

Techniques d’Auto-Apprentissage et Recherche Automatique d’Architecture Neuronale pour la Reconstruction de Cartes de Puissance Radio

Aleksandra Malkova^{1,2}, Loïc Pauletto^{1,3}, Christophe Villien², Benoît Denis², et Massih-Reza Amini¹

¹Université Grenoble Alpes, LIG-APTIKAL, F-38058 Saint Martin d’Hères

²CEA-Leti, Université Grenoble Alpes, F-38000 Grenoble

³Bull/Atos, F-38000 Grenoble

Résumé

Dans cet article, nous présentons un modèle de réseau de neurones (Neural Network, NN) s’appuyant sur une recherche d’architecture neuronale (Neural Architecture Search, NAS) et d’auto-apprentissage, que nous appliquons à la reconstruction instantanée de cartes de puissances radio (Received Signal Strength, RSS) à partir de mesures parcimonieuses, dans le cas où l’augmentation de données par des techniques de simulation déterministe n’est pas possible. Dans cette approche, nous déterminons tout d’abord une architecture NN optimale tout en entraînant conjointement le modèle déduit à partir d’une carte RSS représentant la vérité terrain. Cette carte de référence ainsi que les prédictions du modèle appliqué à un ensemble de points choisis aléatoirement sont ensuite utilisées pour entraîner un second modèle NN utilisant une architecture identique. Des résultats expérimentaux établis sur cinq cartes à grande échelle montrent que les prédictions de ce second modèle sont à la fois meilleures que les techniques d’interpolation de l’état de l’art ne s’appuyant pas sur de l’apprentissage, mais également que celles utilisant des modèles NN sans recherche d’architecture.

Mots-clé : Recherche d’architecture neuronale, Auto-apprentissage, Puissance reçue, Cartographie radio.

1 Introduction

Avec l’essor de l’Internet des Objets (IoT), l’arrivée à maturité de solutions intégrées à très basse consommation, tant en matière de capteurs que de communications sans fil à longue portée, permet aujourd’hui d’en-

visager le déploiement massif de dispositifs connectés à bas coût, au sein d’une pluralité d’environnements applicatifs (ex. villes, habitats et transports intelligents...). Au-delà, ces mêmes technologies favorisent plus spécifiquement la généralisation de systèmes de suivi à large échelle, géo-localisés et potentiellement participatifs (ex. pour la cartographie de mesures physiques, la capture de contexte, la surveillance ou le contrôle d’environnement...) [13].

Parmi les technologies de localisation disponibles, les systèmes de positionnement et de navigation par satellite (GNSS) ont été très majoritairement utilisés à l’extérieur des bâtiments ces vingt dernières années. Toutefois, ces solutions n’autorisent pas des niveaux de consommation énergétique compatibles avec les applications IoT envisagées ci-dessus. Dès lors, de manière à préserver à la fois une faible complexité embarquée au niveau des nœuds-capteurs sans fil et une précision de localisation acceptable, un compromis consiste à tirer profit des seuls liens radio bas débit à longue portée impliquant ces mêmes nœuds (e.g., LPWA ou NB-IoT). Plus spécifiquement, on cherche alors à interpréter les métriques radio disponibles, en tant que ”signatures” indicatrices de la position du dispositif mobile (*finger-printing*) [5, 6, 8, 32, 36]. A titre d’exemple, il est possible d’exploiter la puissance reçue (RSSI) des messages de données qui sont collectés au niveau de plusieurs stations de base. Les méthodes classiques de reconnaissance de signatures radio appliquées à la localisation sans fil [34] nécessitent idéalement la connaissance préalable d’une carte complète de ces métriques radio, couvrant toute la zone d’intérêt. Cependant, dans le cas de systèmes réels, il est difficilement envisageable de collecter des mesures pour chaque emplacement de

la carte, mais on doit plutôt se fier à des données "terrain" spatialement éparées et échantillonnées non uniformément. Des méthodes usuelles d'interpolation de cartes, telles que les méthodes à base de fonctions de base radiales (RBF) ou le *kriging* [7], ont déjà été envisagées dans un tel contexte. Ces approches ont l'avantage d'être rapides et présentent un faible niveau de complexité calculatoire. Toutefois, elles ne permettent pas véritablement de prédire assez finement les dynamiques complexes et hétérogènes généralement observés dans le cas de signaux radio réels (ex. présentant des variations soudaines et très localisées du niveau de signal reçu, dues à des effets environnementaux ou topologiques spécifiques). Des techniques d'augmentation des données ont alors été proposées afin d'accroître artificiellement le nombre de mesures disponibles pour la reconstruction de telles cartes radio. Typiquement, une fois calibrés à partir de quelques mesures réelles, les outils de prédiction déterministe de la propagation permettent de simuler les interactions électromagnétiques complexes entre ondes radio et environnement [16, 25, 31]. Les données synthétiques ainsi générées sont alors utilisées comme données supplémentaires pour entraîner des modèles complexes d'interpolation de cartes radio. Cependant, ces outils nécessitent une connaissance très détaillée de l'environnement physique et ne permettent pas facilement d'anticiper des changements d'environnement en fonction du temps. Leur complexité calculatoire constitue également un frein important.

Dans cet article, nous envisageons donc un problème de reconstruction de cartes radio dans un contexte IoT fortement contraint en termes de coût et de complexité IoT. Cette reconstruction s'appuie sur un unique jeu de données RSS réelles (*single snapshot*) anotées via GPS, constituant la "vérité terrain". Le recours aux techniques d'augmentation de données via simulations déterministes est ici proscrit par défaut, du fait des caractéristiques physiques inconnues de l'environnement. Ce type d'interpolation de cartes en mode *single snapshot* est assez similaire une problématique de restauration d'images, pour laquelle des modèles basés sur des NN avec des architectures neuronales fixes ont déjà été proposés [33]. Dans le cas où trop peu de pixels sont observés dans une même image, ces approches ne parviennent toutefois pas à en dégager la structure sous-jacente, lorsque celle-ci est trop complexe. Pour traiter cette question, nous proposons dans un premier temps une approche basée sur la recherche d'architecture automatique (NAS) de réseaux de neurones (NN), visant à déterminer le modèle le plus approprié, étant donné le nombre réduit de mesures "terrains" présentées en

entrée. À cet effet, nous développons deux stratégies de recherche distinctes, basées respectivement sur des algorithmes génétiques et sur un routage dynamique de l'information. Nous montrons notamment qu'avec cette dernière approche, il est possible d'apprendre les paramètres du modèle, tout en optimisant simultanément l'architecture. Dans une seconde étape visant l'apprentissage du modèle final, on utilise des données supplémentaires associées à des positions non visitées de la carte (c.-à-d., des *pseudo-labels* pour lesquels des données réelles n'ont pas été collectées), en exploitant des prédictions issues du premier NN optimisé lors de la phase de recherche d'architecture, venant ainsi compléter les données réelles disponibles. La solution en deux étapes proposée vise à trouver un compromis entre des techniques d'interpolation "agnostiques" et des approches d'apprentissage reposant sur l'apport de données "externes" (typiquement dans notre problème, des données issues de simulations déterministes de la propagation, jugées prohibitives ou indisponibles). Les résultats expérimentaux obtenus à partir de données réelles sur cinq cartes RSSI à grande échelle montrent que notre approche améliore les performances de reconstruction (en termes d'erreur moyenne et de dispersion) vis-à-vis des principales techniques d'interpolation de l'état de l'art, qu'elles soient sans apprentissage ou avec une architecture de NN fixe.

2 Etat de l'art

Dans cette section, nous présentons les travaux connexes sur la reconstruction des cartes RSSI, ainsi que les techniques existantes proposées pour NAS.

2.1 Techniques d'interpolation

Diverses méthodes d'interpolation spatiale ont été proposées pour la reconstruction de cartes radio dans le contexte du sans fil.

Une première approche, connue sous le nom de krigage ou régression par processus gaussien [18], exploite les informations de distance entre les points mesurés, tout en essayant de capturer leurs dépendances spatiales. Une autre méthode populaire est basée sur les fonctions de base radiales (*Radial Basis Functions*, i.e. RBF) [7, 9, 27]. Cette technique est en quelque sorte plus flexible, fait moins d'hypothèses concernant les données d'entrée (c'est-à-dire qu'elle ne tient compte que de la dépendance à la distance) et se révèle plus tolérante à l'égard de certaines incertitudes [29]. Dans le cas de [7] par exemple, les auteurs ont divisé tous les points

d'une base de données de mesures RSSI en extérieur en sous-ensembles d'apprentissage et de test, et ont comparé différentes fonctions de noyau pour l'interpolation. Les deux méthodes ci-dessus, qui reposent sur les propriétés statistiques sous-jacentes des données d'entrée (c'est-à-dire les corrélations spatiales) et les techniques à noyau, nécessitent une quantité importante de données d'entrée pour fournir des résultats d'interpolation précis. Par conséquent, elles sont particulièrement sensibles aux ensembles de données initiales éparses. Elles ont donc été envisagées en combinaison avec le crowdsensing. Dans [20] par exemple, afin d'améliorer les performances du krigeage de base, on demande de visiter de nouvelles positions/cellules où la valeur interpolée est toujours présumée inexacte. Une méthode de crowdsensing assez similaire a également été appliquée dans [10] après avoir énoncé le problème comme un problème de complétion de matrice utilisant le seuillage de valeur singulière. Dans notre cas, cependant, nous pouvons simplement nous fier à une carte RSSI avec peu de mesures initiales de la vérité du sol. Une autre

approche envisagée dans le contexte de la localisation sans fil à l'intérieur des bâtiments s'appuie à la fois sur les données recueillies sur le terrain et sur un modèle de perte de chemin a priori qui tient compte de l'effet de l'atténuation des murs entre l'émetteur et le récepteur [15]. Dans les environnements extérieurs, des modèles d'affaiblissement sur le trajet local (et donc des distributions RSSI particulières) ont été utilisés pour saisir les effets à petite échelle dans des groupes de points voisins mesurés au lieu d'utiliser les données RSSI brutes [23]. Toutefois, ces modèles paramétriques d'affaiblissement de parcours sont généralement assez imprécis et nécessitent un (auto-)étalonnage sur site peu pratique.

Approches basée sur l'augmentation des données. Une autre façon de construire ou de compléter des bases de données radio s'appuie sur des moyens de simulation déterministes, tels que le Ray-Tracing (par exemple, [16, 25, 31]).

Ces dernières visent à prédire la propagation radio sur site (c'est-à-dire à simuler les interactions électromagnétiques des ondes radio transmises dans un environnement). Une fois étalonnées à l'aide de quelques mesures réelles sur le terrain, ces données de simulation permettent d'alléger les efforts initiaux de métrologie et de déploiement (c'est-à-dire le nombre de mesures requises sur le terrain).

Néanmoins, ces outils nécessitent une description très détaillée de l'environnement physique (par exemple, la forme, les matériaux constitutifs et les

propriétés diélectriques des obstacles, des murs...). De plus, ils nécessitent généralement une complexité de calcul élevée.

Enfin, les simulations doivent être réexécutées, probablement à partir de zéro, chaque fois que des changements mineurs sont introduits dans l'environnement.

2.2 Modèles basés sur des RN entraînés après l'augmentation des données

Des approches d'apprentissage automatique et d'apprentissage profond ont été récemment appliquées pour la RSSI La reconstruction de cartes. Ces méthodes se sont révélées capables de récupérer des modèles spatiaux invisibles avec des effets topologiques hautement localisés et des corrélations cachées. Jusqu'à présent, ces méthodes ont été entraînées sur des ensembles de données simulés par des approches d'augmentation de données.

Dans [17], compte tenu d'un environnement urbain, les auteurs introduisent un réseau neuronal profond appelé RadioUNet, qui produit des estimations de l'affaiblissement du trajet radioélectrique entraînées sur un grand ensemble de données générées à l'aide du modèle de trajet dominant et de l'architecture UNet [28].

Dans une autre contribution, les auteurs ont montré que l'utilisation du réseau neuronal à anticipation pour la modélisation de l'affaiblissement du chemin pouvait améliorer les performances du krigeage [30], car les modèles paramétriques classiques d'affaiblissement du chemin admettent un petit nombre de paramètres et ne tiennent pas nécessairement compte de l'ombrage en plus de l'atténuation de la puissance moyenne.

Outre les applications sans fil, des problèmes similaires de restauration de cartes existent également dans d'autres domaines. Dans l'article [37] par exemple, les auteurs essaient de construire des cartes topographiques de zones montagneuses à partir de mesures éparses des altitudes. À cette fin, ils utilisent une architecture de réseau adversarial génératif (GAN), où, dans le discriminateur, ils comparent des paires de données d'entrée et la carte dite "reçue", soit générée par le générateur, soit basée sur la vraie carte complète.

Un autre problème proche faisant un usage intensif des réseaux neuronaux est le problème de la peinture d'images, où il faut restaurer les pixels manquants dans une seule image partielle. Par analogie, ce type de cadre pourrait également être appliqué dans notre contexte, en considérant la carte radio comme une image, où chaque pixel correspond au niveau RSSI pour un emplacement de nœud donné.

Habituellement, de tels problèmes de peinture d’images peuvent être résolus en minimisant la perte entre les pixels réels et prédits, les premiers étant artificiellement et uniformément supprimés de l’image initiale. Ceci est toutefois impossible dans notre cas, car seules quelques mesures de terrain véridiques peuvent être utilisées.

Contrairement aux approches précédentes, nous considérons des situations pratiques où les techniques d’augmentation des données ne peuvent pas être utilisées, principalement en raison des caractéristiques inconnues de l’environnement et des limitations informatiques, et où il n’y a qu’une petite quantité de mesures de vérité au sol. Notre approche recherche automatiquement un modèle de réseau neuronal optimisé pour la reconstruction de la carte RSSI en cours, et elle est basée sur l’auto-apprentissage pour l’apprentissage d’un modèle à base de RN amélioré avec les mesures initiales de vérité au sol et pseudo-étiquetées obtenues à partir des prédictions du premier modèle de RN sur un ensemble de points choisis aléatoirement dans la carte.

2.3 Méthodes associées à la recherche d’architecture automatique (NAS)

Dans la littérature, on trouve diverses techniques basées sur l’apprentissage par renforcement (RL) [38], l’algorithme évolutionnaire [26] ou l’optimisation bayésienne (BO) [12].

Récemment, de nouvelles méthodes basées sur le gradient sont devenues de plus en plus populaires. L’une des premières méthodes utilisant cette technique est appelée DARTS [22], dans laquelle une relaxation est utilisée pour optimiser simultanément la structure d’une *cellule*, et le poids des opérations par rapport à chaque *cellule*. À la fin, les cellules sont empilées manuellement pour former un réseau neuronal. Sur la base de DARTS, des méthodes plus complexes ont émergé comme *AutoDeepLab* [21] dans lequel un réseau est optimisé à 3 niveaux : (i) les paramètres des opérations, (ii) la structure des cellules et (iii) la macro-structure du réseau.

Malgré une représentation complexe conduisant à des architectures puissantes, cette technique présente certains inconvénients, comme le fait que l’architecture générée est à chemin unique, ce qui signifie qu’elle n’exploite pas pleinement les capacités de la représentation. De plus, comme la phase de recherche porte sur une architecture fixe, elle peut ne pas être la même d’une exécution à l’autre. Il est donc compliqué d’utiliser l’ap-

prentissage par transfert et l’impact d’une formation à partir de zéro peut être important.

Pour surmonter ces limitations, une technique consiste à utiliser le *Routage dynamique* (DR) tel que proposé dans [19]. Cette approche est différente des méthodes traditionnelles basées sur le gradient proposées pour NAS dans le sens où elle ne recherche pas une architecture fixe spécifique mais génère à la volée un chemin dynamique dans un maillage de cellules *sans rechercher*.

3 NAS pour la reconstruction de carte RSSI

Dans cette section, nous introduisons d’abord nos notations et notre cadre, puis nous présentons notre approche principale, désignée par SL_{NAS} dans la suite.

3.1 Notations et configuration

Pour chaque station base, $Y \in \mathbb{R}^{H \times W}$ est la matrice de l’ensemble des valeurs de RSSI constituant la vérité terrain, où $H \times W$ correspond aux dimensions de la zone d’intérêt (discrétisée). On suppose avoir accès uniquement à quelques mesures Y_m de Y telles que $Y_m = Y \odot M$, où $M \in \{0, 1\}^{H \times W}$ est un masque binaire indiquant si une mesure est disponible pour chaque pixel et \odot est le produit d’Hadamard. On suppose que le nombre d’éléments non-nuls de Y_m est bien inférieur à $H \times W$. On décompose le jeu de données Y_m en trois parties ; Y_ℓ (pour l’*apprentissage*), Y_v (pour la *validation*) et Y_t (pour le *test*), de telle sorte que $Y_\ell \oplus Y_v \oplus Y_t = Y_m$, où \oplus est l’opération d’addition matricielle. X_ℓ, X_v, X_t et X_m représentent les positions 2D correspondantes (ou de façon équivalente, les cellules/pixels correspondants) pour la station de base courante et X_u représente le jeu de positions 2D pour lesquelles aucune mesure n’est disponible.

Notre approche est basée sur trois phases principales : *i) phase de recherche automatique d’architecture* - Recherche d’une architecture optimisée pour le modèle de réseau de neurones sous-jacent ; *ii) phase d’augmentation de données* - Assignation de pseudo-labels pour un jeu de données non-labélisées choisies aléatoirement, en exploitant des prédictions issues du modèle de réseau NN entraîné à partir du jeu Y_ℓ ; et *iii) phase d’auto-apprentissage* - Apprentissage d’un second modèle de NN reposant sur la même architecture, à partir du jeu de mesures initial (vérité terrain) et des nouveaux exemples de pseudo-labels générés lors de la seconde phase.

Algorithm 1: SL_{NAS}

Entrées : Jeu de données pour l'apprentissage : (X_ℓ, Y_ℓ) ; Jeu de données pour la validation : (X_v, Y_v) ; Positions 2D dépourvues de mesures terrain : X_u ;

Initialisation : A partir de $(X_\ell, Y_\ell) \cup (X_v, Y_v)$, trouver les mesures interpolées \tilde{Y}_u pour les positions X_u en utilisant la méthode d'interpolation RBF;

Etape 1 : Rechercher une architecture de NN optimisée, à partir de $(X_\ell, Y_\ell) \cup (X_u, \tilde{Y}_u)$;

Etape 2 : Trouver les paramètres θ_1^* du modèle de NN f_θ :

$$\theta_1^* = \arg \min_{\theta} \mathcal{L}(X_\ell, Y_\ell, \theta) \quad \# \text{ (Eq. 1)};$$

Etape 3 : Choisir aléatoirement $X_u^{(k)}$ à partir de X_u et trouver les nouveaux paramètres θ_2^* du modèle de NN f_θ :

$$\theta_2^* = \arg \min_{\theta} \mathcal{L}(X_\ell \cup X_u^{(k)}, Y_\ell \cup f_{\theta_1^*}(X_u^{(k)}), \theta);$$

Sorties : $f_{\theta_2^*}, \tilde{Y}_u$.

Dans les sections suivantes, on détaille chacune des phases précédentes, qui sont aussi résumées au niveau de l'Algorithme 1.

3.2 Phase de recherche d'architecture

Nous considérons ici une première carte de référence RSSI comme une image d'entrée, où les mesures inconnues dans X_u sont obtenues avec un RBF en utilisant des points dans les ensembles de formation et de validation; $(X_\ell, Y_\ell) \cup (X_v, Y_v)$. Cette dernière s'est avérée être la plus efficace parmi les autres techniques d'interpolation de pointe [7]. Soit \tilde{Y}_u l'ensemble des mesures interpolées données par RBF sur X_u . Pour la phase de recherche de NAS, nous avons employé deux stratégies décrites ci-dessous.

3.2.1 Algorithme Génétique (GA)

À partir de l'ensemble $(X_\ell, Y_\ell) \cup (X_u, \tilde{Y}_u)$, nous utilisons un algorithme génétique similaire à [26] pour rechercher l'architecture la plus efficace représentée sous forme de graphe acyclique direct (DAG). Ici, l'ensemble de validation (X_v, Y_v) est mis de côté pour le réglage des hyperparamètres. Les bords de ce DAG représentent le flux de données avec une seule entrée pour chaque nœud, qui est une opération unique choisie parmi un ensemble d'opérations candidates. Nous considérons les opérations habituelles dans le domaine du traitement d'images, qui sont un mélange de

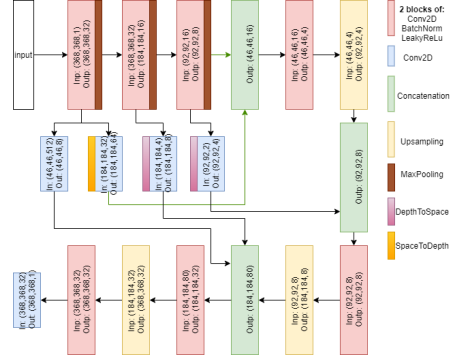


FIGURE 1 – Exemple d'architecture de réseau de neurones optimisée à l'issue d'une étape de recherche automatique, exploitable pour la reconstruction de cartes RSSI à partir de données expérimentales au sein de la ville de Grenoble.

couches de convolution et de mise en commun. Nous considérons également trois variantes de couches convolutionnelles 2D comme dans [33] avec des noyaux de taille 3, 5 et 7; et deux types de couches de mise en commun qui calculent soit la moyenne soit le maximum sur le filtre de taille 4. Les architectures candidates sont ensuite construites à partir d'opérations échantillonnées de manière aléatoire et les modèles RN correspondants sont entraînés.

Les 30 architectures obtenues sont ensuite classées selon un critère d'erreur absolue moyenne par pixel (MAE) entre le résultat interpolé du réseau et l'ensemble des mesures interpolées données par RBF \tilde{Y}_u . L'architecture la plus performante est finalement sélectionnée pour la mutation et placée dans la population entraînée. L'architecture la plus ancienne est supprimée afin de maintenir la taille de la population égale à 20 comme dans [26]. La figure 1 illustre une telle architecture optimisée avec 18 nœuds, qui a été trouvée pour la carte RSSI. Carte de la ville de Grenoble utilisée dans nos expériences (Section 4).

3.2.2 Routage Dynamique (DR)

Pour la phase de formation, nous employons la même structure et le même processus de routage que ceux proposés dans [19] (Figure 2). La structure est composée de 4 niveaux de sous-échantillonnage, où la taille de la carte des caractéristiques est divisée par 2 à chaque niveau, mais la profondeur de cette dernière est multipliée par 2 en utilisant une convolution 2D 1×1 .

Dans nos expériences, nous utilisons un réseau de 9 couches, qui correspondent à 33 cellules au total (en jaune sur la Fig.2). La structure contient également un

module "upsampling aggregation" à la fin (partie rouge sur la Fig.2). Le but de ce module est de combiner les cartes de caractéristiques de tous les niveaux et de reconstruire une carte de la taille de l'entrée. Contrairement à la méthode [19], ici, chaque cellule contient trois opérations *transforming* (c'est-à-dire une convolution 2D avec une taille de noyau de 3, 5 ou 7) afin d'avoir un bon point de comparaison avec la méthode décrite ci-dessus.

Cependant, en raison de la structure du réseau, nous avons décidé de ne pas utiliser les opérations de mise en commun, car cela aurait pu être potentiellement redondant. De plus, nous avons laissé la possibilité de créer des connexions résiduelles en ajoutant l'identité des opérations dans chaque cellule. De plus, nous n'avons pas utilisé les deux premières convolutions, utilisées à l'origine pour réduire la taille de l'entrée, afin de conserver le plus d'informations possible. Au lieu de cela, nous avons utilisé une convolution 2D 1×1 (en violet sur la Fig.2).

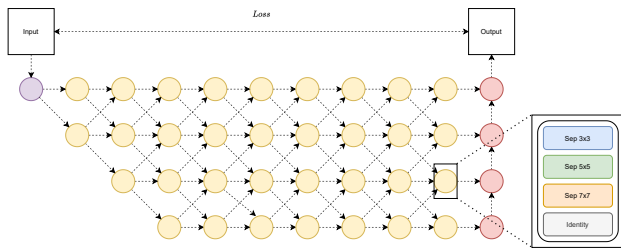


FIGURE 2 – Diagramme de l'architecture optimisée via routage dynamique à partir de nos données expérimentales. Les points violet, jaune et rouge représentent respectivement la convolution "stem", les cellules et le module "upsampling aggregation". Les flèches représentent le flux de données.

3.3 Phases d'augmentation de données et d'auto-apprentissage

Après la phase de recherche, le modèle de RN trouvé avec les paramètres θ , f_θ est entraîné sur (X_ℓ, Y_ℓ) en minimisant la perte suivante :

$$\mathcal{L}(X_\ell, Y_\ell, \theta) = \ell(f_\theta(X_\ell), Y_\ell) + \lambda \|\theta\|_2^2 + \mu \Omega(f_\theta(X_\ell)) \quad (1)$$

où $\ell(\cdot)$ est l'erreur absolue moyenne, et $\Omega(f_\theta(X_\ell))$ est la fonction de variation totale définie de la façon suivante :

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

Où $z_{i,j}$ la valeur de mesure d'un point de coordonnées i, j dans une certaine carte de distribution du signal Z . Cette fonction estime les variations d'amplitude locales des points dans Z qui sont minimisées afin de garantir que les points voisins auront des mesures prédites assez proches (c'est-à-dire, en préservant la continuité/l'homogénéité du signal).

Dans cette dernière, λ et μ sont des hyperparamètres pour respectivement les termes de régularisation et de variation totale et ils sont trouvés par validation croisée.

Avec le routage dynamique utilisé dans la phase de recherche, nous optimisons la structure du réseau et l'apprentissage des paramètres en minimisant (Eq. 1) en même temps. En se référant à l'Algorithme 1, les étapes 1 et 2 sont combinées dans ce cas.

Soit θ_1^* le jeu des paramètres du modèle NN obtenu en minimisant la fonction de perte (Eq. 1) à partir de la vérité terrain (X_ℓ, Y_ℓ) . Ce modèle optimisé est alors appliqué à un jeu de points $X_u^{(k)}$, choisis aléatoirement parmi X_u . Des pseudo-mesures de RSSI $\tilde{Y}_u^{(k)}$ sont ainsi prédites à partir du modèle optimisé $f_{\theta_1^*} : \tilde{Y}_u^{(k)} = f_{\theta_1^*}(X_u^{(k)})$.

Toujours sur la base de cette même architecture, un second modèle $f_{\theta_2^*}$ est enfin obtenu en minimisant une nouvelle fois la fonction de perte (Eq. 1), cette fois à partir du jeu de données augmenté, $(X_\ell, Y_\ell) \cup (X_u^{(k)}, \tilde{Y}_u^{(k)})$.

4 Expérimentations

Dans cette section, on décrit dans un premier temps le dispositif expérimental utilisé et par la suite, les résultats obtenus à partir des données expérimentales ainsi collectées.

Dispositif expérimental. Pour toutes nos expérimentations, on considère des cartes de 368×368 cellules/pixels. L'algorithme proposé est testé à partir de données terrain récoltées au sein de deux environnements urbains distincts, à savoir les villes de Grenoble (France) et d'Antwerp (Pays-Bas). Les mesures disponibles ont été agrégées et moyennées pour des cellules/pixels élémentaires de 10 mètres x 10 mètres. Pour la base de données d'Antwerp, qui est décrite plus en détails en [1], 3 stations de base ont été utilisées, BS'_1 , BS'_2 et BS'_3 , avec respectivement 5969, 6450 et 7118 mesures collectées (vérité terrain). Quant à la base de données de Grenoble, des mesures RSSI à partir de liens radio LPWA (LoRaWAN®), de la même façon qu'en [1], géo-localisées via GPS, ont été collectées vis-à-vis de 2 stations de base, BS_1 and BS_2 ,

avec respectivement 16577 et 7078 mesures collectées (vérité terrain). Pour réaliser l’agrégation de données par cellule, les distances ont été calculées à partir des coordonnées locales exprimées selon le standard *East, North, Up* (ENU). Au sein de chaque cellule, la puissance moyenne reçue a ensuite été calculée à partir de toutes les mesures disponibles agrégées (converties en valeurs de RSSI) puis normalisée entre 0 et 1, avant d’être présentée en entrée de notre algorithme.

Pour chaque station de base, 8% des pixels dotés de mesures ont été choisis pour l’apprentissage (X_ℓ, Y_ℓ), 2% pour la validation (X_v, Y_v) et les 90% restants pour le test (X_t, Y_t). Les données non-labellisées utilisées dans la **Etape 3** de l’Algorithme 1 ont été sélectionnées de manière aléatoire à partir d’un sous-ensemble de 4% des cellules restantes dénuées de mesures terrain. Les résultats sont évalués à partir des données de test, en utilisant l’indicateur d’erreur moyenne MAE, en dB. Cet indicateur d’erreur type est calculé (pour chaque carte), après remise à l’échelle des valeurs de puissance reçue normalisées (c.-à-d., après les avoir exprimées à nouveau selon leur dynamique initiale), puis moyenné sur 20 tirages aléatoires distincts, que ce soit pour les coordonnées des pseudo-labels ou pour toutes les autres coordonnées dotées de mesures terrain, qui sont exploitées pour l’apprentissage, la validation ou le test.

On compare les résultats d’interpolation de la méthode SL_{NAS} proposée avec les résultats de différentes approches de l’état de l’art, dont un algorithme de type RBF [4] utilisant des noyaux linéaires (évalués empiriquement comme les plus performants dans notre problème), un algorithme de *kriging* (KRIG) [24], et la méthode Navier-Stocks (NS) [3]. Pour cette dernière, on considère les deux méthodes de recherche d’architecture déjà présentées (Section 3.2), à savoir un algorithme génétique (GA) d’une part ($\text{SL}_{\text{NAS-GA}}$) et le routage dynamique (DR) d’autre part ($\text{SL}_{\text{NAS-DR}}$). Pour l’approche $\text{SL}_{\text{NAS-GA}}$, on évalue aussi l’apport de la phase d’auto-apprentissage (**Etape 3**) ($\text{SL}_{\text{NAS-GA}}(f_{\theta_2})$) en la comparant avec le modèle de réseau NN issu de l’**Etape 1** ($\text{SL}_{\text{NAS-GA}}(f_{\theta_1})$). L’algorithme évolutionniste de la phase de recherche d’architecture (Section 3.2) a été implémenté en utilisant le *package* NAS-DIP¹ [11]. Ce dernier a été initialement développé pour une méthode de type *Deep Image Prior* (DIP) [33] reposant sur un modèle de NN adapté à la reconstruction d’image. En considérant les cartes RSSI comme des images corrompues, avec seulement une partie de leurs pixels observables (c.-à-d., les mesures constituant la vérité terrain), un modèle de NN disposant de la même archi-

tecture que celle introduite en [33] a également été entraîné, pour comparaison (DIP). Tous les résultats expérimentaux ont été obtenus sur un GPU de type Tesla V100.

	Grenoble		Antwerp		
	BS_1	BS_2	BS'_1	BS'_2	BS'_3
RBF [4]	5.03 [↓]	3.16 [↓]	3.58 [↓]	3.35	3.90
KRIG [24]	5.68 [↓]	4.21 [↓]	3.69 [↓]	4.39 [↓]	4.91 [↓]
NS [3]	5.11 [↓]	3.14 [↓]	4.28 [↓]	3.45	3.87
TV	5.13 [↓]	2.89	3.76	3.51	3.83
DIP [33]	5.14 [↓]	3.22 [↓]	3.53	3.41	3.92
$\text{SL}_{\text{NAS-DR}}$	4.82	2.82	3.48	3.42	3.81
$\text{SL}_{\text{NAS-GA}}(f_{\theta_1})$	4.79	2.81	3.39	3.27	3.75
$\text{SL}_{\text{NAS-GA}}(f_{\theta_2})$	4.76	2.79	3.33	3.27	3.74

TABLE 1 – Valeurs moyennes d’erreur MAE, en dB, obtenues à partir de différentes approches d’interpolation de cartes RSSI, pour chacune des stations de base disponibles au sein des deux environnements urbains testés.

Résultats Expérimentaux. La Table 1 résume les résultats obtenus pour les 5 cartes RSSI considérées (c.-à-d., pour l’ensemble des stations de base disponibles au sein des deux environnements urbains testés), où les erreurs les plus faibles sont reportées en gras. Le symbole [↓] indique que l’erreur est considérée comme ”significativement plus grande” que le meilleur résultat obtenu, selon le test statistique dit ”des rangs signés” de Wilcoxon, avec une valeur de seuil *p*-value de 0.01 [35]. Dans tous les cas testés, $\text{SL}_{\text{NAS-GA}}$ et $\text{SL}_{\text{NAS-DR}}$ obtiennent alors de meilleurs résultats que l’ensemble des autres méthodes de l’état-de-l’art, même en l’absence de phases d’augmentation de données et/ou d’auto-apprentissage ($\text{SL}_{\text{NAS-GA}}(f_{\theta_1})$). On note alors que l’approche DIP, qui repose sur un modèle de NN à architecture fixe, obtient des résultats sensiblement équivalents à l’approche agnostique RBF. Ces résultats tendent à montrer que la recherche d’une architecture et d’un modèle NN optimisés s’avère efficace pour la reconstruction de cartes RSSI, lorsqu’on évolue dans des contextes applicatifs IoT fortement contraints.

La Figure 3 représente l’erreur moyenne MAE en dB en fonction de la distance vis-à-vis de la station de base BS_1 dans la ville de Grenoble, où $\text{SL}_{\text{NAS-GA}}(f_{\theta_2})$ semble fournir de meilleures prédictions principalement à moyenne et longue distances, typiquement au-delà de 250m, où la dynamique du signal semble plus régulière et le niveau RSSI plus faible. Ces résultats, relativement contre-intuitifs, suggèrent de mener de nouvelles investigations quant au modèle de prédiction, en fonction de la dynamique utile du signal RSSI et en gardant

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

à l'esprit l'application finale de *fingerprinting*. Pour les régions à proximité de la station de base où la dynamique du signal RSSI est censée être plus importante, mais aussi plus erratique, on s'attend à ce qu'une forme de connaissance a priori sur l'environnement physique (ex., utilisation de photos aériennes ou du plan de la ville, rendant compte de la présence et/ou la hauteur des bâtiments...) puisse être utile au processus d'apprentissage (typiquement, via l'utilisation d'un canal d'information adjacent, en plus des mesures radio déjà présentées en entrée de l'algorithme), afin de mieux capter des effets topologiques locaux sur la propagation radio notamment.

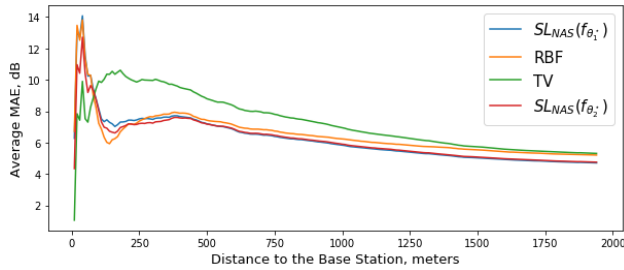


FIGURE 3 – Erreur moyenne d'interpolation MAE, en dB, en fonction de la distance vis-à-vis de la station de base BS_1 .

Les boîtes à moustache de la Figure 4 rendent compte de la distribution de l'erreur moyenne d'interpolation MAE, en dB (distribution sur l'ensemble des tirages aléatoires de partitionnement des données d'entrée) avec les méthodes DIP, RBF et $SL_{NAS-GA}(f_{\theta_2}^*)$, pour la seule station de base BS_1 et pour différents pourcentages de données non-labellisées ($\{4, 7, 10, 14\}\%$) injectées lors de la phase d'auto-apprentissage (Section 3.3). On remarque alors qu'en augmentant le nombre d'exemples non-labellisés, la variance de l'erreur commise avec l'algorithme $SL_{NAS-GA}(f_{\theta_2}^*)$ augmente de manière significative, principalement en raison de l'augmentation relative du nombre de valeurs fortement "bruitées" -et donc moins fiables- parmi les prédictions issues de $f_{\theta_1}^*$. Cela revient à un apprentissage avec superviseur imparfait, déjà étudié dans le cadre de l'apprentissage semi-supervisé [2, 14]. En guise de perspectives, on projète donc d'incorporer un modèle probabiliste d'erreur de labellisation dans l'**Etape 3** de l'Algorithme 1, mais aussi de procéder à l'apprentissage conjoint des paramètres de ce dernier modèle d'erreur et des paramètres du réseau NN.

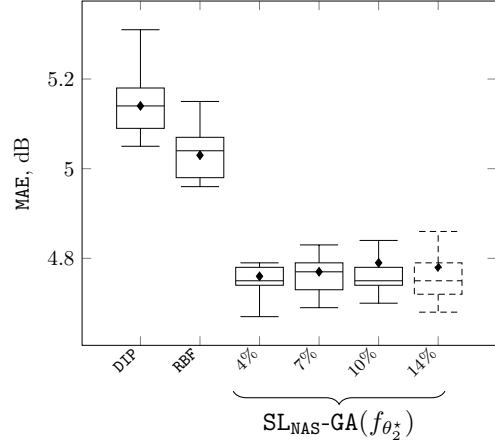


FIGURE 4 – Boîtes à moustache représentant la distribution de l'erreur moyenne MAE, en dB (distribution sur l'ensemble des tirages aléatoires de partitionnement des données d'entrée) avec les méthodes d'interpolation DIP, RBF et $SL_{NAS-GA}(f_{\theta_2}^*)$, pour la seule station de base BS_1 et différents pourcentages de données non-labellisées ($\{4, 7, 10, 14\}\%$) injectées lors de la phase d'auto-apprentissage.

5 Conclusion

Dans cet article, nous avons présenté un modèle de réseau de neurones s'appuyant sur une recherche d'architecture et de l'auto-apprentissage pour la reconstruction instantanée de cartes RSS à partir de mesures parcimonieuses, dans un contexte où l'augmentation de données par simulation déterministe n'est pas applicable. Nous avons présenté deux versions de NAS basées sur des algorithmes génétiques et du routage dynamique. Les résultats expérimentaux obtenus pour des cartes à grandes échelles montrent que notre approche surpasse les techniques d'interpolation de l'état de l'art n'utilisant pas d'apprentissage mais aussi celles utilisant des NN dont l'architecture est définie manuellement.

Références

- [1] Aernouts, M., Berkvens, R., Vlaenderen, K.V., Weyn, M. : Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas. *Data* **3**(2) (2018)
- [2] Amini, M.R., Usunier, N., Laviolette, F. : A transductive bound for the voted classifier with an application to semi-supervised learning. In : Ad-

- vances in Neural Information Processing Systems. pp. 65–72 (2009)
- [3] Bertalmio, M., Bertozzi, A.L., Sapiro, G. : Navier-stokes, fluid dynamics, and image and video inpainting. In : CVPR (2001)
- [4] Bishop, C.M. : Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg (2006)
- [5] Burghal, D., Ravi, A.T., Rao, V., Alghafis, A.A., Molisch, A.F. : A comprehensive survey of machine learning based localization with wireless signals (2020)
- [6] Cheng, L., Wu, C., Zhang, Y., Wu, H., Li, M., Maple, C. : A survey of localization in wireless sensor network. *International Journal of Distributed Sensor Networks* **2012** (12 2012)
- [7] Choi, W., Chang, Y.S., Jung, Y., Song, J. : Low-power lora signal-based outdoor positioning using fingerprint algorithm. *ISPRS International Journal of Geo-Information* **7**(11) (2018)
- [8] Dargie, W., Poellabauer, C. : Fundamentals of Wireless Sensor Networks : Theory and Practice. John Wiley & Sons (2010)
- [9] Enrico, A., Redondi, C. : Radio Map Interpolation Using Graph Signal Processing. *IEEE Communications Letters* **22**(1), 153–156 (2018)
- [10] Fan, X., He, X., Xiang, C., Puthal, D., Gong, L., Nanda, P., Fang, G. : Towards system implementation and data analysis for crowdsensing based outdoor RSS maps. *IEEE Access* **6**, 47535–47545 (2018)
- [11] Ho, K., Gilbert, A., Jin, H., Collomosse, J. : Neural architecture search for deep image prior (2020)
- [12] Jin, H., Song, Q., Hu, X. : Auto-keras : An efficient neural architecture search system. In : Proceedings of the 25th ACM SIGKDD. pp. 1946–1956 (2019)
- [13] Khelifi, F., Bradai, A., Benslimane, A., Rawat, P., Atri, M. : A Survey of Localization Systems in Internet of Things. *Mobile Networks and Applications* **24**(3), 761–785 (Jun 2019)
- [14] Krithara, A., Amini, M.R., Renders, J.M., Goutte, C. : Semi-supervised document classification with a mislabeling error model. In : 30th European Conference on Information Retrieval. pp. 370–381. Glasgow (2008)
- [15] Kubota, R., Tagashira, S., Arakawa, Y., Kitasuka, T., Fukuda, A. : Efficient survey database construction using location fingerprinting interpolation. In : 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). pp. 469–476 (2013)
- [16] Laaraiedh, M., Uguen, B., Stephan, J., Corre, Y., Lostanlen, Y., Raspopoulos, M., Stavrou, S. : 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). pp. 276–280 (2012)
- [17] Levie, R., Yapar, , Kutyniok, G., Caire, G. : Path-loss prediction using deep learning with applications to cellular optimization and efficient d2d link scheduling. In : ICASSP. pp. 8678–8682 (2020)
- [18] Li, J., D.Heap, A. : A review of comparative studies of spatial interpolation methods in environmental sciences : Performance and impact factors. *Ecological Informatics* **6**(3), 228 – 241 (2011)
- [19] Li, Y., Song, L., Chen, Y., Li, Z., Zhang, X., Wang, X., Sun, J. : Learning dynamic routing for semantic segmentation (2020)
- [20] Liao, J., Qi, Q., Sun, H., Wang, J. : Radio environment map construction by kriging algorithm based on mobile crowd sensing. *Wireless Communications and Mobile Computing* **2019**, 1–12 (02 2019)
- [21] Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A.L., Fei-Fei, L. : Auto-deeplab : Hierarchical neural architecture search for semantic image segmentation. In : Proceedings of CVPR. pp. 82–92 (2019)
- [22] Liu, H., Simonyan, K., Yang, Y. : Darts : Differentiable architecture search. arXiv preprint arXiv :1806.09055 (2018)
- [23] Ning, C., et al. : Outdoor location estimation using received signal strength-based fingerprinting. *Wireless Pers Commun* **99**, 365–384 (2016)
- [24] Oliver, M., Webster, R. : Kriging : a method of interpolation for geographical information systems. *International Journal of Geographical Information System* **4**(3), 313–332 (1990)
- [25] Raspopoulos, M., Laoudias, C., Kanaris, L., Kokkinis, A., Panayiotou, C.G., Stavrou, S. : 3d ray tracing for device-independent fingerprint-based positioning in w lans. In : 2012 9th Workshop on Positioning, Navigation and Communication. pp. 109–113 (2012)

- [26] Real, E., Aggarwal, A., Huang, Y., Le, Q.V. : Regularized evolution for image classifier architecture search. In : AAAI. vol. 33, pp. 4780–4789 (2019)
- [27] Redondi, A.E.C. : Radio map interpolation using graph signal processing. *IEEE Communications Letters* **22**(1), 153–156 (2018)
- [28] Ronneberger, O., P.Fischer, Brox, T. : U-net : Convolutional networks for biomedical image segmentation. In : *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. LNCS, vol. 9351, pp. 234–241. Springer (2015)
- [29] Rusu, V., Rusu, C. : *Radial Basis Functions Versus Geostatistics in Spatial Interpolations*. vol. 217 (10 2006)
- [30] Sato, K., Inage, K., Fujii, T. : On the performance of neural network residual kriging in radio environment mapping. *IEEE Access* **7**, 94557–94568 (2019)
- [31] Sorour, S., Lostanlen, Y., Valaee, S., Majeed, K. : Joint indoor localization and radio map construction with limited deployment load. *IEEE Transactions on Mobile Computing* **14**(5), 1031–1043 (2015)
- [32] Tahat, A., Kaddoum, G., Yousefi, S., Valaee, S., Gagnon, F. : A look at the recent wireless positioning techniques with a focus on algorithms for moving receivers. *IEEE Access* **4**, 6652–6680 (2016)
- [33] Ulyanov, D., Vedaldi, A., Lempitsky, V. : Deep image prior. *CoRR* **abs/1711.10925** (2017)
- [34] Vo, Q.D., De, P. : A Survey of Fingerprint-Based Outdoor Localization. *IEEE Communications Surveys & Tutorials* **18**(1), 491–506 (2016)
- [35] Wilcoxon, F. : Individual comparisons by ranking methods. *Biometrics* **1**(6), 80–83 (1945)
- [36] Yu, K., Sharp, I., Guo, Y. : *Ground-Based Wireless Positioning*. John Wiley & Sons, Ltd (06 2009)
- [37] Zhu, D., Cheng, X., Zhang, F., Yao, X., Gao, Y., Liu, Y. : Spatial interpolation using conditional generative adversarial neural networks. *International Journal of Geographical Information Science* **34**(4), 735–758 (2020)
- [38] Zoph, B., Le, Q.V. : Neural architecture search with reinforcement learning. *arXiv preprint arXiv :1611.01578* (2016)