Transfer Learning for Tilt-Dependent Radio Map Prediction

Claudia Parera, Qi Liao, Ilaria Malanchini, Cristian Tatino,
Alessandro E. C. Redondi, and Matteo Cesana

***Abstract*—Machine learning will play a major role in handling the complexity of future mobile wireless networks by improving network management and orchestration capabilities. Due to the large number of parameters that can be monitored and configured in the network, collecting and processing high volumes of data is often unfeasible or too expensive at network runtime, which calls for taking resource management and service orchestration decisions with only a partial view of the network status. Motivated by this fact, this paper proposes a transfer learning framework for reconstructing the radio map corresponding to a target antenna tilt configuration by transferring the knowledge acquired from another tilt configuration of the same antenna, when no or very limited measurements are available from the target. The performance of the framework is validated against standard machine learning techniques on a data set collected from a 4G commercial base stations. In most of the tested scenarios, the proposed framework achieves notable prediction accuracy with respect to classical machine learning approaches, with a mean absolute percentage error below 8%.**

***Index Terms*—Radio map prediction, antenna tilt, transfer learning.**

I. Introduction

Fifth generation wireless networks (5G) are expected to improve the performance of cellular systems, achieving higher data rates, reduced latency, higher reliability and support for greater numbers of users. To achieve this, 5G resorts to dense and heterogeneous deployments, coupled with higher flexibility in the network access and core domains, which can be dynamically managed in either a centralized or distributed manner. To cope with such a complex scenario, it is foreseen that machine learning tools will play a major role in enabling the transition from current mobile networks to future 5G architectures [2]. By exploiting the increased availability of data in 5G coming from network devices and user terminals, machine learning tools will be able to assist network operators in dealing with the increasing complexity of configuring parameters for network optimization. Thus, machine learning tools will form the basis for automated and smart network management techniques.

Among the manifold parameters that can be configured at the base station (BS), one of the most important is the antenna tilt, which is the angle formed by the vertical direction which the antenna is facing and the horizon. Antenna tilt can be controlled either mechanically (by physically tilting the antenna up or down) or electronically (relying on beam-forming techniques that steer the main beam of the antenna towards a desired vertical direction), or by a combination of the two. The antenna tilt directly impacts the performance of the cell served by the BS in terms of network coverage, signal strength and inter-cell interference, and therefore determines the quality of service experienced by end users. In particular, when the antenna tilt is changed in the BS, its effect on the antenna gain over distance also changes, which further leads to a change of the Reference Signals Received Power (RSRP) values [3]. Therefore, different radio maps can be generated as a function of the selected tilt configuration. We refer to these as *tilt-dependent radio maps*.

From an operator's perspective being able to predict cell performance without carrying out extensive trials or measurement campaigns is of key importance for two reasons: firstly, extensive measurement campaigns, such as test driving, are time consuming and costly. Secondly, even if measurements were obtained inexpensively (e.g., directly from user terminals through crowd-sourcing), testing all possible antenna configurations might still be impractical at network runtime.

Given such difficulties, a solution which is particularly appealing to network operators is transferring the knowledge acquired from a single measurement campaign (for a given antenna tilt setting) to a new *domain* (a new tilt setting) without needing to acquire a complete set of additional measurements. In this case, the data distributions of the training (source) and testing (target) sets are different. Therefore, we formalize and solve this problem via *transfer learning*, a paradigm that has received increasing attention in the last few years [4].

In this paper, we study the possibility of performing transfer learning for the task of predicting the radio signal strength

Manuscript received June 25, 2019; revised October 31, 2019; accepted December 27, 2019. This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement 643002. This work extends the preliminary study in [1]. The associate editor coordinating the review of this article and approving it for publication was W. Saad. *(Corresponding author: Claudia Parera.)*

Claudia Parera is with DEIB, Politecnico di Milano, 20133 Milan, Italy, and also with Bell Labs, Nokia, 70435 Stuttgart, Germany (e-mail: claudia.parera@polimi.it).

Qi Liao and Ilaria Malanchini are with Bell Labs, Nokia, 70435 Stuttgart, Germany (e-mail: qi.liao@nokia-bell-labs.com; ilaria.malanchini@nokia-bell-labs.com).

Cristian Tatino is with ITN, Linköping University, 60221 Linköping, Sweden (e-mail: cristian.tatino@liu.se).

Alessandro E. C. Redondi and Matteo Cesana are with DEIB, Politecnico di Milano, 20133 Milan, Italy (e-mail: alessandroenrico.redondi@polimi.it; matteo.cesana@polimi.it).

Digital Object Identifier 10.1109/TCCN.2020.2964761

map of a particular BS. We start from a dataset of signal strength measurements collected from commercial, Long Term Evolution (LTE) BSs and analyze the performance of a transfer learning approach based on a deep neural network, where a *domain* is defined as the knowledge acquired for a particular antenna tilt setting. This is then transferred to a different *domain*, i.e., a different tilt configuration of the same antenna. As a benchmark, we compare the performance of our proposed method against the performance of standard machine learning techniques when applied to the same problem. The performance evaluation is carried out in two different scenarios: firstly, we use a single tilt configuration as the source domain. Secondly, we augment the source domain by adding data available from other tilt configurations of the same antenna. We study the behavior of the proposed transfer learning approach when the data available from the target tilt configuration is limited, and further analyze different strategies to select the limited points of the target domain.

In summary, the main contributions of this paper are as follows:

- We propose a transfer learning framework based on deep neural networks, i.e., Feed-Forward Neural networks (FFNs), for tilt-dependent radio map prediction. Contrary to work in the area of deep and transfer learning for computer vision and natural language processing, where more complex architectures (Convolutional Neural Networks (CNNs), BERT [5] and ULMFit [6]) are used, we evaluate the joint use of a simpler FFN architecture and transfer learning for a different kind of data, namely, radio access network data.
- We describe the optimization of the reference architecture, as well as of the parameters for both source and target domains. The system is trained exclusively on data coming from an existing network deployment.
- Through numerical experiments, we evaluate prediction performance against standard machine learning approaches. The proposed approach is shown to achieve notable prediction accuracy, specifically when the amount of data available from the target domain is limited. Moreover, under realistic assumptions on the data availability, we show the scenarios where transfer learning leads to performance improvements. Finally, we show how data augmentation leads to further performance improvement.

The rest of this paper is organized as follows: Section II reviews related works in the area of radio map prediction, with a particular focus on those works dealing with antenna tilt. It further reviews the state of the art of transfer learning and its applications in the area of network planning and optimization. Section III describes in detail the scenario outlined above, as well as preliminary data collection and data pre-processing steps. Section IV focuses on the machine learning tools used for this work. Experiments and discussion of the obtained results, with special emphasis on the use cases where transfer learning outperforms traditional machine learning approaches, are reported in Section V. Finally, Section VI concludes the paper.

II. Background and Related Work

In this section, we briefly review the works on antenna tilt-dependent radio map prediction, introduce background information on transfer learning, and comment on related applications of transfer learning techniques to wireless networks.

A. Tilt-Dependent Radio Map Prediction

Tilt-dependent radio map prediction plays a crucial role in the context of network planning and proactive network optimization [7]. The predicted propagation condition can be exploited for a reliable decision making process to dynamically optimize antenna tilts in a time-varying network environment [7], [8]. Although radio map prediction has been extensively studied [9], its dependency on antenna tilt has been investigated only in few works. The authors in [10] propose a geometrical-based extension to various traditional log-distance path loss models (Okumura-Hata, Walfisch-Ikegami) to take into account the antenna tilt during the prediction of the signal strength at a given distance from the BS. The proposed extension, named vertical gain correction (VGC), is calculated directly from the antenna patterns provided by the manufacturer and is added to the signal strength estimated by the path loss models to compensate for the antenna tilt. Experimental results on data collected from LTE BSs show that VGC improves the performance of signal strength prediction when compared to traditional models. Similarly, the work in [11] investigates the effect of antenna tilt on radio maps, comparing the path loss models developed by the 3rd generation partnership project (3GPP) [12] for different propagation environments. The results were obtained using a ray tracing tool able to take into account antenna tilts and demonstrate that changing antenna tilt has a significant impact on the shadowing map. This calls for a rethinking of currently available 3GPP propagation models and assumptions, which apply an identical shadowing map independently from the antenna tilt.

B. Overview of Transfer Learning

Traditional machine learning algorithms work under the assumptions that training and testing data are taken from the same distribution and have the same feature space. However, in real world applications these assumptions do not always hold. Firstly, the data distribution may not be static, but vary over time, making it difficult to apply a trained model to a new scenario at a different time period. Secondly, training and testing data could also differ in terms of geographic location, or the equipment used for recording the measurements (e.g., a different mobile device). In such cases, transfer learning is a promising approach for exploiting and sharing knowledge among different domains.

In this paper, a *domain* $\mathcal{D} := \{\mathcal{X}, P(X)\}$ consists of a feature space $\mathcal{X}$ and its probability distribution $P(X)$, $X \in \mathcal{X}$. A *task* $\mathcal{T} := \{\mathcal{Y}, f(\cdot)\}$ consists of a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$, where $f(\cdot)$ can be written as $P(Y|X)$, $Y \in \mathcal{Y}$ and $X \in \mathcal{X}$. Formally, the definition of transfer learning is given as follows.

Definition 1 (Transfer Learning [4]): Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.

Three seminal papers [4], [13], [14] review the state of the art of transfer learning in classification, regression, unsupervised and reinforcement learning. When dealing with a transfer learning problem, the main research questions are: *what to transfer*, *how to transfer* and *when to transfer*. *When to transfer* is mainly related to the issue of avoiding negative transfer, which happens when transfer learning has a negative impact on the performance of target learning. The literature is primarily focused on the first two questions. For the purpose of studying *what to transfer*, we will use the categorization found in [4], where transfer learning can be divided into:

- *Inductive transfer:* Different source and target tasks and same or different source and target domains.
- *Transductive transfer:* Same source and target tasks but different source and target domains.
- *Unsupervised transfer:* Similar to inductive transfer, with different but related source and target tasks. The focus is solving an unsupervised learning task on the target domain. There is no labeled data available from source and target domains during training.

To answer the question *how to transfer*, the most common transfer learning approaches are:

- *Instance transfer:* Labeled samples in the source domain are reweighted and used in the target domain; it can be applied to inductive and transductive learning [15], [16], [17].
- *Feature transfer:* Aims at finding a 'good' feature representation that can minimize the domain difference, as proposed in [18], [19]; it is applied to inductive and transductive learning.
- *Parameter transfer:* Works under the assumption that individual models for related tasks share parameters or a combination of hyperparameters [20], [21]; it is mostly applied to inductive and transductive transfer learning.
- Relational knowledge transfer: Is applied to problems where there is some kind of relation in the data (e.g., network or social network data) [22], [23]; it is mostly applied to inductive transfer learning.

C. Applications of Transfer Learning

Some of the areas where transfer learning has been successfully applied are computer vision, natural language processing and speech recognition [4]. Due to recent advances in the field of deep learning, recent approaches combine deep and transfer learning. For instance, in [24] mid-level image representations learned with a CNN are transferred to other visual recognition tasks. The same idea is followed in [25] for character recognition from Latin to Chinese. However, the applications of transfer learning in the field of wireless and mobile networks are still limited [26]. Work has been done for localization by transferring knowledge across devices, time and space in [27], [28], [29], [30]. More recently, transfer learning has been applied to caching [31], resources optimization [32], fault classification [33] and resource management in Wireless Virtual Reality [34].

D. Motivation of Our Study

Our work shares the same research objectives as the work on tilt-dependent radio map prediction (Section II-A), but with one fundamental difference: in all the aforementioned works the source domain for predicting the signal strength is similar to, or the same as, the target domain. For example, the signal strength radio map of an antenna under a given tilting configuration is predicted using available signal strength samples collected for the same antenna in the same tilt configuration. Instead, we analyze the case where the performance of the target antenna configuration is predicted using training data from a different tilt configuration. In our previous work [1], we investigated the dependency between the transferability of the knowledge and the domain difference, when considering the task of tilt-dependent radio map prediction and by using standard machine learning tools. In this work, we aim to solve a similar problem, with improved performance, by applying transfer learning and exploiting different data sources as source and target domains.

Our work mainly falls into the category of *transductive learning* ($\mathcal{T}_S = \mathcal{T}_T$ and $\mathcal{D}_S \neq \mathcal{D}_T$). Furthermore, our solution is inspired by the *feature transfer* and *parameter transfer* approaches introduced in [24], where the authors propose to extract some internal layers from a CNN, trained with sufficient data collected from the source domain. They add an adaptation layer to correct the difference between distributions in the source and target domain. The resulting network is trained with a limited amount of data from the target domain. However, unlike the approach proposed in [24], we do not use CNNs, due to the format, scarcity and small feature space dimension (i.e., geospatial information and Reference Signal Received Power (RSRP) values) of the collected data. For the same reasons, we do not consider recurrent architectures such as Long Short-Term Memory (LSTM) networks, which have been adopted successfully for problems where data, unlike the current radio maps, has a strong temporal structure (e.g., time series prediction or speech recognition) [35]. Instead, we take the internal layers of a fully connected FFN, trained in a given tilt configuration (source domain), and add a new layer. Then, we retrain the final network on a new tilt configuration (target domain). To exploit parameter transfer, we assume that different domains share the same combination of hyperparameters (same neural network architecture). In addition, we quantify the minimum amount of labeled data required from the target domain to carry out predictions. We showcase real world scenarios where a transfer learning solution outperforms traditional machine learning algorithms. Finally, we show how augmenting the source domain by adding data from other available tilt configurations of the same antenna helps to improve the performance of our transfer learning approach. To the best of our knowledge, this is the first work transferring knowledge across different network configurations by partially retraining deep neural networks, in the area of wireless and mobile networks.

III. Problem Statement and Dataset

We address the following problem: "*how to predict the performance of a given network configuration by leveraging information from different network configurations*". The performance measure that we target is the received signal strength in the downlink. The network configuration domains include the tilting configurations of the transmitting BSs.

We consider a BS that can work in H different tilt configurations, indexed by $h = 1, \ldots, H$. Let $s_h(\mathbf{x}_i)$ be the measured signal strength received at location $\mathbf{x}_i = \{y_i, z_i\}$ when the h-th tilt configuration is selected at the BS, where y_i and z_i indicate the latitude and the longitude of the i-th location, respectively. Let $\mathcal{M}_h$ be the set of location indexes where measurements have been taken with configuration h.

The problem at hand can be defined as follows: given $\{s_h(\mathbf{x}_i) : i \in \mathcal{M}_h\}$, estimate the unknown signal strength $\hat{s}_n(\mathbf{x}_j)$ at the same or different locations, $\mathbf{x}_j$, with $j \in \mathcal{M}_n$, under different network configuration domains, $n \neq h$.

A. Data Collection

The dataset used in this work is composed of RSRP outdoor measurements collected in Espoo, Finland, in November 2016 from two commercial LTE BSs with three different 120° sectors each and operating at 2.6 GHz. Figure 1 shows the positions of the two antennas and the representation of the target area. The measurements were collected from three different Physical Cell Identifiers (PCIs), which will be referred to as PCI 1, 2 and 3. PCIs 1 and 2 refer to two different sectors of the same BSs, whereas PCI 3 is a sector of a different BS. The RSRP measurements were collected using an Android device equipped with an application capable of storing the RSRP from all the received cells, the cell identifier, the Global Positioning System (GPS) position of the device and the timestamp. Such measurements were carried out at a frequency of 1 Hz while walking along routes of 8 km within each cell coverage area, with a minimum and maximum distance from the BS of 30 m and 900 m, respectively. By design, the testing paths were planned to include different propagation conditions: university campus with two or three-story buildings, residential areas, parking lots, lower density rural and open areas with different types of roads (e.g., pedestrian, cycling and main roads). Each testing path was walked once for each electronic tilt setting. The available tilt settings are 2, 3 and 6 degrees for each PCI, respectively. The receiver was placed at the height of 1.5 m and always kept at the same orientation. The weather conditions were stable and cloudy, and the route was covered by snow for most of the measurement campaign. The RSRP values were collected from an operating mobile network. According to [36], these values include the power from co-channel serving and non-serving cells as well as adjacent channel interference, but only on the resource elements that carry reference signals. Since these values are measured only in the symbols carrying the reference signal, they exclude most of the wide band noise and interference from other cells. Overall, they are proportional to the SNR on average [37]. Therefore, they are still a good indicator to be used in radio map reconstruction, reflecting the channel propagation conditions.

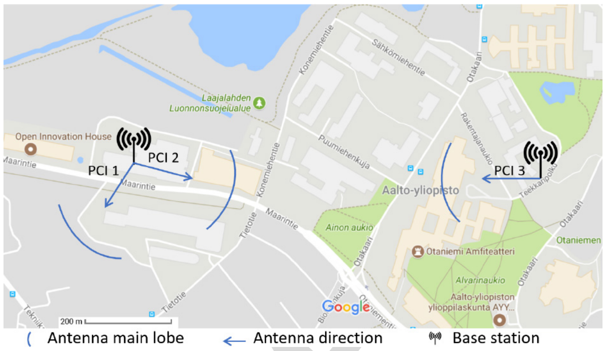


Fig. 1. Map showing the BS positions and the PCIs in the reference dataset.

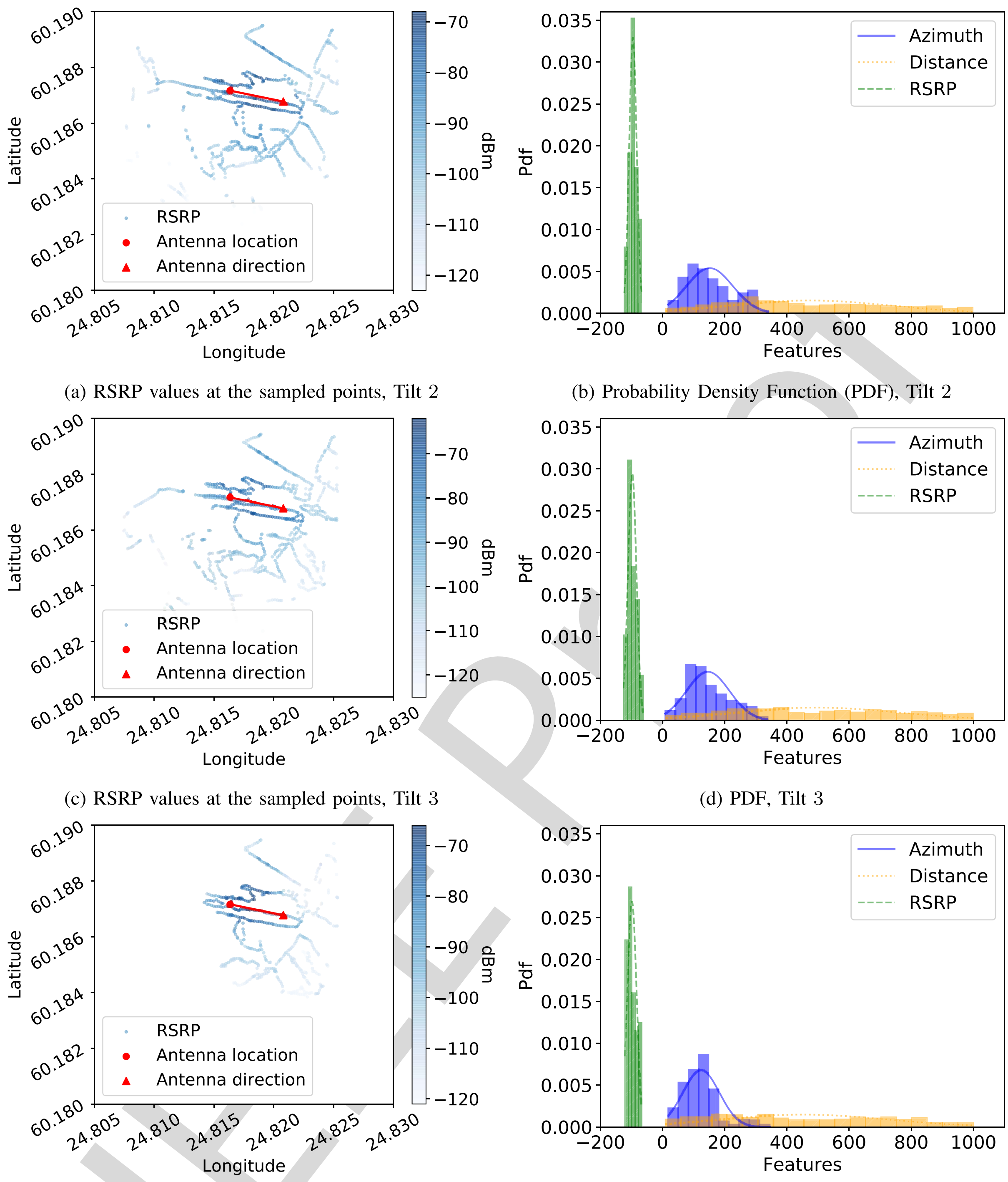
B. Data Preprocessing

In total, about $3 \cdot 10^5$ RSRP measurements were obtained. Each observation contains the following fields:

- Measurement position (latitude and longitude coordinates)
- RSRP value (downlink signal strength)
- PCI (physical cell identifier)

The raw dataset was preprocessed to remove corrupted samples: for example, at the beginning of each experiment the GPS receiver requires some initialization time during which position is recorded incorrectly. Moreover, we overlaid the considered area with a grid. For each grid element of size 20 m × 20 m, we replaced the RSRP values with their average to reduce noise. After the preprocessing steps, the reduced dataset consisted of ~ 600 observations per PCI and per tilt configuration, for a total of $\sim 5.8 \cdot 10^3$ measurements. Before training, the data is scaled between 0 and 1 by using a Min-Max scaler, which is fitted to the training set and applied to the cross validation and test sets. The scaling transformation is then reversed before evaluating the algorithm performance. In our previous work [1], we analyze the transferability across different tilt settings of the same PCI as well as the transferability across different PCIs. In particular, we show that the transferability within the same PCI is much higher than the transferability across different PCIs. Therefore, we focus hereafter on the task of transferring the knowledge from one tilt configuration to another within the same PCI. Unlike our previous work, in which the standard machine learning tools are used, we apply transfer learning with deep neural networks.

Figure 2 shows a representation of the data collected for different tilt configurations of PCI 1. Figures 2(a), 2(c), 2(e) show RSRP values (in dBm) over the considered geographic area, when the antenna was tilted at 2, 3 and 6 degrees, respectively. It can be observed that the spatial distribution of the data follows a similar pattern for different tilt configurations of PCI 1. For example, points located in the main direction of the antenna have higher signal strength values than the rest of the points. In addition, points closer to the



(a) RSRP values at the sampled points, Tilt 2

(b) Probability Density Function (PDF), Tilt 2

(c) RSRP values at the sampled points, Tilt 3

(d) PDF, Tilt 3

(e) RSRP values at the sampled points, Tilt 6

(f) PDF, Tilt 6

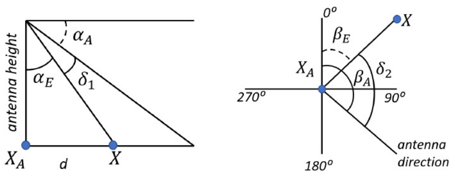
Fig. 2. Tilt-dependent radio maps, normalized histograms and Probability Density Functions (PDFs) for three metrics RSRP, azimuth and distance, PCI 1.

antenna also follow a similar pattern, while points which are far apart have lower RSRP values. To give an idea of the domain differences, in Figures 2(b), 2(d), 2(f) we show the normalized histograms, and the continuous approximations of the PDFs of the azimuth, distance and RSRP for the different tilt configurations of PCI 1.

Even if some similarities can be observed between the statistical characteristics of the data collected under different tilt configurations, the data does not come from the same distributions. For example, the azimuth distribution for a greater tilt value (Figure 2(f)) has a lower standard deviation than the distributions for lower tilt values (Figures 2(b) and 2(d)).

IV. Prediction Approaches

Given the base station location $\mathbf{x}_A$, let $\mathbf{x}$ and h be the target position and the configured antenna tilt, respectively. The



(a) Relative distance (b) Relative azimuth

Fig. 3. Relative angles on the vertical (left) and horizontal (right) planes between the antenna pointing direction and the direction towards the test position **x**.

following set of features, derived from (**x**, *h*) and shown in Figure 3, is considered for the prediction task:

- the physical *distance* between the antenna and the measurement position, $d(\mathbf{x}) := d(\mathbf{x}, \mathbf{x}_A)$
- the *relative elevation angle* between the down-tilt of the antenna and the vertical direction from the antenna emitting element to the measurement position, defined as:
$$\begin{aligned}\delta_1(h, \mathbf{x}) &= 90^\circ - (\alpha_A + \alpha_E(\mathbf{x}, \mathbf{x}_A)) \\ &= 90^\circ - (h + \alpha_E(\mathbf{x}, \mathbf{x}_A)), \qquad (1)\end{aligned}$$
where $\alpha_A := h$ is the antenna down-tilt (mechanical plus electrical) and α_E is the angle at which the antenna 'sees' the target position depending on the antenna position $\mathbf{x}_A$ and the target location **x**
- the *relative azimuth* between the horizontal orientation of the antenna and the horizontal direction to the measurement position defined as:
$$\delta_2(\mathbf{x}) = \beta_A - \beta_E(\mathbf{x}, \mathbf{x}_A), \qquad (2)$$
where β_A denotes the horizontal orientation of the antenna and β_E is the horizontal orientation of the target position with respect to the antenna position

Each sample in the training dataset is, therefore, associated with a tuple of values (d, δ_1, δ_2). The logarithmic transformation is applied to *d* since the RSRP values are measured in dBm. Finally the feature vector $[d(\mathbf{x}), \delta_1(\mathbf{x}, h), \delta_2(\mathbf{x})]^T$ is obtained and used as input to our models.

A. Transfer Learning Approach

The proposed transfer learning approach has been inspired by the fields of computer vision and natural language processing [24], [25], where deep neural networks constitute the state of the art for classification and prediction tasks. The core idea of our approach is to train a neural network for the signal strength prediction task in a source domain (reference tilt configuration) and then 'wisely' build a new neural network to obtain fine-grained predictions in the target domain (target tilt configuration). The neural network architectures used in our approach are FFNs, which are well-known for being powerful nonlinear function approximators [38]. We opt for FFNs instead of more complex network architectures, such as CNNs or recurrent neural networkss (RNNs) for two main reasons. Firstly, from preliminary experimental results (see Figure 6), we observed that the achieved training and cross validation losses are already very low and close to each other for the problem at hand. Therefore, using a more complex architecture with the same limited amount of data available for training could lead to a bigger gap between training and cross validation, causing overfitting and thus worsening the performance. Secondly, more complex architectures would require more parameters and hyperparameters to be found, causing an increased training time.

We use the Mean Square Error (MSE) as loss function, which is the standard metric used in regression tasks. Here the goal is to minimize the difference between the real and predicted RSRP values. It is worth noting that the MSE is well known for being sensitive to outliers, however this is not a concern in this case since outliers have been removed in previous preprocessing steps (see Section III-B). By using FFNs as the basic building blocks of our architecture, the flow of information only travels forward, and the layers of the network are fully connected. Formally, FFNs learns a combination of parameters to find the best function approximation. In our case, we aim at finding a set of parameters $\boldsymbol{\theta}$ for the hidden layers and a set of parameters *w* for the output layer to estimate $\hat{s}(\mathbf{x}) \in \mathbb{R}^q$ for $\mathbf{x} \in \mathbb{R}^p$, as shown in Eq. (3):

$$\hat{s}(\mathbf{x}) = f(\mathbf{x}; \boldsymbol{\theta}, \mathbf{w}) = \boldsymbol{\phi}(\mathbf{x}, \boldsymbol{\theta})^T \mathbf{w} \qquad (3)$$

where $\boldsymbol{\phi} : \mathbb{R}^p \to \mathbb{R}^q$, is a nonlinear transformation defining the hidden layers, and parameters $\mathbf{w} \in \mathbb{R}^q$ map from $\boldsymbol{\phi}$ to the desired output. Each input is represented by a tuple containing distance, relative azimuth and relative angle (i.e., (d, δ_1, δ_2)) and each output is the RSRP value $\hat{s}(\mathbf{x})$ associated to a given input. Therefore, $p = 3$ and $q = 1$ are the input and output dimensions, respectively.

The proposed transfer learning approach is composed of the following:

- $\mathcal{D}_S$*:* source domain which consists of the feature space of the reference tilt configuration and its marginal probability distribution
- $\mathcal{D}_T$*:* target domain which consists of the feature space of the target tilt configuration and its marginal probability distribution
- $\mathbf{M}_S = \hat{f}_S(\cdot)$*:* an FFN with *n* layers approximating the predictive function in the source domain $f_S(\cdot)$
- $\mathbf{M}_T = \hat{f}_T(\cdot)$*:* an FFN with *m* layers approximating the predictive function in the target domain $f_T(\cdot)$
- $\{p_1, \ldots, p_K\}$*:* the best combination of hyperparameters shared by both FFNs associated with the source and target domains respectively.[1]

The steps of our transfer learning algorithm are defined as follows:

1) We select the source domain $\mathcal{D}_S$ and train $\mathbf{M}_S$ on $\mathcal{D}_S$, finding the best combination of hyperparameters $\{p_1, \ldots, p_K\}$. We use Bayesian optimization [39] since it is an effective way of finding a suboptimal solution in less time, when compared to random search [40], for example. The problem of choosing the hyperparameters is modeled as a sample of a Gaussian process (GP). We start with an initial combination of hyperparameters and dynamically update the searching space based on the

[1] For parameter transfer we assume that the models for source and target domains share a combination of hyperparameters.

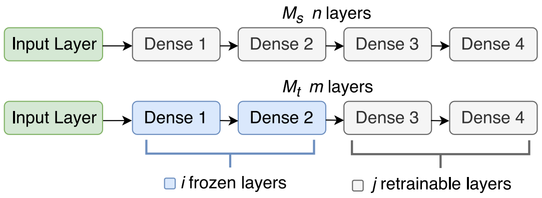


Fig. 4. Transfer learning model.

built surrogate probability model mapping from hyperparameters to the probability of a score on the objective function (see Section V-B1 for numerical results). It is worth noting that the optimization process has been carried out on $\mathcal{D}_S$ since we assume we do not have enough data available from $\mathcal{D}_T$ to find a model that perform well on $\mathcal{D}_T$. Moreover, the main goal of our approach is learning the best model possible on $\mathcal{D}_S$ by using sufficient data and transferring this knowledge to $\mathcal{D}_T$ which has limited data. After the Bayesian optimization step, $\mathbf{M}_S$ is trained on $\mathcal{D}_S$.

2) Once we have obtained $\mathbf{M}_S$, we model $\mathbf{M}_T$ by taking the first $i \leq m$ layers of $\mathbf{M}_S$ with the associated weights and adding new $j \leq n$ layers that are initialized with random weights. The reason for this is that the first layers of the network can capture more general characteristics about the feature space, while the latter ones capture more specific behaviors. For choosing the best values of *i* and *j*, we tried all possible combinations of values such that $0 \leq i \leq m$ and $j = m - i$ and selected the one that led to the best accuracy (details are provided in Section V-B2). Figure 4 shows a graphical representation of $\mathbf{M}_S$ and $\mathbf{M}_T$, where $\mathbf{M}_T$ contains the first three layers of $\mathbf{M}_S$ and two new layers.

3) Finally, we train $\mathbf{M}_T$ on the few data available from $\mathcal{D}_T$ using the hyperparameters $\{p_1, \ldots, p_K\}$. We freeze the first *i* layers and retrain only the last *j* layers of $\mathbf{M}_T$ with data from $\mathcal{D}_T$. We refer to this approach as DNN T. It is worth observing that $\mathbf{M}_S$ and $\mathbf{M}_T$ have the same complexity (i.e., number of layers and hidden units) in order to ensure fairness when comparing $\mathbf{M}_S$ and $\mathbf{M}_T$. In addition, using a much more complex architecture with a limited amount of data on $\mathcal{D}_T$ is more likely to increase overfitting and worsen the performance, while using a much simpler architecture does not improve performance (see DNN T 2F 1R on Figure 7).

We use the Keras framework [41] on top of TensorFlow [42] due to its flexibility for implementing this transfer learning approach and performing hyperparameters search. In total, the training and testing phases of the two models do not last more than two minutes. We use a laptop with 16 GB of RAM and a 7th generation, Intel Core i7 processor.

B. Baseline Methods

In this section, we describe the baseline methods used to benchmark our work: Heuristic (H) using data provided by antenna manufacturer as well as k-Nearest Neighbors (k-NN) and Random Forest (RF), which performed the best for the task at hand in our previous work [1].

1) Heuristic: This is the simplest baseline method, where the predicted values are extracted from the data sheets provided by the antenna manufacturer. Given a set of locations at a given tilt configuration, for each sample we create the feature vector by calculating *distance*, *relative angle* and *relative azimuth* (Section IV). In a second step, we use the data sheets provided by the antenna manufacturer to extract the antenna gain on the vertical and horizontal planes. Finally, we apply the path loss model to calculate the predicted values. Formally, the process is defined as follows:

1. Given $\mathcal{M}_h$ as the set of location indexes where measurements for the considered base station running configuration *h* have been taken, we calculate for each location $\mathbf{x} \in \mathcal{M}_h$ a tuple of values (d, δ_1, δ_2). Then we create the feature vector $\mathbf{z} := [d(\mathbf{x}), \delta_1(\mathbf{x}, h), \delta_2(\mathbf{x})]^T$ as shown in Section IV.
2. Let $\eta(\mathbf{x})$ and $\gamma(\mathbf{x})$ be the horizontal and vertical gain of the antenna in dB, respectively, as taken from the manufacturer antenna sheets. Given the known position $\mathbf{x}$, we formally define $\Delta(\mathbf{x})$ as:
$$\Delta(\mathbf{x}) = \eta(\mathbf{x}) + \gamma(\mathbf{x}) \tag{4}$$
3. Given $\Delta(\mathbf{x})$, we use the path loss model to generate the labels, $\hat{s}(\mathbf{x})$, by applying the following:
$$\hat{s}(\mathbf{x}) = \phi_0 + \phi_1 10 \log(d(\mathbf{x})) - \Delta(\mathbf{x}), \tag{5}$$
where ϕ_0 and ϕ_1, similar to [10], are the linear regression coefficients calculated for the reference dataset.

2) k-Nearest Neighbors With Inverse Distance Weighting: This technique is one of the simplest multivariate interpolation methods which extends the classical nearest neighbor approach [43]. We apply the technique on the same feature vector $\mathbf{z}(\mathbf{x}) := [d(\mathbf{x}), \delta_1(\mathbf{x}), \delta_2(\mathbf{x})]^T$ as defined in the above-mentioned Heuristic approach. It predicts the signal at an unknown target location $\mathbf{x}$ (corresponding to a feature vector $\mathbf{z}(\mathbf{x})$) as a weighted average of the signals at the *k* locations with the closest distances calculated based on feature vectors.

$$\hat{s}(\mathbf{z}) = \sum_{i \in \mathcal{M}(\mathbf{z})} \omega_i s(\mathbf{z}_i) \tag{6}$$

The set $\mathcal{M}(\mathbf{z})$ includes the feature vectors which are the closest to the unknown target vector $\mathbf{z}$, with cardinality $|\mathcal{M}| = k$. Weights ω_i are chosen to be inversely proportional to the distance $d(\mathbf{z}_i, \mathbf{z})$ and their sum is normalized to one, using the equation below:

$$\omega_i = \frac{d(\mathbf{z}_i, \mathbf{z})^{-1}}{\sum_{j \in \mathcal{M}(\mathbf{z})} d\left(\mathbf{z}_j, \mathbf{z}\right)^{-1}}. \tag{7}$$

3) Random Forest: Is one of the ensemble methods used for classification and regression purposes. The algorithm was introduced by Ho [44] in 1995, and later extended by Breiman and Cutler [45], it uses the idea of bagging to perform predictions. During the process several trees are grown independently using different bootstrapped samples of the data and majority voting or averaging is used for the final prediction. In contrast to traditional trees, the variable used to perform the split in each node is chosen randomly from a set of predictors [45]. RF is known to sometimes outperform other machine learning techniques, such as neural networks, due to its resistance to overfitting [46].

V. Experiments

In this section, we describe the set of experiments carried out. We compare the prediction error of our transfer learning method (i.e., DNN T in Section IV-A) against the baseline methods (i.e., H, k-NN, RF in Section IV-B). In the following, the suffix T is used to denote the transfer learning approach (i.e., DNN T). Similarly, the suffix S is used to denote the methods that do not use transfer learning (i.e., H S, k-NN S, RF S and DNN S). It is worth noting that DNN T is trained on data from a different tilt configuration (source domain) whereas H S, k-NN S, RF S and DNN S are trained on data from the same tilt configuration (target domain). In this way, we compare the performance of the proposed transfer learning solution against the performance of traditional machine learning solutions to reveal the scenarios where a transfer learning solution is preferred. We also train a model on the source domain and apply it to the target domain without the retraining and fine tuning step. This last approach is referred as DNN BS. It does not require data from the target domain since no retraining is performed. In this case, the purpose is carrying out comparisons against the transfer learning solution to evaluate the real need for the retraining and fine tuning step.

We carry out two different sets of experiments that differ in the way the source domain is built. In Section V-C, the source domain consists of measurements from a single tilt configuration, which differs to the one used for target domain. In Section V-D we augment the source domain by adding measurements from other available tilt configurations of the same PCI. In both cases, we analyze the impact on the performance when a limited amount of data from the target domain is available in the training phase. We study two strategies to select data from the target domain: (i) uniformly distributed in the reference area or ii) non-uniformly distributed according to a predefined sampling strategy (i.e., different distance ranges from the antenna location).

For each tilt configuration the amount of data available is about 600 measurements. In all the experiments, the data is divided into training, cross validation and test sets. We use 80% of samples for training, 10% for cross validation and 10% for testing. We vary the quantity of data taken from the target domain for training or fine tuning. For the DNN T, this is the number of samples used to train and fine tune M_T. For the k-NN S, RF S and DNN S this is the quantity of data available for training a model on the target domain using data from the same target domain. In contrast, H S does not need training data. One of the main objectives is to map the amount of labeled data required from the target domain and corresponding performance, assessed in terms of Mean Absolute Percentage Error (MAPE), which is defined as:

$$\mathrm{MAPE} = \frac{100}{k}\sum_{i=0}^{k-1}\left|\frac{s_i - \hat{s}_i}{s_i}\right|, \tag{8}$$

where *k* is the number of target positions in the testing dataset.

A. Domain Distance

Since the performance of the transfer learning approach depends on the similarity between the training and testing sets on the target domain, we introduce a measure of the *degree of similarity* between datasets which is then used throughout this section. We quantify similarity in terms of Kullback-Leibler (KL) divergence index [47], which measures the relative entropy of a given probability distribution with respect to another one. Given two reference datasets, one used for training and one used for testing (both in the target domain), we derive the KL divergence indexes of the probability distributions of the logarithm of the distance (*d*), relative angle (δ_1) and relative azimuth (δ_2) of the two datasets. Formally, the symmetric KL divergence index of the distance probability distributions is given by:

$$\begin{aligned} SD_{KL}(d) = &\sum_{i=1}^{k} P_d^{(\mathrm{tr})}(i)\log\frac{P_d^{(\mathrm{tr})}(i)}{P_d^{(\mathrm{te})}(i)} \\ &+ \sum_{i=1}^{k} P_d^{(\mathrm{te})}(i)\log\frac{P_d^{(\mathrm{te})}(i)}{P_d^{(\mathrm{tr})}(i)}, \end{aligned} \tag{9}$$

where $P_d^{(\mathrm{tr})}(i)$ and $P_d^{(\mathrm{te})}(i)$ with $i = 1 \dots k$ define the discrete probability distributions of the distance in the training and testing sets of the target domain, respectively and *k* is the amount of bins used to estimate either $P_d^{(\mathrm{tr})}$ or $P_d^{(\mathrm{te})}$. Similar definitions hold for the KL divergence indexes related to the relative angle δ_1 and relative azimuth δ_2. Finally, to give a more succinct representation of domain similarity, we introduce the Domain Distance (DD) measure by summing the three indexes together:

$$\mathrm{DD} = SD_{KL}(d) + SD_{KL}(\delta_1) + SD_{KL}(\delta_2). \tag{10}$$

Figure 5 shows the average DD across PCIs for all the possible combinations of training and testing sets on the target domains and the amount of points from the target domain used for training. The solid curve in Figure 5 shows the DD when the available samples taken from the target domain for training or fine tuning are uniformly sampled in the reference area. The dashed curve in Figure 5 shows the DD when the samples are taken between 300 m and 600 m of distance from the antenna location. Smaller DD values indicate higher domain similarity and vice versa. For instance, when the samples are uniformly distributed the similarity between training and testing sets in the target domain is higher, which makes the DD values lower, i.e., they range from 0.8 to 1.3 (see Figure 5 solid curve). In contrast, when the available measurements are located at a certain distance range from the antenna (i.e., 300 to 600 m), similarity is lower, meaning the DD values are higher ranging from 1.5 to more that 2 (see Figure 5 dashed curve).

B. Hyperparameter Search

Hyperparameters are chosen in a hybrid manner by using a mixture of Bayesian optimization and manual fine tuning.

1) Hyperparameter Search on $\mathcal{D}_S$: Bayesian optimization requires, as starting point, a network architecture that converges. It also requires that the hyperparameters search space

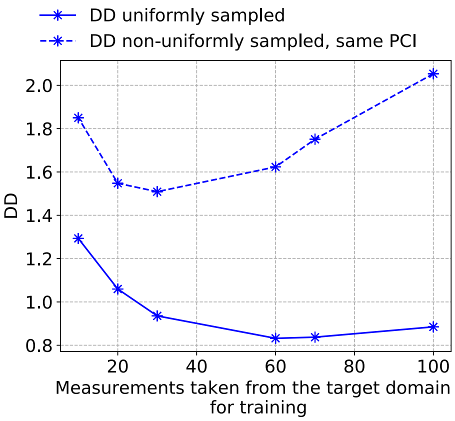


Fig. 5. Domain distance.

TABLE I
MAPE for Variations in the Amount of Layers and Hidden Units for PCI 1, Tilt 2, 3 and 6

$\mathcal{D}_S$ Network architecture	Avg. across Tilts
4, 10, 4, 1	**4.30**
4, 10, 4, 4, 1	11.94
4, 10, 1	5.58
4, 20, 8, 1	6.82
4, 5, 2, 1	5.59

is specified. We begin with an architecture that contains 4 layers with 4, 10, 4 and 1 hidden units, respectively, $1e-3$ as the initial learning rate and ReLu as the initial activation function. The activation function search space contains Sigmoid and ReLu, and the learning rate search space goes from $1e-7$ to $1e-1$. After 50 iterations the process converges and we find out that for all the tilt configurations and PCIs the best learning rate is approximately 0.099, and the choice of activation function that leads to the minimal error is Sigmoid. During this process, the model is shown to achieve good performance in the source domain. Figure 6 shows the training and cross validation errors for PCI 1 and tilts 2, 3 and 6. Comparable results are achieved for the rest of the PCIs and tested tilt combinations. It can be observed that the training and cross validation errors decrease dramatically during the first 150 epochs for tilts 2 and 3 and the first 20 epochs for tilt 6. After this they keep decreasing steadily, becoming very close to 0, which means the chosen architecture fits the data coming from $\mathcal{D}_S$. We note that both errors are close to each other, meaning that our model is not overfitting. Once the learning rate and activation functions are chosen, we carry out experiments on $\mathcal{D}_S$, increasing and decreasing the model complexity by adding and removing layers and hidden units, respectively. Table I shows the MAPE obtained leveraging the tested architectures. The reported MAPE is an average across all the available PCIs and all the available combinations of tilt configurations as source and target domain. It can be observed that increasing or decreasing complexity worsens the performance for all the tilt configurations on average, therefore the initial combination of 4 layers containing 4, 10, 4 and 1 hidden units respectively is the one that leads to the best performance. Table II summarizes the best combination of hyperparameters found by a mixture of Bayesian and manual optimization.

TABLE II
Hyperparameters Found by Bayesian and Manual Optimization

Number of epochs	500
Batch size	128
Number of inputs	3
Number of layers	4
Hidden units per layer	4, 10, 4, 1
Activation function	Sigmoid
Optimizer	Adam
Learning rate	0.099

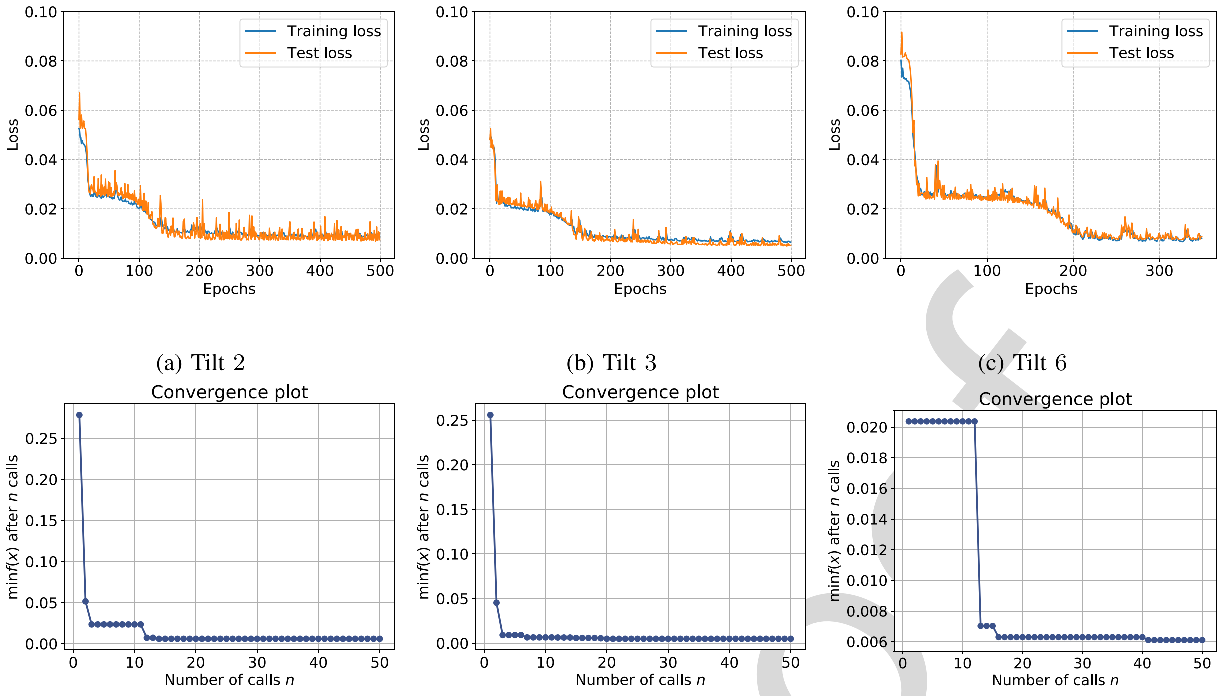
2) Frozen and Re-Trainable Layers: As explained in Section IV-A, the amount of layers to freeze and retrain is chosen through an empirical approach, trying all possible combinations and choosing the best one. Figure 7 reports the MAPE averaged across all the PCIs and all the possible combinations of tilt configurations as source and target domain when using different numbers of layers to freeze, i and the number of retrainable layers, j. DNN T method indicates that the weights in the retrainable layers of $\mathcal{M}_T$ are randomly initialized, whereas the DNN T W indicates that the weights in the retrainable layers are initialized with the weights from $\mathcal{M}_S$ after training. We use F and R to denote the number of layers to freeze and retrain on $\mathcal{M}_T$, respectively. We select $i = 2$ and $j = 2$ (i.e., DNN T 2F 2R), since it is the combination of values that leads to the best MAPE on $\mathcal{D}_T$.

C. Single Tilt Transfer

We use a dataset obtained under a given tilt setting (source domain) to predict the performance of the same antenna under a different tilt configuration (target domain). In particular, we consider two different scenarios: when the data available from the target domain is limited and sampled uniformly (see Section V-C1) and when the data is still limited but sampled according certain criteria, for instance at a given range of the antenna location (see Section V-C2).

1) Limited and Uniformly Sampled Measurements: In this case measurements represent a wide range of relative distances, azimuth and RSRP values. The amount of instances taken for training or fine tuning varies between 0 to 100. Figure 8 shows the average MAPE across all the PCIs and possible pairs of training and testing tilt combinations, obtained by the different prediction approaches described in Section IV. We can draw the following conclusions:

- All the machine learning methods (i.e., k-NN S, RF S, DNN S, DNN T) outperform the heuristic approach (i.e., H S) for any number of instances taken from the target domain for training or fine tuning. Therefore, the machine learning algorithms trained on real data are more effective at capturing the non linearity of RSRP values than the heuristic approach which uses the path loss model to extract RSRP values from the sheets provided by the antenna manufacturer.
- The prediction error is impacted by the amount of samples taken from the target domain. In particular, the amount of data taken from the target domain can be decreased by up to 90%, if we consider an initial amount of 590 instances taken for training or fine tuning, with



(a) Tilt 2 (b) Tilt 3 (c) Tilt 6

(d) Tilt 2 (e) Tilt 3 (f) Tilt 6

Fig. 6. Training curves and Bayesian convergence on the source domain, PCI 1.

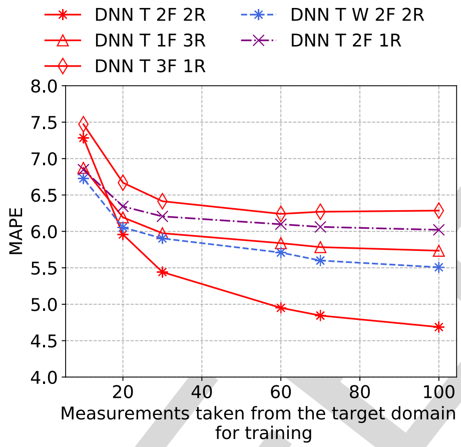


Fig. 7. MAPE for the different values of frozen (F) and retrainable (R) layers using random or source domain weights initialization.

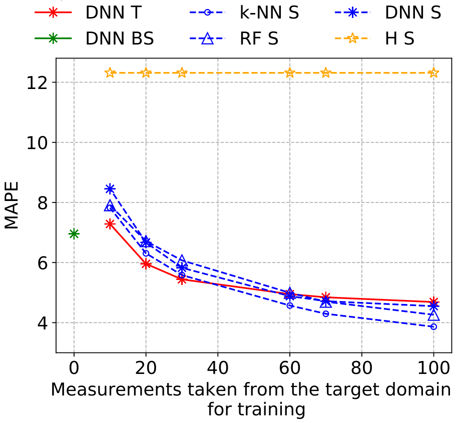
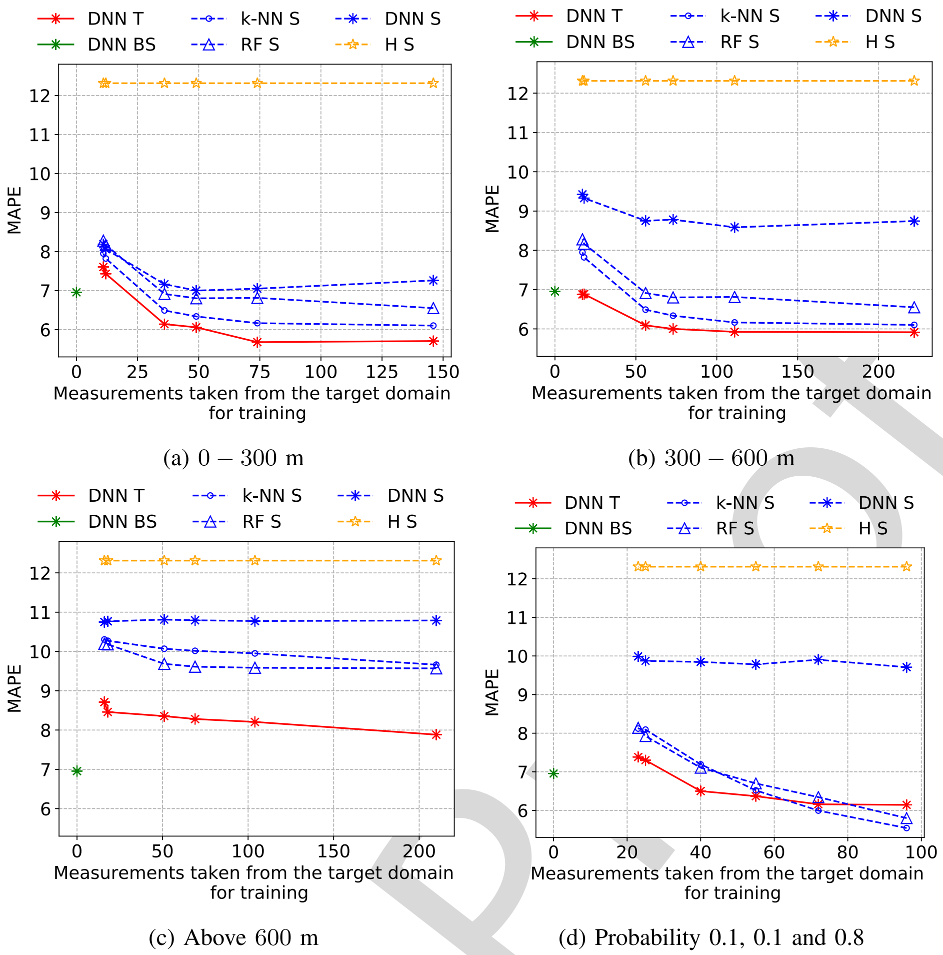


Fig. 8. MAPE when training or fine tuning on uniformly sampled measurements.

a maximum increase in error rate of 2% for the transfer learning approach. It is worth noting that if no data is taken from the target domain, the transfer learning approaches must be used under the assumptions of traditional machine learning, where source and target domain are similar. As this is not the case, DNN T outperforms DNN BS when the amount of instances taken for training is more than 20. This justifies the need to model our problem under the framework of transfer learning in order to decrease the prediction error.

- The transfer learning approach (i.e., DNN T) outperforms the methods that use data from the same tilt configuration (k-NN S, RF S and DNN S) for training, when the amount of samples taken from the testing set is less than 40 instances out of 590. This is because, transfer learning approaches better capture the physical properties of antenna propagation. Thus, being more robust when information is missing from the cross-domain.
- When the number of data samples is larger than 60, transfer learning performs worse than the non-transfer methods. This indicates, more than 60 points chosen uniformly for training a model are enough to capture all the possible patterns (different RSRP values) in a given radio map while achieving good prediction error (see DD values uniformly sampled curve in Figure 5 for more than 60 measurements). In contrast, if less than 60 points



(a) $0 - 300$ m

(b) $300 - 600$ m

(c) Above 600 m

(d) Probability 0.1, 0.1 and 0.8

Fig. 9. MAPE when training or fine tuning on non-uniformly sampled measurements.

are taken, which accounts for 12% of the total amount of points initially used for training, there are too few points to capture all the possible patterns. Therefore, using data from $\mathcal{D}_S$ and retraining via transfer learning leads to performance improvement when the number of data samples is less than 12%.

2) Limited and Non-Uniformly Sampled Measurements: We define different antenna distance ranges since we assume to have available only measurements collected in one of those locations. Figure 9 shows the obtained average MAPE across all PCIs and all possible combinations of training and testing tilts. Figures 9(a), 9(b) and 9(c) show the average MAPE for measurements collected between 0 to 300, 300 to 600, and more than 600 m from the antenna location, respectively. In addition, we consider in Figure 9(d), the case where we take points from all of the three ranges with probability 0.1, 0.1 and 0.8, respectively. This set of experiments, is motivated by the fact that in a realistic scenario points at a given distance range might be the only ones available to carry out predictions. For instance, in some areas it might not be possible to take measurements due to the existence of obstacles or private properties. In other cases there might be budget constraints (both in terms of resources and time) which do not allow for an extensive measurement campaign of the whole area. These scenarios are particularly challenging, because traditional methods do not work at their best. Therefore we focus on these to highlight the benefits of transfer learning. To study these cases, we consider two possible options: (i) we use the available points as the training set to carry out predictions under the same tilt configuration (i.e., H S, k-NN S, RF S, DNN S) or (ii) we use the available points as part of the retraining step in the transfer learning pipeline (i.e., DNN T). In both cases, the model output is the predicted radio map for the whole area.

We can draw the following conclusions:

- As before, all the machine learning methods (i.e., H S, k-NN S, RF S, DNN S and DNN T) outperform the heuristic approach (i.e., H S) and are impacted by the amount of instances taken from the target domain for training or fine tuning the models.
- The prediction error is never more than 2% higher than when using uniformly sampled data. An increase in error is expected since training and testing sets in the target domain are more dissimilar than when samples are taken uniformly (see DD values in Figure 5). However, depending on the application and data restriction when collecting samples, non-uniformly distributed data could still be used to carry out predictions when uniformly sampled data is not available.
- The transfer learning approach (i.e., DNN T) outperforms the methods that use data from the same tilt configuration (i.e., k-NN S, RF S and DNN S) to carry out predictions

TABLE III
DD Values Before and After Data Augmentation

Algorithm	DD	$\mathcal{D}_S$ Number of samples
No augmentation	1.15	600
Augmentation same PCI	1.39	1200
Augmentation other PCIs	3.53	4800

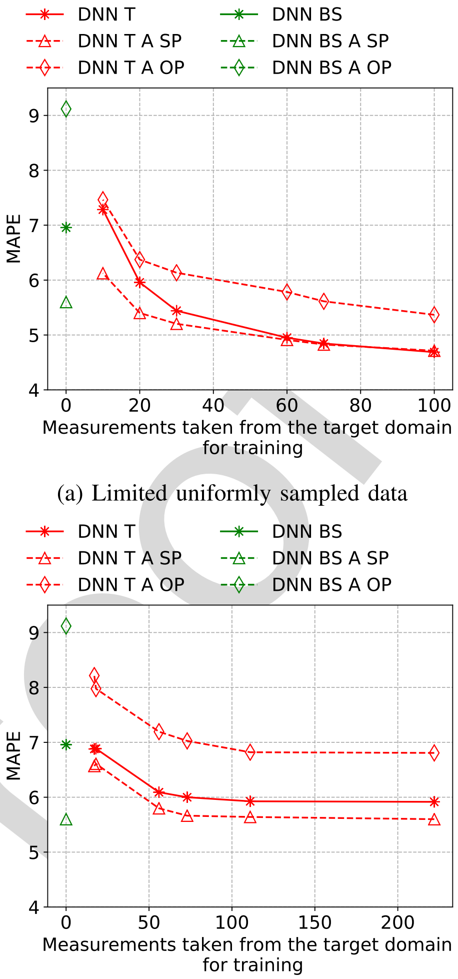
by a larger margin than with uniformly sampled data. This method is proven to be robust against the bias introduced between training and test sets in the target domain. As such, it performs extremely well when there is a large different between the training and testing sets in the target domain.

- In Figure 9(d), where points are taken with different probabilities over all distance ranges, the gains of using transfer learning are higher than in the case where the data is uniformly sampled (see Figure 8).

In conclusion, the proposed transfer learning approach is able to outperform other methods when using real measurements for both uniformly and non-uniformly sampled data. It has benefits compared to the benchmark methods in both of the following situations: (1) when the available samples of the target domain data are sampled uniformly, but the number of the samples is limited, and (2) when the available measurements used for training or fine tuning on the target domain are non-uniformly sampled. Moreover, considerable accuracy gains are achieved when augmenting the source domain with data coming from other available tilt configurations of the same antenna. This is discussed in the next section.

D. Tilt Augmentation Transfer

Data augmentation has been shown to be successful in the area of computer vision. By augmenting an existing dataset with new data that follows the same distribution as the data used for training, overfitting can be reduced [48]. In our case, we take inspiration from this idea and we augment the source domain by adding data from other available tilt configurations within the same PCI (i.e., suffix A SP on the graphs below) and from different PCIs (i.e., suffix A OP). We map the obtained MAPE to the DD to analyze the cases where data augmentation improves performance. Table III shows the DD between the training set in $\mathcal{D}_S$ and the test set in $\mathcal{D}_T$ both before data augmentation and after data augmentation. Data augmentation is performed by either adding data from the same PCI or adding data from the same and different PCIs. Table III also shows the total amount of training samples used in each case in $\mathcal{D}_S$. It can be noted that, DD values are much higher when using data from different PCIs than in the rest of the cases. This is because adding data from a different PCI will increase the difference between the training set in $\mathcal{D}_S$ and the test set in $\mathcal{D}_T$, therefore overfitting will be increased in $\mathcal{D}_T$. However, the degree of similarity between radio maps coming from the same PCI is higher, thus adding data with a greater similarity to the training set in $\mathcal{D}_S$ and test set in $\mathcal{D}_T$ can help to reduce overfitting and improve accuracy. We evaluate the gains of performing transfer learning from a bigger and more diverse source domain.



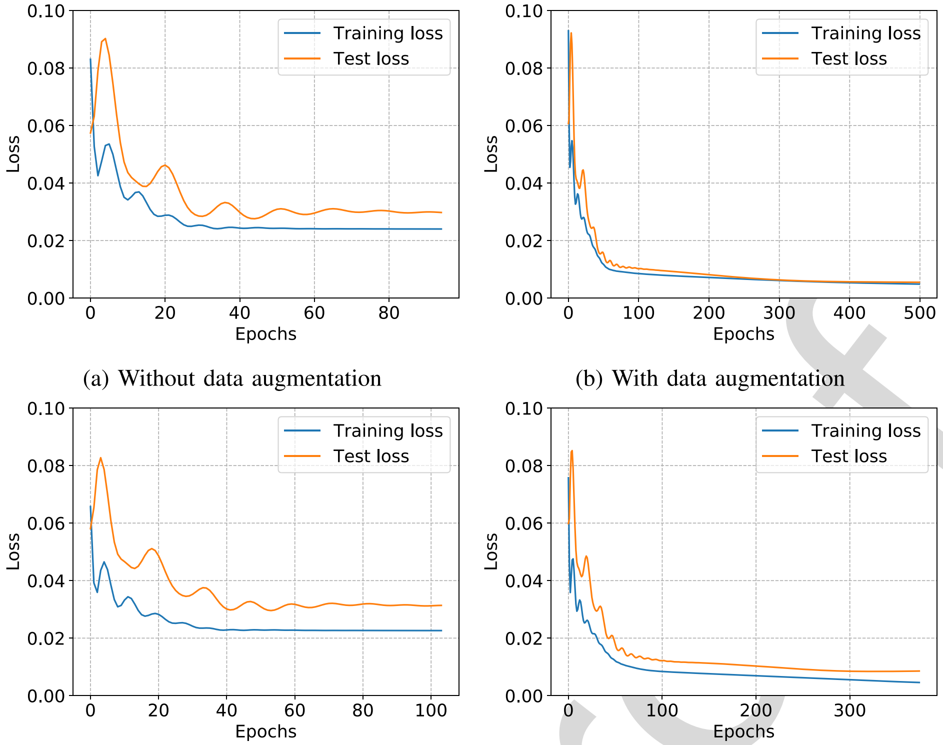
(a) Limited uniformly sampled data

(b) Limited non-uniformly sampled data

Fig. 10. MAPE with and without data augmentation.

Figure 10 shows the average MAPE across all the PCIs and pairs of training and testing tilt combinations possible when performing data augmentation on the source domain. Figure 10(a) shows the average MAPE when the instances taken from the target domain for training or fine tuning are limited and sampled uniformly. In contrast, Figure 10(b) shows the average MAPE for cases when the measurements taken from the target domain were collected at a distance range from the antenna between 300 and 600 m. We can draw the following conclusions:

- When using data augmentation on the source domain, the prediction error decreases by more than a 1% when the amount of instances taken from the target domain varies between 10 and 40 (see Figure 10(a)).
- In Figure 10(b) we can also observe a performance improvement when compared to the performance achieved without augmenting the source domain.
- In both cases, the performance improvement can be explained by the fact that data augmentation reduces overfitting. Figure 11 shows the training and cross validation curves for PCI 1 when Tilts 6 and 2 are used as source and target domains, respectively. Figures 11(a) and 11(b) illustrate the training and cross validation losses without



(a) Without data augmentation

(b) With data augmentation

(c) Without data augmentation

(d) With data augmentation

Fig. 11. Training curves on the target domain, PCI 1 and Tilts 6 and 2: (a), (b) Limited uniformly sampled data, (c), (d) Limited non-uniformly sampled data.

and with data augmentation, respectively, when a limited number of samples are taken uniformly. Figures 11(c) and 11(d) show the training and cross validation losses without and with data augmentation, respectively, when samples are taken at a distance between 300 and 600 m from the antenna location. In both cases, the gap between training and cross validation errors decreases when the source domain is augmented by adding data available from other tilt configurations, from the same PCI. This ensures the transfer learning model is less prone to overfitting.

- Adding data from different tilt configurations from different PCIs (i.e., DNN T A OP) does not lead to performance improvements. This is expected since the DD between the training set in $\mathcal{D}_S$ and the test set in $\mathcal{D}_T$ is much higher (see Table III, DD $= 3.53$) than in the rest of the cases (see Table III, DD $= 1.39$), therefore overfitting is more likely to happen.

VI. Conclusion

In this paper, we addressed the problem of predicting the signal strength in the downlink of a real LTE network, where the antennas can be tuned to operate with different antenna tilt configurations. Different approaches were considered as candidates for predicting the signal strength. All of them were based on refined features related to propagation and antenna configuration. As opposed to other works in the field of radio map inference, we studied the quality of prediction of the aforementioned approaches when the datasets used for training and testing are related, but not sampled from the same distribution. We observed that the performance of the predictive models is dependent on the amount of data taken from the testing domain for training or fine tuning. Furthermore, the proposed transfer learning algorithms are shown to be more efficient in cases where the amount of data available from the target tilt configuration is very limited, or available at different distance ranges from the antenna location. Finally, we have shown how augmenting data from the source domain by adding data available from other tilts configurations of the same antenna improves the performance of the proposed transfer learning approaches. Augmenting the source domain decreases the prediction error by 1% when the data available from the target domain for training or fine tuning is limited, or at a distance range between 300 and 600 m from the antenna location.

References

[1] C. Parera, A. E. C. Redondi, M. Cesana, Q. Liao, L. Ewe, and C. Tatino, "Transferring knowledge for tilt-dependent radio map prediction," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2018, pp. 1–6.

[2] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Trans. Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[3] M. Danneberg, J. Holfeld, M. Grieger, M. Amro, and G. Fettweis, "Field trial evaluation of UE specific antenna downtilt in an LTE downlink," in *Proc. IEEE Int. ITG Workshop Smart Antennas (WSA)*, 2012, pp. 274–280.

[4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Ch. Assoc. Comput. Linguist. Human Lang. Technol.*, 2019, pp. 4171–4186.

[6] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, 2018, pp. 328–339.

[7] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, "A mobile network planning tool based on data analytics," *Mobile Inf. Syst.*, vol. 2017, Nov. 2017, Art. no. 6740585.

[8] H. C. Nguyen *et al.*, "Validation of tilt gain under realistic path loss model and network scenario," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, 2013, pp. 1–5.

[9] C. Phillips, D. Sicker, and D. Grunwald, "A survey of wireless path loss prediction and coverage mapping methods," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 255–270, 1st Quart., 2013.

[10] I. Rodriguez *et al.*, "A geometrical-based vertical gain correction for signal strength prediction of downtilted base station antennas in urban areas," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, 2012, pp. 1–5.

[11] D. W. Kifle, B. Wegmann, I. Viering, and A. Klein, "Impact of antenna tilting on propagation shadowing model," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, 2013, pp. 1–5.

[12] "Evolved universal terrestrial radio access (E-UTRA); Further advancements for (E-UTRA) physical layer aspects," 3GPP, Sophia Antipolis, France, Rep. TR 36.814, 2006.

[13] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, 2016.

[14] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, no. 7, pp. 1633–1685, 2009.

[15] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguist.*, 2007, pp. 264–271.

[16] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Multisource domain adaptation and its application to early detection of fatigue," *ACM Trans. Knowl. Disc. Data*, vol. 6, no. 4, p. 18, 2012.

[17] L. Duan, D. Xu, and I. W.-H. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.

[18] S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, "Learning a meta-level prior for feature relevance from multiple related tasks," in *Proc. ACM 24th Int. Conf. Mach. Learn.*, 2007, pp. 489–496.

[19] M. Long, J. Wang, G. Ding, S. J. Pan, and S. Y. Philip, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[20] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*, 2008, pp. 283–291.

[21] T. Tommasi, F. Orabona, and B. Caputo, "Learning categories from few examples with multi model knowledge transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 928–941, May 2014.

[22] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising Markov logic networks for transfer learning," in *Proc. AAAI 22nd Nat. Conf. Artif. Intell.*, 2007, pp. 608–614.

[23] J. Davis and P. Domingos, "Deep transfer via second-order Markov logic," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 217–224.

[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1717–1724.

[25] D. C. Cireşan, U. Meier, and J. Schmidhuber, "Transfer learning for Latin and Chinese characters with deep neural networks," in *Proc. IEEE Int. Joint Conf. Neur. Netw. (IJCNN)*, 2012, pp. 1–6.

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

[27] S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan, "Adaptive localization in a dynamic WiFi environment through multi-view learning," in *Proc. 22nd Nat. Conf. Artif. Intell.*, 2007, pp. 1108–1113.

[28] V. W. Zheng, S. J. Pan, Q. Yang, and J. J. Pan, "Transferring multi-device localization models using latent multi-task learning," in *Proc. AAAI*, vol. 8, 2008, pp. 1427–1432.

[29] S. J. Pan, D. Shen, Q. Yang, and J. T. Kwok, "Transferring localization models across space," in *Proc. 23rd Nat. Conf. Artif. Intell.*, 2008, pp. 1383–1388.

[30] J. Pan, "Feature-based transfer learning with real-world applications," Ph.D. dissertation, Dept. Comput. Sci. Eng., Hong Kong Univ. Sci. Technol., Hong Kong, 2010.

[31] E. Baştuğ, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," in *Proc. 13th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, 2015, pp. 161–166.

[32] A. Galindo-Serrano, L. Giupponi, and G. Auer, "Distributed learning in multiuser OFDMA femtocell networks," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, 2011, pp. 1–6.

[33] W. Wang, J. Zhang, and Q. Zhang, "Transfer learning based diagnosis for configuration troubleshooting in self-organizing femtocell networks," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, 2011, pp. 1–5.

[34] M. Chen, W. Saad, C. Yin, and M. Debbah, "Data correlation-aware resource management in wireless virtual reality (VR): An echo state transfer learning approach," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4267–4280, Jun. 2019.

[35] Z. Tang, D. Wang, and Z. Zhang, "Recurrent neural network training with dark knowledge transfer," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2016, pp. 5900–5904.

[36] E. U. T. R. Access, "Physical layer-measurements (3gpp ts 36.214 version 9.0. 0 release 9)," Standard ETSI TS 136 214, 2010.

[37] F. Afroz, R. Subramanian, R. Heidary, K. Sandrasegaran, and S. Ahmed, "SINR, RSRP, RSSI and RSRQ measurements in long term evolution networks," *Int. J. Wireless Mobile Netw.*, vol. 7, no. 4, pp. 113–123, 2015.

[38] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, U.K.: MIT Press, 2016.

[39] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. 25th Int. Conf. Neur. Inf. Process. Sys.*, 2012, pp. 2951–2959.

[40] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, 2012.

[41] F. Chollet *et al.*, (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras

[42] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Conf. Oper. Syst. Design Implement.*, 2016, pp. 265–283.

[43] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[44] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, 1995, pp. 278–282.

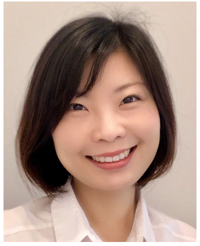
[45] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[46] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[47] J. M. Joyce, "Kullback-Leibler divergence," in *International Encyclopedia of Statistical Science*. Heidelberg, Germany: Springer, 2011, pp. 720–722.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neur. Inf. Process. Syst.*, 2012, pp. 1097–1105.

Claudia Parera received the B.S. degree in computer science from the University of Havana, Cuba, in 2010, and the M.S. degree in advanced computer science from the University of Bradford, U.K., in 2016. She is currently pursuing the Ph.D. degree from the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano. She is an Early State Researcher in the Anticipatory Networking Techniques in 5G Networks and Beyond (ACT5G) Marie Curie project. As part of the current Ph.D. degree, she is being actively involved in academic and industrial research with Nokia Bell Labs, Stuttgart. Her research interests include applying machine and transfer learning techniques to network analytics, especially in the field of anticipatory networking for 5G networks.



Qi Liao received the M.S. degree in E.E. and the Dr.-Ing. degree from Heinrich-Hertz-Chair for Information Theory and Theoretical Information Technology, Technical University of Berlin in 2010 and 2016, respectively. From 2010 to 2013, she was a Research Associate with the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin. From 2013 to 2014, she was the Ph.D. intern with the Department of Statistics and Learning Research, Bell Labs, Murray Hill, NJ, USA. Since 2015, she has been a Research Scientist with the E2E Mobile Network Solutions Lab, Nokia Bell Labs, Stuttgart, Germany. She holds more than 50 peer reviewed journal articles, conference papers, and granted or filed patents. Her current research interests include multiobjective optimizations, optimization for multiagent systems, resource allocation, stochastic optimization, and machine learning techniques.

Ilaria Malanchini received the B.S. and M.S. degrees in telecommunications engineering from the Politecnico di Milano, Italy, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from Drexel University, Philadelphia, and Politecnico di Milano in 2011. She is a Senior Research Engineer with E2E Network & Service Automation Laboratory and has been with Bell Labs Stuttgart since 2012. She was awarded the Meucci-Marconi Award and the Chorafas Foundation Prize for her Master's and Ph.D. thesis, respectively. She has published more than 25 peer reviewed journal and conference papers and has more than 10 granted or filed patents. Her research interests focus on optimization models, mathematical programming, game theory, and machine learning, with the application of these techniques to wireless network problems, such as wireless resource allocation, anticipatory network optimization, infrastructure and resource sharing, and network slicing.

Cristian Tatino received the B.S. and M.S. degrees in telecommunications engineering from the Universitá di Napoli Federico II, Italy, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree at the Communications and Transport Systems Division, Department of Science and Technology, Linköping University, Sweden, where he is currently an Early State Researcher within the ACT5G Marie Curie project. From 2013 to 2015, he worked as a System Engineer on wireless telecommunications networks for railways application. He has been a Visiting Fellow with Nokia Bell Labs, Stuttgart, Germany, in 2016 and 2018. He focuses his research on the wireless link status anticipation for millimeter-waves wireless networks. In particular, he studied the impact of the signal reflections on the coverage probability for non-line-of-sight communications, multiconnectivity, and relaying solutions for communication reliability.

Alessandro E. C. Redondi received the M.S. degree in computer engineering and the Ph.D. degree in information engineering from Politecnico di Milano, Italy, in July 2009 and February 2014, respectively, where he is currently an Assistant Professor with the Dipartimento di Elettronica, Informazione e Bioingegneria. From September 2012 to April 2013, he was a visiting student with the EEE Department, University College of London. His research activities are focused on the design and optimization of IoT systems and on network data analytics.

Matteo Cesana received the M.S. degree in telecommunications engineering and the Ph.D. degree in information engineering from the Politecnico di Milano, Italy, in July 2000 and in September 2004, respectively, where he is a Full Professor with the Dipartimento di Elettronica, Informazione e Bioingegneria. From September 2002 to March 2003, he was a Visiting Researcher with the Computer Science Department, University of California at Los Angeles (UCLA). His research activities are in the field of design, optimization, and performance evaluation of wireless networks with a specific focus on communication technologies for the Internet of Things and Future Generation Cellular Networks. He is an Associate Editor of the *Ad Hoc Networks Journal* (Elsevier).