. Neural Syst.*, 33(5):2350026:1–2350026:15, 2023.*

[66] Y. Yin, A. Pajot, E. De Bézenac, and P. Gallinari. Unsupervised Inpainting for Occluded Sea Surface Temperature Sequences. In Proceedings of the 9th International Workshop on Climate Informatics (CI)*, Paris, France, DEC 2019.*

[67] K. Yu, I. Sharp, and Y. Guo. Ground-Based Wireless Positioning*. John Wiley & Sons, Ltd, 06 2009.*

[68] J. Zhang, L. Zhang, Y. Wang, J. Wang, X. Wei, and W. Liu. An efficient multi-objective evolutionary zero-shot neural architecture search framework for image classification. Int. J. Neural Syst.*, 33(5):2350016:1–2350016:15, 2023.*

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

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

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

8. Appendix: RSSI measurements in Grenoble

The Grenoble data were 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 similar to one described for Antwerp, but has a bit different structure, see Table 6.

The data was collected by different types of movements: walking of pedestrians, riding a bike or driving a car. Total amount of stored lines in the database collected 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 with

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 6

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.

personal tags. One example of stored data is shown in Table 6.

To collect the data, COTS telecom grade gateways iBTS from Kerlink manufacturer (see Fig. 19) 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 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 was used, as it is shown in Table 6 (the amount of stored metrics is bigger – like Signal-to-Noise Ratio, Time of Arrival, uplink network parameters : frequency, DataRate, etc. but here we will focus on the data used in our study).

During the data collection, there was different deployment characteristics of the base stations and different amount of them. We consider the second version of the dataset (Grenoble-2) consisted of four base stations located in Grenoble and one which was far from the city region and thus was discarded because of the flatten signal in the zone with the biggest amount of measurements (the first one could be found in [41]). Position of four considered base stations are shown in the Fig. 7. This deployment is also interesting as one of the base stations (BS_3) is installed on the mountain peak higher than all the other base stations and thus quite specific to this database.

Fig. 19. Gateway BS_3 which is installed on the roof near the cable car station at Bastille, Grenoble

Dataset preprocessing and analysis. 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 occurs 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 respect 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).

`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 Fig. 20 and Table 7. 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 mean `RSS` signal dynamic is such that the dispersion around the cell mean value (i.e., the mean 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 mean 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.

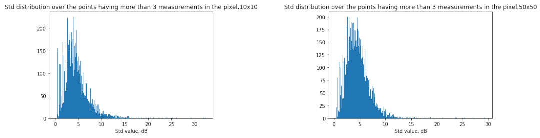
Granularity	Mean all, dB	Median all, dB	Std, dB
10x10	4.50	4.04	2.69
50x50	4.50	4.24	2.26

Table 7

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

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 Fig. 21). This indicator indeed gives a hint of 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 (Fig. 21b), 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.

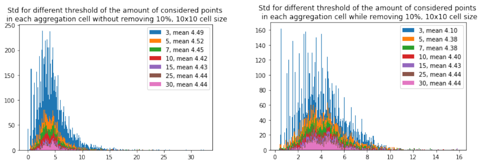
Accordingly, in terms of data preprocessing strategy, in the following (for further model parameters extraction or before applying our map interpolation algorithms), we will systematically reject 10% of the cells with the highest `RSS` variance divided by number of measurements, while keeping $10m \times 10m$ cells with at least 3 measurements.



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

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

Fig. 20. 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.

Fig. 21. Mean value over std with different threshold over amount of points in each cell, dB, for $10m \times 10m$ cell size

Extraction of path loss model parameters For any tag position $v = (x, y)$, we consider a set of independent mean 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. 5):

$$P_i^{\mathrm{dBm}}(v) = P_{0,i}^{\mathrm{dBm}} - 10n \log_{10} \frac{d_i(v)}{d_{0,i}} + w_i, \qquad (5)$$

where $d_i(v) = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ is the distance from a base station i of 2D Cartesian coordinates (x_i, y_i) to a tag of Cartesian coordinates $v = (x, y)$, $P_{0,i}^{dBm}$ is the free-space mean 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 $P_i^{\mathrm{dBm}}(v) = P_i$, $d_{0,i} = 1$ m and $P_{0,i}^{\mathrm{dBm}} = P_0$, $\forall i = 1..N$ and $d_i(v) = d_i$.

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. 5, 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 line-of-sight (`LoS`) and non-line-of-sight (`NLoS`) conditions [19] 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 5. For each base station BS, it hence comes:

$$\mathbf{m}^{BS} = \mathbf{P}(\eta^{BS}) + \epsilon^{BS}, \tag{6}$$

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:

$$(\mathbf{m}^{BS} - \mathbf{A}^{BS}\eta^{BS})^{\intercal}(\mathbf{m}^{BS} - \mathbf{A}^{BS}\eta^{BS}) \to \min_{\eta^{BS}} \tag{7}$$

So the set of optimal parameters is calculated as follows :

$$\hat{\eta}^{BS} = ((\mathbf{A}^{BS})^{\intercal}\mathbf{A}^{BS})^{-1}(\mathbf{A}^{BS})^{\intercal}\mathbf{m}^{BS} \tag{8}$$

Retrospectively and in first approximation, residuals can be interpreted as noise in Equation 5, so that the standard deviation parameter σ^{BS} is simply determined with the optimal model parameters and Equation 5:

$$\hat{\sigma}^{BS} = std(\mathbf{A}^{BS}\hat{\eta}^{BS} - \mathbf{m}^{BS}) \tag{9}$$

The results of this LS data fitting process for a few representative Base Stations of the Grenoble dataset are reported in Tables 8 and 9. 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 artifacts 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 mean `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 mean 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 over-weighting 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
Bastille (BS_2)	4.01	11.90	-44.74

Table 8

Extracted path loss model parameters for some of the Base stations from Grenoble dataset, as shown on Fig. 7, with data aggregation in $10m \times 10m$ cells.

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
Bastille (BS_2)	3.65	11.18	-50.49
$BS_1 \cup BS_2 \cup BS_4$	2.89	10.51	-51.88

Table 9

Recomputed path loss model parameters (after removing 10% of the data) for the same Base stations from Grenoble dataset, as shown on Fig. 7, with data aggregation in $10m \times 10m$ cells.

In Table 9, 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 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 (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 mean. 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 Fig. 22 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 mean values over the three considered base stations.

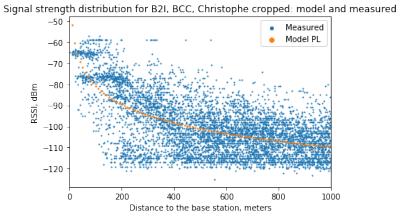
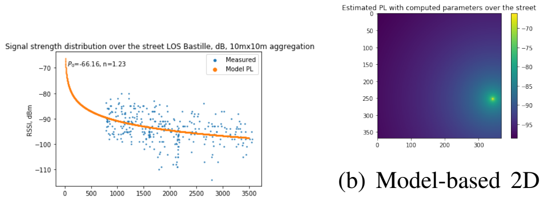


Fig. 22. `RSS` distribution as a function of transmission range, over 3 base stations: measured values (blue) vs. model-based prediction (orange) with extracted parameters P_0 = -51.88dBm, n=2.89, σ = 10.51 dB.

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, where it is theoretically easier to model the behavior 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 from Eq. 5, the parameters have been determined through data fitting and the resulting model is confronted to the measurements, as shown in Fig. 23. As we can see, it becomes evident that the log-distance classical path-loss model aligns reasonably well with the overall trend of the collected data. However, there remains a substantial degree of dispersion around the anticipated or predicted values. This dispersion serves as an indicator that conventional parametric model-based positioning (i.e., not based on fingerprinting) would pose challenges, even in the presence of`LoS` conditions.

As in this `LoS` case we can model the signal behavior, then it is possible to compare the gradients received by this model with the possible interpolations methods and then use it as one of the quality metrics in the map reconstruction problem.



(a) `RSS` measurements distribution and model-based 1D mean 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.

Fig. 23. Illustration of `RSS` distribution over one particular street: measured `RSS` values vs. model-based line-of-sight predictions.