UNIVERSITY OF CALIFORNIA,
IRVINE

Mobile Coverage Maps Prediction

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Networked Systems

by

Emmanouil Alimpertis

DISSERTATION Committee:
Professor and Chancellor's Fellow Athina Markopoulou, Chair
Professor Carter T. Butts
Doctor Ioannis Broustis

2020

DEDICATION

In memory of my father, John,
who gave me the greatest gift anyone could give another person: He believed in me.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I feel extremely lucky for doing my Ph.D. at the Networked Systems program of UC Irvine, and for having Professor Athina Markopoulou as my advisor. I would like to express my deep gratitude to Professor Markopoulou for her support and her trust throughout the years. She allowed me to pursue my research interests and this work would not have been possible without her guidance. She helped me develop the necessary skill of being critical of my own work, she always motivated me to sharpen my thinking and she helped me improve my (very) rough drafts, for which I am forever thankful. I am also especially indebted to Professor Carter T. Butts; thank you from the bottom of my heart for formalizing many of my floating and abstract ideas as well as for providing many more novel and elegant approaches to consider in my dissertation. A special thanks also goes to Dr. Ioannis Broustis; working with him was a stepping stone to success and developing my skills. The real world experience and understanding of cellular systems I gained under his guidance and mentorship during my AT&T Internships were crucial for the completion of this thesis.

Working on coverage maps prediction requires access to real-world datasets. In this thesis, I utilized two types of datasets. The first, was collected using a tool developed as part of the thesis on the UC Irvine campus, and referred to as “Campus Dataset”. I would like to thank my lab mates who contributed their mobile phones for collecting data across campus; without you, I would not be able to progress on my research. The second type of datasets were generously provided by Tutela Technologies, referred to throughout the thesis as the “mobile analytics/crowdsourcing company”. These include two city-scale datasets in the metropolitan areas of NYC and LA, and they are among the largest studied in the literature. Their scale and quality were instrumental in obtaining valuable insights and provided unique opportunities to study prediction of mobile coverage maps. I would like to wholeheartedly thank Tutela in general, and Hunter Macdonald in particular, for allowing me to study their datasets, which was a key asset for this thesis.

I am grateful to all of those with whom I have had the pleasure to collaborate with in the past; Dr. Blerim Cici and Dr. Minas Gjoka helped me decisively in my first steps in my PhD; Prof. Aggelos Bletsas, Prof. Michail Lagoudakis and Prof. Polichronis Koutsakis introduced me to research and their guidance led me to pursue a PhD degree. I would also like to offer my special thanks to the members of the UCI Networking Group, Evita Zoi Bakopoulou, Hieu Lee, Anastasia Shuba, Milad Asgari, Janus Varmarken and Balint Tillman; my years in UCI would not have been as productive without you.

I have also learned much from the mentors, collaborators, and friends with whom I have crossed paths during my internships in industry: Josh Graessley, Tommy Pauly, Eric Kinnear, Scott Herscher, Dr. Ioannis Broustis (he deserves a second ack) and Brad Zeschuk; thank you for showing me the real world and the impact our research can have in it.

On a personal note, I would like to express my appreciation to all the friends and the special connection I made with them while in UCI. Dimitris, thank you for your support in some of the most difficult moments of my life so far and being there with all the 8PM messages,

which were the signal for our nightly outings. Evita, we just shared for 3 months the same cubicle because of COVID19, but regardless our friendship helped me to cope with the PhD grad life; we will eventually co-author papers apart from the various other drafts we have already collaborated for. Anna, a thank you cannot describe my appreciation for all the exploration and ski outings we did; our friendship made my life in UCI much better.

There are also two other special people whose friendship the last 13 years has changed my life to better and has motivated me to unlock and achieve so many goals. Nikos Kofinas (Kofi) and Nikos Pavlakis (NP), thank you again for everything; there are not words to describe my gratitude for our friendship. When my phone rings (when there is mobile coverage), the majority predictor would yield Kofi's name; these calls have made a lot of difficult PhD days to feel better. Niko (NP), you have always helped me in so many different ways that I would need a separate chapter to list them all; I do miss the days we were neighbors in Chania.

Other people whose friendship has supported me throughout these years and I would not be able to complete my thesis without: Nikos Fasarakis-Hilliard who has been next to me when I mostly needed both academically and personally, Dimitris Iliou and the amazing experiences we have shared, Dr. Giannis Demertzis, Dr. Sofia Maria Nikolakaki, Nikoleta Kassapaki and Ioanna Kaza. Thank you all. And of course a big thanks to my family; I would not be here without you.

I have greatly appreciated all the funding support from the University of California, Irvine through the Networked Systems Fellowship and the UCI Electrical Engineering Fellowship sponsored by Broadcom Foundation; the National Science Foundation Awards 1649372, 1526736, 1939237, 1900654; and AT&T Research Labs, M2Catalyst and Apple for the summer internships. Last but not least, I would like to thank the Association for Computing Machinery (ACM) for granting me permission to disclose copyrighted materials presented on my publications.

CURRICULUM VITAE

Emmanouil Alimpertis

EDUCATION

Doctor of Philosophy in Networked Systems **2020**
University of California, Irvine *California, USA*

Master of Science in Networked Systems **2020**
University of California, Irvine *California, USA*

Master of Science in Electronic and Computer Engineering **2014**
Technical University of Crete *Chania, Greece*

Engineering Diploma in Electronic and Computer Engineering **2012**
Technical University of Crete *Chania, Greece*

RESEARCH EXPERIENCE

Graduate Research Assistant **2014–2020**
University of California, Irvine *Irvine, California*

Graduate Research Assistant **2012–2014**
Technical University of Crete *Chania, Greece*

REFEREED CONFERENCE PUBLICATIONS

Emmanouil Alimpertis, Athina Markopoulou, Carter T. Butts and Konstantinos Psounus. **City-Wide Signal Strength Maps: Prediction with Random Forests**. In *Proceedings of the ACM World Wide Web Conference, 2019.*

Blerim Cici, Emmanouil Alimpertis, Alexander Ihler and Athina Markopoulou. **Cell-to-cell activity prediction for smart cities**. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016.*

REFEREED POSTER PRESENTATIONS

Emmanouil Alimpertis and Athina Markopoulou. **A system for crowdsourcing passive mobile network measurements**. In *Poster session of the 14th USENIX NSDI Conference, 2017.*

TEACHING EXPERIENCE

Teaching Assistant - Telecommunications Systems II **2013**
Technical University of Crete *Chania, Greece*

Teaching Assistant - Telecommunications Systems II **2012**
Technical University of Crete *Chania, Greece*

PROFESSIONAL EXPERIENCE

Software Engineer Intern **2019**
Apple *Cupertino, California*

Student Summer Intern **2017**
AT&T Labs Research *Bedminster, New Jersey*

Data Science Intern **2016**
M2Catalyst *Alisso Viejo, California*

Student Summer Intern **2015**
AT&T Labs Research *Bedminster, New Jersey*

ABSTRACT OF THE DISSERTATION

Mobile Coverage Maps Prediction

By

Emmanouil Alimpertis

Doctor of Philosophy in Networked Systems

University of California, Irvine, 2020

Professor and Chancellor's Fellow Athina Markopoulou, Chair

Mobile coverage maps consist of various key performance indicators such as the received signal signal strength levels per location, and are of great importance to cellular operators. However they are expensive to obtain, incomplete or inaccurate in some locations, imperfectly reflective of call quality outcomes and potentially constructed from biased samples. In this dissertation, we develop a principled machine learning framework for predicting missing values of mobile coverage maps. It provides the knobs for operators to express their objectives and preferences, as well as tools for data valuation.

First, we develop a prediction framework based on random forests (RFs) to improve signal strength maps from limited measurements. The proposed RFs-based predictor utilizes a rich set of features including but not limited to location, time, cell ID and device hardware, which are considered jointly for the first time. We show that our RFs-based predictor can significantly improve the tradeoff between prediction error and number of measurements needed compared to state-of-the-art data-driven predictors, *i.e.,* requiring 80% less measurements for the same prediction accuracy, or reduces the relative error by 17% for the same number of measurements.

Second, we extend the framework beyond signal strength and mean square error (MSE) minimization to provide knobs to operators to (i) optimize prediction for coverage maps quality outcomes such as coverage indicators and call drop probability; and (ii) deal with sampling bias. We show that we can improve the relative error for the call drop probability up to 32% in the high CDP regime of greatest concern to cellular operators, which corresponds to improvement of signal strength prediction itself in its low values regime. Similarly, we improve recall from 76% to 92% for predictions of coverage loss, where false negatives are costly to operators. We also introduce weight functions that allow operators to specify which points are more important to predict accurately. We propose a reweighting scheme to obtain unbiased error metrics in settings for which the available signal strength data is not sampled proportionally to the target distribution of interest. We demonstrate a benefit of up to 20% of training models with reweighted errors for two intuitive cases: (i) uniform loss with respect to spatial area; and (ii) loss proportional to user population density. Combining both techniques shows improvement up to 5%.

Third, we apply, for the first time, the notion of data Shapley valuation in the context of mobile coverage maps prediction. We demonstrate data valuation for various operators metrics and we show how our reweighted errors fit naturally the data Shapley framework. Assessing the data Shapley values of training data points enables improving prediction, data minimization, and pricing of mobile data. For instance, we are able to remove up to 65% of the low valued training data points and simultaneously improve the recall of coverage loss from 64% to 99%.

Throughout this thesis, we leverage two types of real-world mobile (LTE) datasets to evaluate our methods and gain valuable insights: the first was collected at our university campus by an android App we developed and the second provided by a mobile crowdsourcing company for NYC and LA metropolitan areas, including approximately 11 million measurements. Our work can be useful for mobile analytics companies and cellular operators, particularly in the context of the upcoming 5G deployments.

Chapter 1

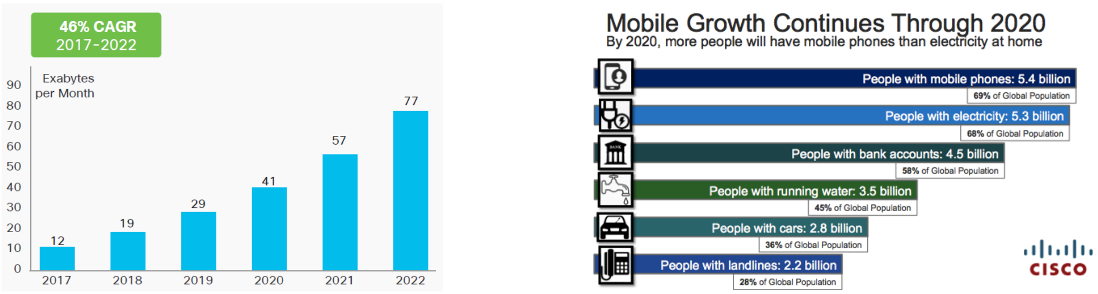
Introduction

1.1 Motivation

1.1.1 Mobile Is King

Cellular mobile telephony (*e.g.,* 2G/3G and the newer 4G LTE/LTE-A) is used by approximately 5 billion unique subscribers [64, 23], as of 2020, indicating great success of the relevant technologies. Mobile phones are ubiquitous: although there are currently 5 billion mobile phone users, only 3.4 billion people have access to running water [23] (see Fig. 1.1b). Furthermore, the global mobile data traffic has been increasing exponentially and is expected to reach 77.5 exabytes per month by 2022 [23] (see Fig. 1.1a).

The basic architecture of the network remains essentially the same across the cellular generations. Cellular networks are built using a set of Base Transceiver Stations (BTS or simply BS) that are in charge of communicating with mobile devices. A mobile phone (*a.k.a.* user equipment, UE) is connected (attached) to a unique cell (offered by a BS) in the area of coverage at a time, and phone calls initiated by individuals are being routed through that



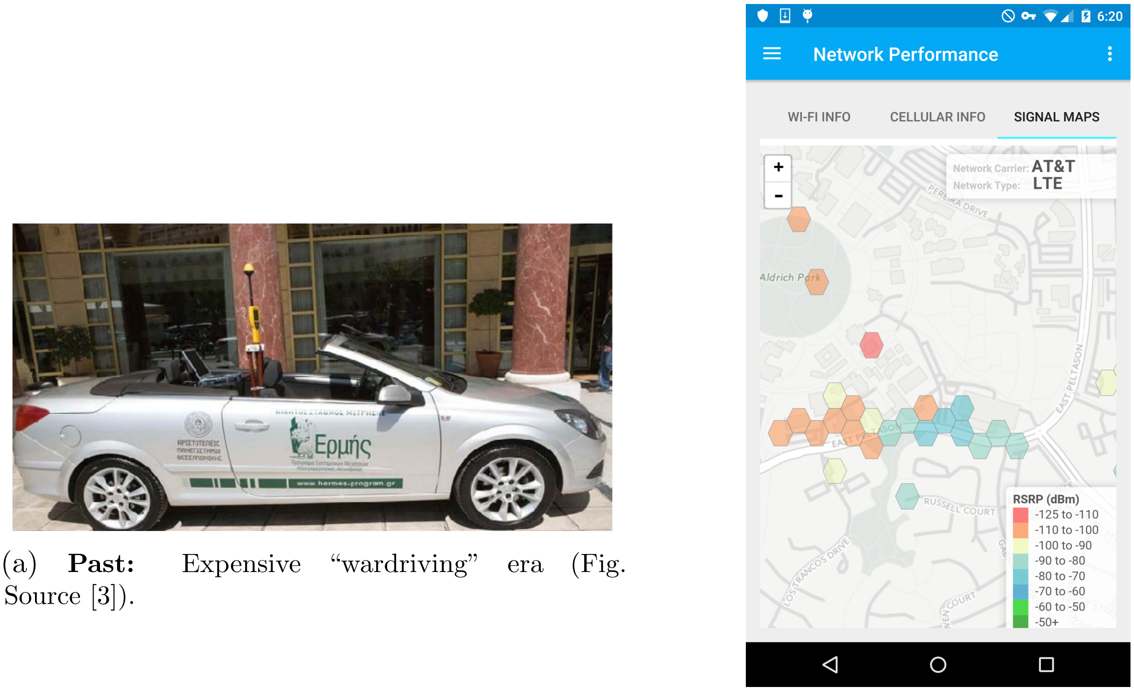
(a) Exponential Global Mobile Data Traffic Growth.

(b) More People with Mobile than Running Water.

Figure 1.1: Mobile is King: (a) The exponential increase of mobile data traffic (b) and the ubiquitousness of mobile phones *vs.* other essential goods. Source: Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022.

BS. The received signal power levels (*a.k.a.* received signal strength, RSS) of the wireless connection between the BS and the UE at each geographical region, define the quality of service. The RSS from the serving BS in a UE, is represented by the familiar signal bars on our mobiles' screen; in popular culture the frustration with a dropped call usually refers to "low bars". At any given moment, one or more BSs can provide coverage to the mobile phones, therefore, a UE is usually assigned to a BS (cell selection) with the strongest RSS (among other criteria). In a nutshell, received signal strength measurements are utilized by mobiles for cell selection, handovers decisions (*i.e.,* change from a cell to another), mobility measurements and numerous other network operations.

Thus, received signal strength is a fundamental property of mobile connectivity and cellular operators rely heavily on such key performance indicators (KPIs) to understand the performance and coverage of their network, as well as that of their competitors, in their efforts to provide the best user experience. KPIs usually include several wireless channel measurements (*e.g.,* RSS in the older GSM or channel quality indicator, CQI, and reference signal received power, RSRP, in LTE) as well as other performance metrics (*e.g.,* throughput, delay, jitter) and other information associated with the measurement (*e.g.,* frequency band, location of receiver, time, *etc.*). Mobile coverage maps, which indicate the level of service per location, are constructed from a large number of KPI measurements; a representative



(a) **Past:** Expensive "wardriving" era (Fig. Source [3]).

(b) **Present:** "Dataism" era; a smartphone monitors and records its signal strength map. An example from the coverage (signal) maps collection tools built in this thesis.

Figure 1.2: Methods for crowdsourcing coverage maps data, present *vs.* past.

example of a mobile coverage map is shown in Fig. 1.4. They are of crucial importance to cellular providers, for network management, maintenance, upgrades and for commercial advertising their network.

1.1.2 Dataism Meets Cellular and 5G

Traditionally, one way that operators obtain detailed and accurate measurements was by hiring dedicated vans with special equipment, to drive through, measure and map the received signal strength (RSS) in a particular area of interest, a technique referred to as "wardriving" [73]. However, this method is expensive, inherently limited and it usually requires costly radio spectrum equipment; it cannot scale and provide large scale (city- or country-wide) measurements. Nonetheless, collecting traces through wardriving and using them as input to complicated wireless propagation models such as COST-231 Walfisch-Ikegami [25] or others [76, 16] was the only option available to create coverage maps back in the 1990s and early 2000s.

In the last decade though, the rapid emergence of smartphones has enabled a significant shift in the efforts for mapping the cellular network coverage. Nowadays, a smartphone can be a mobile coverage sensor itself [3] since it inherently monitors its signal strength and, most importantly, it is equipped with a GPS module allowing granular location tracking. Moreover, there is an abundance of other sensors (*e.g.,* accelerometer and others) and system level APIs which can provide rich contextual information (*e.g.,* time, speed and altitude, low-level network information, *etc.*). Basically, the coverage maps have inevitably followed the trend of dataism[1], where "information flow" is essential. This trend includes several efforts from the research community (*e.g.,* crowdsourcing systems with large public datasets for coverage maps from iPhone devices [3] with an example depicted in Fig. 1.3 or Android devices [53]), as well as from the industry with publicly available network performance monitoring apps from mobile analytics companies, such as Tutela [70], OpenSignal [55], RootMetrics [62, 63] *etc.*, who aim to monetize the collected data.

Although operators can collect measurements on the network edge themselves, they increasingly choose to outsource the collection of data for mobile coverage maps to the aforementioned third parties for a variety of reasons, including: cost, liability related to privacy concerns of collecting data on end-user devices, and lack of access to competitor networks. This practice of operators to buy signal map data from these specialized mobile analytics companies, has created a huge market for the mobile coverage performance analysis. These companies crowdsource measurements directly from end-user devices, via standalone mobile apps [55], or measurement SDKs [70] integrated into popular partnering apps, typically games, utilities

[1] "Dataism" is a term that has been used to describe the mindset or philosophy created by the emerging significance of Big Data. It was first used by David Brooks in the New York Times in 2013. The term was popularized by the book "Homo Deus" from Yuval Noah Harari, where it has been expanded to describe an emerging ideology or even a new form of religion, in which "information flow" is the "supreme value".

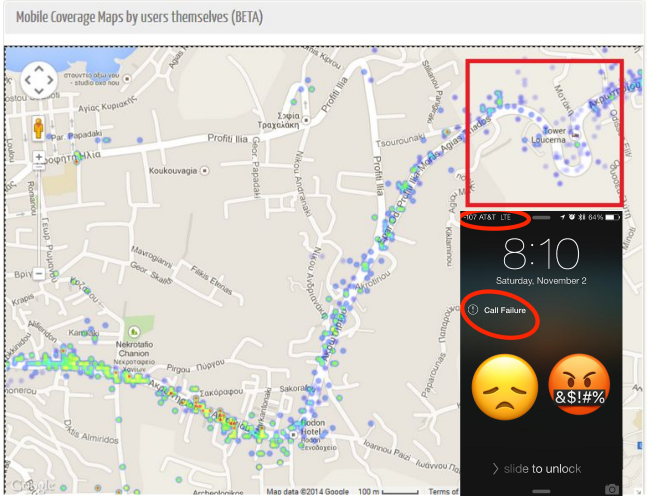
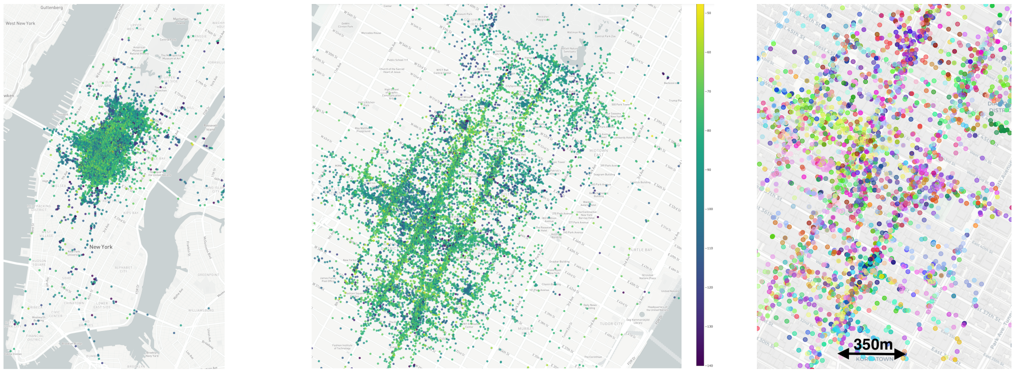


Figure 1.3: Another example of a received signal strength (*i.e.,* coverage) map from our past work in [3] with data automatically collected by several phones.(i) It can be clearly seen, that there is a coverage hole in the commute trace with consistent dropped calls. (ii) It can be seen that the rest of the area has not been sampled since users just cross the area by driving. This could lead to biased sampling which is being addressed in Chapter 5.

or streaming apps. This way, they crowdsource measurements at large (city, country, or world-wide) scale and over long periods of time, but the measurements can be sparse in space (depending on end-user location) and time (measurements are collected infrequently so as to not drain user resources, such as battery or cellular data).

What if data are missing and/or are expensive? Coverage maps (*a.k.a.* signal strength maps) are expensive for both carriers (paying millions to third parties to collect data) and crowdsourcing companies (most of which use cloud services, thus collecting more measurements increases their operational cost). Yet, collecting a large number of measurements is necessary to obtain good accuracy and spatial completeness for signal maps. Current technology and application trends, such as (i) 5G dense deployment of small cells with network virtualization [10] and (ii) smart city/IoT monitoring and control at metropolitan scales, will only increase the need for accurate performance measurements [36, 41, 13]. The problem is only exacerbated by the fact that data may be sparse, unavailable, or expensive to obtain in some locations, times, frequencies other parameters of interest.



(a) NYC Manhattan subset. (b) NYC: LTE RSRP map. (c) NYC Manhattan zoomed in.

Figure 1.4: LTE RSRP map examples from NYC `dataset`, for a representative group of LTE cells in the Manhattan Midtown area; there are also millions of other data points for other NYC neighborhoods. (a-b) Display LTE RSRP (*i.e.,* signal strength). (c) Different colors indicate measurements from different cells.

Coverage maps prediction. Thus, our goal in this thesis is to develop a principled machine learning framework to predict values for mobile coverage maps in order to fill the gaps in space and other features of interest, as well as to predict coverage maps optimized for objectives and error metrics of interest for cellular operators. Moreover, the prediction framework we develop, it also assigns data valuation to our measurements.

1.1.3 Technical Limitations

Broadly speaking, signal strength prediction can be done through propagation models or data-driven approaches, including geospatial interpolation, [33, 59] and Machine Learning [29]. We identify several limitations of prior work, which will be addressed by this thesis.

A. Limited set of features, limited scale of data and not readily available features. *Propagation models:* State-of-the-art wireless propagation and path loss (equation-based) models include WINNER I/II [16], Ray tracing [76] and many others. However, this family of models requires a detailed map of the environment (*e.g.,* topology, street width, antennas' heights, number of floors, in some cases 3D maps, *etc.*) and fine-grained tuning of parameters or a vast number of measurements [5].

Geospatial interpolation: [19, 59, 33]. This family of predictors (*e.g.,* Ordinary Kriging) cannot naturally incorporate additional dimensions such as time, frequency, hardware and network information and they are inherently limited to spatial features. Moreover, extra preprocessing is required to identify sub-regions with similar radio propagation characteristics.

Machine learning: Prior work that has used ML for RSS modeling for localization, has only focused on spatial features (*e.g.,* measurements' latitude and longitude in [61]) or it has used detailed 3D maps from Light and Range Detection Data (LiDAR) for RSRP prediction [29]. However, these specialized data are not readily available from smartphones measurements.

B. Minimizing solely the Mean Squared Error (MSE). To the best of our knowledge, prior work has focused on predicting the raw signal strength itself, minimizing solely the mean square error (MSE) (*e.g.,* [29, 19]), which does not necessarily map directly to cellular operators' objectives. First, the operator may be more interested in predicting *quality functions* (such as the number of signal bars and the call drop probability), which depend on but are different from measurable KPIs. For instance, the operator may be more interested in predicting accurately good *vs.* poor coverage than in minimizing the MSE of signal strength. Second, an operator, may be interested in some data points more than others, *e.g.,* locations chosen uniformly at random, locations with dense user population, or specific locations of interest (*e.g.,* to 911 dispatchers, to beat competition *etc.*), while relying on sampling distributions that are different from target distributions. For example, both in smaller studies (Fig. 1.3) and larger scale data collection (Fig. 1.4) we can clearly see that the data are primarily collected while commuting and in a lesser extent to the residential blocks.

C. Absence of Data Valuation tools. Although there have been significant ML developments in the last years, only recent literature addressed valuation of training data points

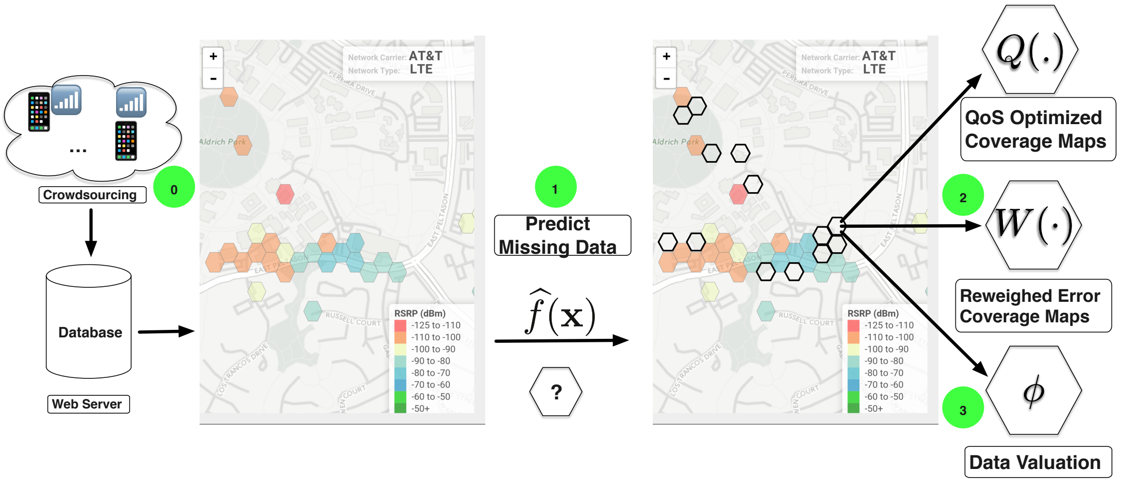


Figure 1.5: Thesis Overview. We develop a principled machine learning framework to **(1)** predict missing value of mobile coverage maps (Chapter 4), with **(2)** mobile coverage maps optimized beyond the standard MSE in order to match the mismatch between (i) the cellular operators' quality (QoS) outcomes of interest and the raw signal strength and (ii) sampling and target distributions (Chapter 5). Finally, this thesis offers **(3)** data valuation for mobile coverage maps with data Shapley (Chapter 6). Throughout this thesis, we needed realistic mobile datasets; in Chapter 3 we developed tools for collecting such data ourselves.

in the context of medical tasks classification [35], but not in the context of mobile maps.

1.2 Overview and Thesis Contributions

In this thesis, we propose a principled machine learning framework for predicting missing values of mobile coverage maps, at particular spatiotemporal points and potentially considering other features as well. Our goal is to improve the existing tradeoff between cost (*i.e.*, number of measurements) and quality (*i.e.*, accuracy) of signal strength maps prediction. We also provide operators with a framework with knobs to tackle the mismatch between (1) operators quality functions and raw signal strength as well as (2) sampling and target distributions. More specifically, we make the following contributions, summarized also in Fig. 1.5.

1.2.1 City-Wide Coverage Maps: Prediction with Random Forests

In Chapter 4, we develop a powerful machine learning framework based on random-forests (`RFs`), considering a rich set of features including, but not limited to, location, time, cell ID, device hardware, distance from the tower, frequency band, and outdoors/indoors location of the receiver, all of which affect the wireless properties. To the best of our knowledge, this is the first time that location, time, device and network information are considered jointly for the problem of signal strength prediction compared to geospatial prediction [50, 19, 59], which does not naturally extend beyond location features. We show that our `RFs`-based predictors can significantly improve the tradeoff between prediction error and number of measurements needed, compared to state-of-the-art data-driven predictors. They can achieve the lowest error of these baselines with 80% less measurements; or they can reduce the $RMSE$ (root mean square error) by 17% for the same number of measurements. In absolute terms, we demonstrate improvements up to 2dB.

1.2.2 Quality and Weight Functions for Mobile Coverage

In Chapter 5, we extend further our framework to handle the two limitations introduced by solely minimizing the mean squared error (MSE) and we provide cellular operators (and mobile analytics companies) with knobs to tackle the mismatch between (1) operators' quality functions and raw signal strength as well as (2) sampling and target distributions. We built on the predictor we previously developed based on Random Forests, but our techniques are applicable to any arbitrary ML model.

First, we identify *quality functions* based on signal strength, such as mobile coverage indicators and call drop probability (CDP), which are not directly optimized by learning on signal strength. While prior work minimizes only the MSE for signal strength (*e.g.,* [29]), we train models directly on these functions and we show that we can improve the relative error up to 32% (or *alternatively* improve signal strength prediction itself by up to 3dB) in the high CDP regime of greatest concern to operators and recall from 76% to 92% for predictions of coverage loss (where false negatives are costly to operators). Our methodology optimizes directly the function of interest and allows operators to put more emphasis in the values and use cases of signal maps that matter most.

Second, we introduce *weight functions* that can express the importance operators give to particular locations or data. This reweighting is rooted at the framework of importance sampling and allows us to obtain unbiased error metrics in settings for which the available data is not sampled proportionally to the target distribution of interest (*a.k.a.* dataset shifting problem [67]). We demonstrate two intuitive weight function classes, respectively encoding (i) uniform loss with respect to spatial area; and (ii) loss proportional to user population density. Training models with reweighted errors shows an average improvement of 5% and up to 20% for oversampled regions. Combining both techniques shows improvement up to 5.5% for the estimation of CDP adjusted with population and uniform distributions.

1.2.3 Data Shapley Valuation for Mobile Coverage Prediction

In Chapter 6, we apply, for the first time, the problem of data Shapley valuation for mobile coverage maps. We build on and extend the framework provided by [35] with our custom error metrics from Chapter 5, and we obtain the value of a each training point for a particular prediction algorithm, error metric and dataset. We analyze the distribution of data Shapley values in our datasets and we apply it for improving prediction and for data minimization.

We define jointly a specific prediction task and the performance-error metric of interest under the umbrella of data Shapley in order to quantify the data valuation. We demonstrate data valuation for various operators metrics instead of the standard accuracy and MSE in classification and regression respectively and we also show how our reweighted errors fit naturally the data Shapley framework. Assessing the data Shapley values of training data points enables improving prediction, data minimization, and pricing of mobile data. For instance, we are able to remove up to 65% of the low valued training data points and simultaneously improve the recall of coverage loss from 64% to 99%.

1.2.4 Datasets

In order to study the problems in this thesis, we needed realistic mobile traces. We used two such real world LTE datasets: one collected by ourselves and one provided by a crowdsourcing company, both presented in Chapter 3. First, we built a real world crowdsourcing system, which consists of an Android app and a central server for gathering the data (see Chapter 3). We used this system to collect a small but dense `Campus dataset`, in the area of University of California, Irvine campus. The second dataset consists of a large but sparser set of measurements from `NYC and LA` metropolitan areas, provided by a mobile data analytics company (see examples in Fig. 1.4). The dataset contains approximately 11M LTE measurements, in areas of 300km^2 and 1600km^2 for `NYC` and `LA` respectively. To the best of our knowledge, the `NYC and LA datasets` are among the largest used to date for coverage maps (or other signal strength) prediction, in terms of any metric (number of measurements, geographical scale and number of cells), enabling us to gain unique and valuable insights into the problem.

Chapter 2

Background and Related Work

2.1 Problem Statement and Preliminaries

2.1.1 Signal Strength - Key Performance Indicators (KPIs)

Traditionally, performance evaluation of cellular networks was primarily focused on: received signal strength (RSS). For example, back in the days of the GSM protocol[1] things were relatively simple: each mobile phone was using a single carrier over a single channel of just 200KHz bandwidth, accessing it with TDMA. The RSS of the GSM protocol was measured and reported on BCCH (broadcast control channel) and/or SACCH (slow associated control channel). Most importantly, these 200KHz contain the entire useful signal for the mobile's communication.

In contrast, today's LTE is much more complicated; LTE uses OFDM (orthogonal frequency division multiplexing) with wide-band channels (up to 10MHz), therefore multiple users share

[1] GSM it was deployed in the 1990s and has been retired since 2017 `https://www.gsmarena.com/at_t_has_officially_shut_down_its_2g_network-blog-22811.php`, accessed April 2020.

all the available bandwidth. Received signal strength indicator (RSSI) includes interference, transmissions from other users and other cells. Thus, in addition, multiple comprehensive "key-performance indicators" (KPIs) have been introduced for LTE's operation and evaluation. Representative examples include the reference signal received power (RSRP), the reference signal received quality (RSRQ) and the channel quality indicator (CQI).

These LTE KPIs related to RSS and the overall wireless link performance (*i.e.,* RSRP, RSRQ, CQI, throughput) from the end-users devices, are defined by 3GPP [30], the standardization entity for the mobile broadband standard under the term "minimization of drive tests" (MDT) data [44]. Another term that is being used for LTE KPIs is "user measurement data" (UMD) [49] (*e.g.,* state-of-the-art work from a tier-1 mobile operator [61, 49]).

Reference Signal Received Power (RSRP): 3GPP [30] defines RSRP, y^P, as the average over the power contributions of the resource elements that carry cell-specific reference signals within the considered frequency bandwidth (*e.g.,* 5 or 10MHz wide-band LTE channels). RSRP is typically reported in dBm by UEs (user equipment) and is a RSS indicator type since it is the average received power of a single reference signal (RS) of one resource element (equivalent to a 15KHz subcarrier) [28]. Basically, RSRP excludes interference and noise from other sectors, estimating more accurately the signal power of the serving cell. It is of great importance for LTE since RSRP (jointly with RSRQ) measurements are mainly utilized by smartphones for cell selection, handover decisions, mobility measurements (*e.g.,* signal bars which are defined later in Sec. 5.2.3) and power control calculations.

Reference Signal Received Quality (RSRQ): The RSRQ measurement, y^I, is a proxy for measuring a channel's interference. It is defined as the ratio of the power used by resource blocks (RBs) over the total received power RSSI (which includes power from other sectors/cells, thermal noise co-channel interference *etc.*) over the same bandwidth:

$RSRQ = (N \times RSRP)/RSSI$. RSRQ can essentially be seen as the portion of the useful signal and reported in dB units as a ratio.

Channel Quality Indicator (CQI): The CQI, y^C, is a unit-less metric ($y^C = \{0, \cdots, 15\}$) of the overall performance of the wireless channel. For example, higher CQI could trigger more aggressive modulation by LTE; or CQI values are used for LTE scheduling decisions. The exact CQI calculation (as well as that of RSRP and RSRQ) details differ across devices and manufacturers, which usually consider their implementations proprietary, since 3GPP [30] just provides generic guidelines.

LTE Network Architecture: (LTE Cells *vs.* LTE TA) Next, we briefly review the basic LTE network structure, which will inform how to build coverage maps and predict these values. A UE (*i.e.,* a mobile phone) is served by a base station (*a.k.a.* cell tower) and is being attached to a specific cell. A serving LTE cell is uniquely identified by the CGI (cell global identifier), which is the concatenation of the following identifiers: the MCC (mobile country code), MNC (mobile network code), TAC (tracking area code) and the cell ID. We abbreviate and refer to CGI as cell ID or cID. LTE also defines Tracking Areas (which we will refer to as LTE TA) by the concatenation of MCC, MNC and TAC, to describe a group of neighboring cells, under common LTE management for a specific area. The term cell tower refers to the physical location where several antennas are serving multiple different cells, usually indicated by a common prefix in cID. Please note that the size of a cell varies from a few hundred square meters in an urban environment to up to several square kilometers in rural areas. In this thesis, we examine models of the coverage map both per cID and LTE TA. We refer the reader to [56] for a short LTE network architecture primer.

Quality of Service (QoS). Moreover, we need to mention the connection between the quality of service (QoS) and KPI terms because QoS is a ubiquitous concept in telecommunications. According to [12], QoS in cellular networks is defined as: "The capability of the cellular operators to provide a satisfactory service, which includes voice quality, signal strength, low call blocking, dropping probability and high data rates for multimedia-data applications *etc.* For network-based services, QoS depends on the following factors: throughput, delay, packet loss, error rate, *etc.*" To make things even more complicated, official documentation from another cellular telephony organization (GSMA [9]) explicitly states that "A QoS parameter is also called quality key-performance-indicator (KPI)".

In this thesis, we try to simplify the high heterogeneity of the terminology of the field and we use the terms: "key-performance indicators" (KPIs) for LTE RSRP, RSRQ, CQI and QoS for the functions defined on signal strength and KPIs y such as call drop probability, signal bars (4 class quality) and binary quality of the cellular network.

2.1.2 Signal Strength and Coverage Maps Definition

Traditionally, coverage maps are designed to indicate the service areas of transmitting base stations, which consequently refer to the levels of the UEs' received signal strength. However, in LTE there are so many parameters that define the level of service; *e.g.,* we might experience very good RSRP but the interference (RSRQ) could be high, causing poor performance. Thus, in this thesis, we use the term coverage maps to refer to maps of all KPIs, including signal strength, and all the QoSs such as call drop probability, and "mobile coverage indicator" (*i.e.,* yes or no coverage); please note that where necessary we disambiguate and use explicitly the term coverage *indicator*. We use the term mobile coverage maps as a superset including both continuous and discrete forms of maps as well as QoS maps; signal strength maps refers to only the continuous version and "coverage indicator" refers to the binary problem.

An *observed coverage map* is a collection of N measurements $(\mathbf{x_i}, y_i)$, $i = 1, 2...N$, where the label $y_i = \{y^P, y^I, y^C\}$ of data y_i denotes the KPI of interest given the feature vector $\mathbf{x}_i$ which specifies the location, time, hardware and other features which the KPI is to be mapped. In general, an operator's interest is not in the observed signal strength map, but in an underlying true signal strength map, defined by the conditional distribution $Y|\mathbf{x}$ for an arbitrary $\mathbf{x} \in \mathbb{X}$, where Y is the (generally unobserved) KPI at $\mathbf{x}$ and $\mathbb{X}$ specifies a region of interest (e.g., an areal unit, time period, *etc.*). This denotes an estimate of the true signal strength map by machine learning (ML), where our goal is to answer queries regarding $Y|\mathbf{x}$ or functions thereof by training a predictor $\widehat{y}$ on the observed signal strength map.

Moreover, in this thesis, we also study coverage maps in the quality domain: $(\mathbf{x_i}, Q(y_i))$. The benefits of this approach is two-fold: First, this is a standalone problem itself: the operators' interest is not always in KPI y itself, but in $Q(y)$ (*e.g.,* call drop probability) or predict and create the 4-bars [63, 55] or 0-1 indicator [34] used for commercial representations of the maps. Second and more importantly, it can be used to implicitly modify the loss function of the coverage map; in this thesis, we extensively demonstrate how we can leverage $Q(y)$ domain prediction to build better maps in y and vice versa ($Q(y) \leftrightarrow y$).

Last but not least, cellular coverage maps are described in the literature with a wide range of different names: "RF Coverage Maps" [61, 49], "Mobile Coverage Maps" [3, 33, 32], "Cellular Coverage Analysis/Prediction" [14, 50],"Signal Map" [39], "Signal Strength Maps" [7] and "Radio Environment Maps" [75, 34], to name just a few. We use the terms "coverage maps" and "signal (strength) maps" interchangeably since a big part of this work handles the prediction of RSRP which is the signal strength metric for LTE.

2.1.3 The Coverage Map Prediction Problem

The goal is to predict signal (coverage) map value y_j at a given location, time, and/or other features of interest (as specified by $\mathbf{x}_j \in \mathbb{X}$), based on available historical measurements with labeled data $(\mathbf{x}_i, y_i)$, either in the same cID or in the same LTE TA. For example, this might be needed by cellular operators for planning, maintenance, as input to network algorithms [41], or used from mobile analytics companies (*e.g.,* [55]) to produce cellular maps for the areas where data are not available or are expensive to obtain.

The real world underlying phenomenon for y (denoted as Y) is a complex process which depends on numerous wireless factors and environmental characteristics [60]. Most of prior work has focused on developing increasingly sophisticated model-based (*e.g.,* [76, 16]) and machine learning techniques for predicting directly $\widehat{y}$, both of which require complicated environment data (the latter needs LiDAR data and the former detailed topologies, see Sec, 2.2). Moreover, prior work has focused on a single task: the minimization of the mean squared error of the prediction task with geospatial predictors that can handle just location features (*e.g.,* [59, 19]).

In sharp contrast to the prior art, we develop a prediction framework that (i) uses a rich set of features readily available by Android APIs, (ii) allows cellular operators to express operational objectives and optimize the prediction and (iii) enables valuation of training data points and data minimization.

The first coverage map problem we consider (Chapter 4) is to develop a predictor for the missing signal map values y_j (*e.g.,* LTE RSRP) for the feature space $\mathbf{x}$. Our goal is to predict an RSS value at a given location, time, device and potentially considering additional contextual information. We treat the problem as a regression ML problem where the goal is to minimize the mean squared error (MSE) metric.

The second coverage map problem we study (Chapter 5) builds on the first problem and considers the *loss* to be minimized: particular choices of loss functions will improve performance for certain objectives, while degrading it in others. We consider two general factors relating to the choice of loss. First, operators' interest is not always in KPI y itself, but in some *quality of service function*, Q, that depends on y (we already defined examples of coverage maps for this case in Section 2.1.2). While prior work focused on predicting y (*e.g.,* w.r.t.

mean squared error), we instead consider signal map prediction that minimizes error in the predicted value of $Q(y_j)$ itself and we set up the relationship between $Q(\widehat{y}) \leftrightarrow \widehat{Q}(y)$. The nonlinear dependence of quality-of-service on raw signal strength makes this direct approach superior for many practical applications. Second, the operator may wish to assign more importance to some values of $\mathbf{x}$ more heavily than others. While prior work with conventional training schemes, implicitly assumes that importance corresponds to data sampling frequency, we instead consider optimization w.r.t. an application-specific weight function $W(\mathbf{x})$ that may or may not be the same as the sampling distribution.

The third and last problem we consider (Chapter 6) builds naturally on the different evaluation schemas we develop above. We define the data Shapley value ϕ_i, which quantifies the importance of a datum $(\mathbf{x}_i, y_i)$ for the combination of a given dataset D, predictor algorithm $\mathcal{A}$ and performance metric $V(f)$, *i.e.,* $\phi_i = \{D, \mathcal{A}, V(f)\}$. Basically, the choice of $\widehat{f_y}(\mathbf{x})$ and $\widehat{f_Q}(\mathbf{x})$ corresponds to $\mathcal{A}$ and the application-specific $W(\mathbf{x})$ defines the performance score. This allows the valuation of the training points used in our ML predictors, which in turns, can be used to remove low quality data and for data minimization.

In summary, this thesis develops predictors $\widehat{y} = \widehat{f_y}(\mathbf{x})$ for signal strength (*e.g.,* LTE RSRP) as well as $\widehat{Q}(y) = \widehat{f_Q}(\mathbf{x})$ for quality functions Q, where $\widehat{f_y}(\mathbf{x})$ and $\widehat{f_Q}(\mathbf{x})$ are optimized w.r.t. an appropriate weight function $W(\mathbf{x})$. For each of these predictors ($\widehat{f_y}(\mathbf{x})$ and $\widehat{f_Q}(\mathbf{x})$) the most valuable data points are mined for different choices of $W(\mathbf{x})$ and/or $\widehat{f_Q}(\mathbf{x})$ through the data Shapley values.

2.1.4 Notation Summary

From now on, we use the term “coverage” to refer to y or $Q(y)$. Throughout this thesis we use the following notation unless specified otherwise. We use a boldface capital letter to denote a matrix (e.g. $\mathbf{X}$) and a lower case bold letter for a vector (*e.g.,* $\mathbf{x}$). Letters that are not bold describe scalars, with uppercase letters typically used for dimensions or count (such as N, M), lower case used for indexing (i for training data, j for the unobserved, test data) or greek for hyperparameters (γ, λ). The uppercase calligraphy letters $\mathcal{D}$ denotes a set of data and $\mathcal{A}$ denotes an estimation algorithm.

The letter $\widehat{y}$ represents the prediction of a regressor or a classifier and the symbols $\widehat{f}_y(\mathbf{x})$, $\widehat{f}_Q(\mathbf{x})$ denote the prediction function which produces $\widehat{y}$ and $\widehat{Q}(y)$ respectively. The letter y is used for the labels of our data, *i.e.,* the signal strength values and the other KPIs ($y_i = \{y^P, y^I, y^C\}$). For simplicity, we imply $y = y^P$, since we primarily demonstrate prediction with LTE RSRP in this thesis, unless otherwise noted. Finally, because this chapter has defined various terms and notations that we will use throught this thesis, for convenience Table 2.2 summarizes the definitions and terminology and Table 2.1 summarizes the common symbols and notation used throughout this thesis (more specific notations for the technical contributions of each chapter will be included in each chapter accordingly).

Terminology for Cellular Operators: We use the terms “cellular operators” and mobile network carriers (MNCs) interchangeably; the latter can be found mainly in the various technical cellular specifications and in the data description in the network APIs, where, more specifically, MNC stands for “mobile network code” and provides a unique identifier for the network carrier. Moreover, the term Mobile Network Operators (MNOs) is also being used in the literature. For certain parts of the data description and results, we adopt the acronym MNC, which implies MNO, but overall we use the term cellular operators.

2.2 Related Work

Wireless signal strength (*a.k.a.* received signal strength, RSS) is a fundamental property of wireless networks. Estimation and modeling for mobile coverage maps (*e.g.,* [50, 32, 33]) are relevant in many other application contexts, such as location estimation techniques (*e.g.,*

	Notations	Definitions-Description
Data: KPIs	$y = \{y^P, y^I, y^C\}$	Label - KPI (Key Performance Indicator)
	y^P	RSRP: Received Signal Reference Power
	y^I	RSRQ: Received Signal Reference Quality (Interference)
	y^C	CQI: Channel Quality Indicator
Data: Features	$\mathbf{x}$	Measurement's Features
	$\mathbf{l} = (l^x, l^y)$	Location Features (spatial coordinates)
	$\mathbf{t} = (d, h)$	Time Features (day, hour)
	dev	Device Model (hardware)
	out	Indoors or outdoors indicator
	$\|\|\mathbf{l}_{\text{BS}} - \mathbf{l}_j\|\|$	Euclidean distance between transmitter and receiver
	$freq_{dl}$	EARFCN (*a.k.a.* LTE Frequency channel)
	cID	LTE cell unique identifier (*i.e.,* Cell Global Identifier)
Network Quality Functions	$Q(y)$	Network Quality Function (*e.g.,* $Q_c(y^P)$, $Q_{cdp}(y^P)$)
	$Q_c(y^P)$	Mobile Coverage Indicator
	$Q_{cdp}(y)$	Call Drop Probability
Predictors' Notation	$\widehat{y}$	Prediction of a regression or classifier
	$\widehat{f}_y(\mathbf{x})$	Predictor for a signal map value
	$\widehat{f}_Q(\mathbf{x})$	Predictor for a QoS Value
Error Scores/ Importance-Sampling	$L(\widehat{y}, y)$	Loss function of its arguments (this thesis: squared loss)
	$s(\mathbf{x})$	Sampling Distribution
	$W(\mathbf{x})$	Weighting Function

Table 2.1: Symbols and notation used throughout this thesis.

[37, 61, 5]), resource allocation in wireless networks (*e.g.,* [24, 54]), robots navigation (*e.g.,* [37]), *etc.* In this section, we review the relevant literature on RSS prediction and signal maps construction and we discuss the state-of-the-art which provides background and baselines for all of the following chapters in this thesis. We compare them accordingly, in more depth, with our proposed methodology in the corresponding chapters. Broadly speaking, signal strength prediction can be done through propagation models or through data-driven approaches, which include geospatial interpolation (*a.k.a.* geostatistics), [50, 33, 59] and Machine Learning [29, 61], to predict signal strength maps from historical data.

Term	**Definiton**
RSS	Received Signal Strength
GPS	Global Positioning System
GSM	Global System for Mobile Communications (2G)
LTE	Long Term Evolution (LTE-Advanced conforms to 4G)
BS	Base Station
UE	User Equipment (mobile device)
KPI	Key Performance Indicator
UMD	User Measurement Data
MDT	Minimization of Drive Tests
RSRP	Received Signal Reference Power
RSRQ	Received Signal Reference Quality
CQI	Channel Quality Indicator
EARFCN	E-UTRA Absolute Radio Frequency Channel Number
MNC	Mobile Network Carrier (*a.k.a.* Cellular Operator)
MNO	Mobile Network Operator (aka Cellular Operator)
RFs	Random Forests

Table 2.2: Miscellaneous abbreviations used throughout the entire thesis.

2.2.1 Propagation Path Loss Models

Wireless propagation (*a.k.a.* radio frequency propagation) and path loss modeling have been extensively studied. This body of work usually combines measurement campaigns, physical layer and environment modeling, in order to model the path loss of the signal at the receiver. Historically, this was the prevalent way to estimate signal strength given that measurements were much more expensive and limited in the 1990s and early 2000s. For example, one way that operators used to obtain detailed and accurate measurements is by hiring dedicated vans (*a.k.a.* war-driving [73]) with special equipment, to drive through, measure and map the received signal strength (RSS) in a particular area of interest. The data from war-driving could either be used directly or be fitted to the propagation path loss models. Early examples of this family of models back in the GSM era, included the Hata model [66] and the COST 231 [25] (*e.g.,* Walfisch-Ikegami model).

State-of-the-art wireless propagation and path loss models include WINNER I/II [16], which tries to develop a single ubiquitous radio access system adaptable to a range of mobile communication scenarios. Recently, Ray tracing [76] for radio propagation modeling has gained a lot of attention, due to its accuracy. However, it is computationally expensive and requires a detailed mapping of the environment (*e.g.,* buildings' topology). The same applies to the entire family of these models (ray tracing, WINNER I/II, COST 231), which typically require a fine-grained tuning of many different parameters and inputs (*e.g.,* environment -indoors, outdoors, rural, downtown-, topology, the number of floors in the building, the street width, the height of transmitters/receivers, occasionally 3D maps [76] *etc.*).

A simple, yet widely used, propagation model is the Log Distance Path Loss (`LDPL`) model [60] and its variant for the indoor environments [8], which assumes wireless shadowing (*i.e.,* large scale fading [66]) following a log-normal distribution (*i.e.,* normal in dBm) and the path loss could be modeled with a logarithmic attenuation. Later in Chapter 4, we utilize a homogeneous and heterogeneous `LDPL`, *i.e.,* a different power loss exponent per location, as the representative baselines of propagation models and we compare them to our proposed data driven approach with Random Forests (`RFs`).

2.2.2 Data Driven Prediction (I): Geospatial Interpolation

Both the complexity of propagation models and the abundance of available mobile data in the recent years, have shifted the research efforts towards data-driven prediction. Prior work used geospatial interpolation for RSS prediction, where RSS at a particular location is predicted by interpolating neighboring measurements. Geostatistics predictors have been extensively used in environmental sciences, meteorology (*e.g.,* humidity, temperature estimation), remote sensing and many other fields when there are some available measurements and missing values need to be predicted. The output of the estimator is a weighted average of the neighboring measurements after solving an optimization problem.

For example, work in [50] compares the accuracy of various geospatial interpolation techniques for cellular signal strength prediction, using crowdsourced measurements values from Android devices. It examines the impact of (a) inaccurate locations, (b) sparse measurements and (c) non-uniformity in crowdsourced datasets and concludes that Ordinary Kriging (OK) is the one of the best of this category. Furthermore, work in [59] (coverage maps of a 2.5GHz WiMax network) and [19] (a spectrum sensing database for cellular bands and DTV bands) have developed methods which incorporate wireless propagation characteristics in geospatial models, namely Ordinary Kriging with Detrending (OKD) and regions Partionioning (OKP, OKPD). An interesting framework which uses geostatistics is ZipWeave [33]. ZipWeave identifies sub-regions with similar radio propagation characteristics and high predictive value in order to reduce sample size of the required data and improve prediction accuracy. Both the former (OKP, OKPD) and the latter (ZipWeave) solve separately a different optimization problem for each local subregion, which make them impractical for city-wide signal maps.

Apart from the scalability, geostatistics have additional limitations. For example, these methods (*e.g.,* OK, OKD, ZipWeave [59, 19, 33, 33]) cannot naturally incorporate additional dimensions such as time, frequency, hardware and network information (since the optimization problem would become non tractable) as our proposed ML model in Chapter 4 of this thesis does. Furthermore, OK has time complexity $\approxeq O(N^3)$ which makes it non-efficient for real world applications. A faster implementation of OK, namely Fixed Rank Kriging (FRK), which considers antenna directionality characteristics has also been proposed [14]; however, (i) the issue of adding easily additional features dimensions and (ii) fine-tuning of the correlation matrix (semivariogram) per region, still remain.

2.2.3 Data Driven Prediction (II): Machine Learning

Recent literature has started applying Machine Learning for signal (coverage) maps prediction. Work in [29] uses deep neural nets (DNNs) along with detailed 3D maps from "light and range detection data" (LiDAR) for LTE RSRP prediction; however LiDAR are expensive to obtain and not always readily available. Supervised ML is also used in [39], which uses Bayesian Compressive Sensing (BCS), to develop a framework for inference of missing signal strength values jointly with users' incentives control. However, BCS requires the fine tuning of separate spatial and temporal correlation matrices for each different environment, which could be computationally expensive and limiting for large scale signal strength prediction in city-wide scale. The experimental results are limited to a couple of thousand data points and a very small geographical area of just 7km^2.

Localization and RSS modeling: RSS modeling [31, 72] is also important in the context of mobile devices (*a.k.a.* UEs) localization [61] or other wireless sources localization [5, 4]. Interestingly enough, GPS-free [5, 4] or assisted-GPS (aGPS) location estimation has been increasingly relying on wireless signal strength measurements; the latter tries to mitigate the GPS' battery impact and the former offers location estimation when GPS is not available at all in the device [5, 4]. Although the final goal is slightly different, the fundamental goal remains to provide a statistical model (*i.e.,* an ML model) for the RSS measurements in order to facilitate localization. This ML model could be used for prediction as well: *e.g.,* assuming a Gaussian distribution for our data, the maximum likelihood estimation (MLE), would be given by the mean value of our data.

Examples of state-of-the-art work in ML models for facilitating UE localization come directly from a major cellular operator research lab [61, 49]. Work in [61] develops UE localization algorithms based on UEs' UMD records (*e.g.,* RSRP, RSRQ, RSSI), where the LTE RSRP likelihood is modeled via `RFs`, with training features only including the measurements' latitude and longitude. Similarly, [49] focuses on UE localization, given large scale UMD records and builds synopsis of RF coverage maps in order to facilitate the online matching of the RF measurements with the locations. The authors consider the grid size, compare model driven maps *vs.* data driven maps and they conclude that RSRP can be modeled as a normal distribution, $\mathcal{N}(\mu, \sigma^2)$, with parameters estimated from the data, but considering only spatial coordinates as the features. In sharp contrast to the above, we use `RFs` with an extended set of features, including but not limited to location, time, frequency, device and network information specifically for the problem of LTE KPIs (*e.g.,* RSRP) prediction and coverage maps creation. Last but not least, other examples come from the field of robotics localization; Gaussian processes (GPs) for spatiotemporal signal strength modeling have also been applied for users' localization [74] and location estimation via particle filters (PFs) estimators [5, 31] has used RSS modeling. Gaussian Processes are computationally expensive ($\approxeq O(N^3)$) and similarly to geostatistics, cannot easily incorporate additional features.

Data Volume (Throughput) Maps: Apart from signal strength, state-of-the-art focuses on constructing the mobile traffic volume map (*i.e.,* KPI throughput) [77, 71]. Work in [77] deploys DNNs to capture relations between neighboring input points and spatiotemporal locality in feature representations as well as the current traffic trends. With a double spatiotemporal neural networks (STN) technique the authors are able to predict accurately throughput per location for the city of Milan. Similarly for throughput maps, work in [71] uses long short-term memory units (LSTMs) for temporal modeling and global stacked autoencoder (GSAE).

2.2.4 Tools and Datasets

Although there is an abundance of crowdsourcing platforms and studies (*e.g.,* [53, 2, 68]), the public available received signal strength (RSS) datasets are limited (*e.g.,* [5, 46]) and they do not include any large scale LTE measurements. There is a number of measurement tools that can collect cellular performance measurements from end-devices. Some of them are commercial/proprietary (*e.g.,* Speedtest.net), but some are made available by the research community. One significant effort in the latter category is Mobilyzer [53] and Mobiperf-App [40], whose library allows active and passive measurements. However, neither includes some key features used in this thesis (such as the cell-IDs) and precise location information nor all the LTE KPIs used in this thesis for prediction.

Given the lack of publicly available datasets for LTE performance (KPIs/signal strength), we developed a tool for collecting the `Campus dataset` ourselves, and the `NYC and LA datasets` used in this thesis were provided to us by a mobile data analytics company. To the best of our knowledge, this is the largest of its kind used for LTE signal (coverage) maps prediction at metropolitan scale, and provides novel insights into city-wide prediction. They contain 10.9 million LTE data points in areas of 300km^2 and 1600km^2 for `NYC` and `LA` respectively, instead of at most a couple of tens of square kilometers in prior work [59, 39, 19, 29] and a couple of dozens of thousands measurements at maximum [33, 19]. We defer to Chapter 3, for background on measurements collection technqieus, the design of our crowdosourcing system, as well as the external mobile analytics dataset.

Spectrum Monitoring - Cognitive Radios. A category of work related to signal strength, but not directly related to the development of prediction algorithms of the signal maps themselves, is spectrum monitoring [52, 80] of cellular bands, TV bands [79], radar bands *etc.* and cognitive radio modeling [54]. This body of work utilizes GNU USRP, and/or usb-dongles (RTL-SDR) in smartphone for signal strength and related spectrum measurements. The Specsense framework, which was discussed earlier for the development of geostatistics methods, firstly developed a spectrum sensing platform with RTL-SDRs and USRP devices in [18].

Machine Learning Methodology. There is an extensive body of work in ML related to the prediction of missing values from historical data. We review the algorithms where appropriately. Work in [45] utilizes importance sampling to modify the training procedure (more specifically the order of the data in the stochastic gradient descent) to improve prediction. In sharp contrast, we utilize importance sampling to define general reweighted error metrics and to handle the problem of the mismatch between the training and the target distribution problem (*a.k.a.* dataset shift [67]).

Chapter 3

Datasets

> *"Dataism", is the new mindset of Big Data, where information flow is "essential" and the "supreme value".*
>
> Yuval Noah Harari, "Homo Deus"

3.1 Overview

We needed realistic datasets in order to study the problem of mobile coverage map prediction. In this chapter, we present the two types of mobile LTE network datasets used throughout this thesis and the relevant crowdsourcing system we built to collect LTE measurements. Table 3.1 summarizes the two types of datasets used in this thesis: the first is a campus dataset and the second consists of two city-wide datasets from NYC and LA. The former was

The material in this chapter is based on the poster *A system for crowdsourcing passive mobile network measurements* by E. Alimpertis, and A. Markopoulou. presented in the Poster session of the 14th USENIX NSDI Conference, 2017 Boston **and** the paper *"City-Wide Signal Strength Maps: Prediction with Random Forests"* by E. Alimpertis, A. Markopoulou, C. T. Butts and K. Psounis, published in the ACM Proceedings of World Wide Web Conference (WWW'19), May 13-17, 2019, San Fransisco, CA, USA. ACM, NY, USA, 7 pages, `https://doi.org/10.1145/3308558.3313726`. © 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

Dataset	Period	Areas	Type of Measurements	Characteristics	Source
Campus	02/10/17 - 06/18/17	Univ. Campus Area $\approxeq 3km^2$	LTE KPIs: RSRP, [RSRQ]. Context: GPS Location, timestamp, *dev*, *cid*. Features: $\mathbf{x} = \left(l_j^x, l_j^y, \mathbf{t}, dev, out, \lvert\lvert\vec{l}_{\text{BS}} - \vec{l}_j\rvert\rvert_2\right)$	No. Cells = 25 No. Meas $\approxeq$ 180K Density ($\frac{N}{m^2}$) Per Cell: 0.01 - 0.66 Overall Density: 0.06	Ourselves
NYC & LA	09/01/17- 11/30/17	NYC Metropolitan Area $\approxeq 300km^2$	LTE KPIs: RSRP, RSRQ, CQI. Context: GPS Location, timestamp, *dev*, *cid*, earfcn. Features: $\mathbf{x} = \left(l_j^x, l_j^y, \mathbf{t}, cid, dev, out, \lvert\lvert\vec{l}_{\text{BS}} - \vec{l}_j\rvert\rvert_2, freq_{dl}\right)$	No. Meas NYC $\approxeq$ 4.2M No. Cells NYC $\approxeq 88k$ Density NYC $\approxeq 0.014\frac{N}{m^2}$	Mobile Analytics Company
		LA metropolitan Area $\approxeq 1600km^2$		No. Meas LA $\approxeq$ 6.7M No. Cells LA $\approxeq$ 111K Density LA $\approxeq 0.0042\frac{N}{m^2}$	

Table 3.1: Overview of the Datasets used in this Thesis.

collected by *ourselves* and the crowdsourcing module was designed as part of AntMonitor network monitoring tool [1]. The latter dataset, was provided by a major mobile analytics company and includes city-wide collected datasets from NYC and LA metropolitan areas. These two real-world datasets are among the largest used in the literature, thus providing us with unique opportunities to evaluate coverage maps prediction in a city-wide scale. The design of the crowdsourcing system itself helped us to gain insights and understand further the practical implications of such systems (*e.g.,* sampling strategies) and thus focus on handling and mitigating these effects (see Chapter 5).

In Section 3.2, we present our system and in Section 3.3 we provide an overview of the datasets generously provided to us by a major mobile crowdsourcing company. Section 3.4 provides the common description of the datasets such as the measurements' information and how we store all the data under a common format. Both datasets are anonymized, *i.e.,* we neither collect nor store any user identities or pseudo-ids.

3.1.1 Background

We already reviewed some of the available datasets and tools for cellular measurements in Chapter 2 (see Section 2.2.4). Next, we present a taxonomy of the potential collection strategies and systems for collecting data for coverage maps available in the literature. Broadly speaking, collection of coverage maps data can be done either (1) inside the network

	Network Infrastructure	MySignals [3]	Speedtest.net, Mobilyzer [53]	**Our Work in AntMonitor**
Granular Large-Scale	✓	✓	✓	✓
Infrastructure Access Free	✗	✓	✓	✓
Precise Location	✗	✓	✓	✓
Network Edge	✗	✓	✓	✓
Cellular Info	✓	✓	✓	✓
WiFi Info	✗	✗	✗	✓
Active Throughput	✗	✗	✓	✗
Passive Throughput per TCP/IP flow	✓	✗	✗	✓
NO Data Overhead	✓	✓	✗	✓
NO User Action	✓	✓	✗	✓

Table 3.2: Network Performance Monitoring Approaches Compared to Our System.

infrastructure or (2) at the network's edge; using (2a) passive measurements or (2b) active measurements collection. Representative examples for (1) include (i) large scale TCP/IP flows stats collection at cellular operator's infrastructure [26] and (ii) LTE UMD (*i.e.,* RSRP, RSRQ measurements) collection in [49] at the LTE eNodeB/EPC. It should be noted that UEs usually report UMD to the LTE eNodeB through feedback or control channels and used in various LTE operations. Although gathering data at the infrastructure offers certain advantages such as large scale and granular measurements, it misses information from the network edge/wireless link (neither precise location information is included nor all UMD data are readily available to be reported) and it requires access to the cellular operators's infrastructure.

Passive measurements at network's edge capture entirely the wireless link, can offer precise UEs location and be deployed on users' smartphones (*i.e.,* the UEs). For example, the same work that collected LTE UMD at network infrastructure [49], it also collected GUMD (GPS-tagged UMD) by installing proprietary software at a subset of the UEs and collect them in a central server (completely independently of the UMD reports)[1]. Mobile analytics

[1] Very interestingly, the purpose of the GUMD collection was to train location estimation algorithms in order to label the rest of the UMD data (either GPS-less services or further evaluation).

companies perform also passive measurements (*e.g.,* OpenSignals [55], Rootmetrics [63] and Tutela [70]) in order to collect signal strength and passive TCP throughput (measuring bytes traffic over time) and offer online coverage maps. This strategy offers the advantage that does not need access to the operators infrastructure and also offers granular measurements at network's edge, however scalability is a big issue[2]. Some active measurements 2(b) rely on the individual user to trigger the data collection (as the most representative examples we refer to the popular `speedtest.net` or mobilyzer [53]). A comparison of our own crowdsourcing system with the existing methodologies and taxonomy is summarized in Table 3.2.

3.2 The UCI Campus dataset

As mentioned in Section 2.2, there are no large scale LTE datasets publicly, therefore, we decided to move forward with the design of our own system and make it available to the research community[3]. The goal is twofold; firstly we gain low-level understanding of the LTE network architecture/KPIs themselves and know-how around the collection strategies, including how the biased sampling strategies could emerge in larger scale datasets from mobile analytics companies. Second and most important, we are able to collect a large number of measurements with different characteristics from the larger datasets that we were able to obtain from industry partners.

3.2.1 Dataset Overview

We collected the first dataset at University of California, Irvine (UCI) campus. This `Campus dataset` is relatively small: $180,000$ data points, collected by seven Android devices that

[2] Lot of similar research studies just include dozens of users as we already reviewed in Section 2.2.4; that's the importance of accessing mobile analytics' company datasets.

[3] https://github.com/UCI-Networking-Group/AntMonitor/

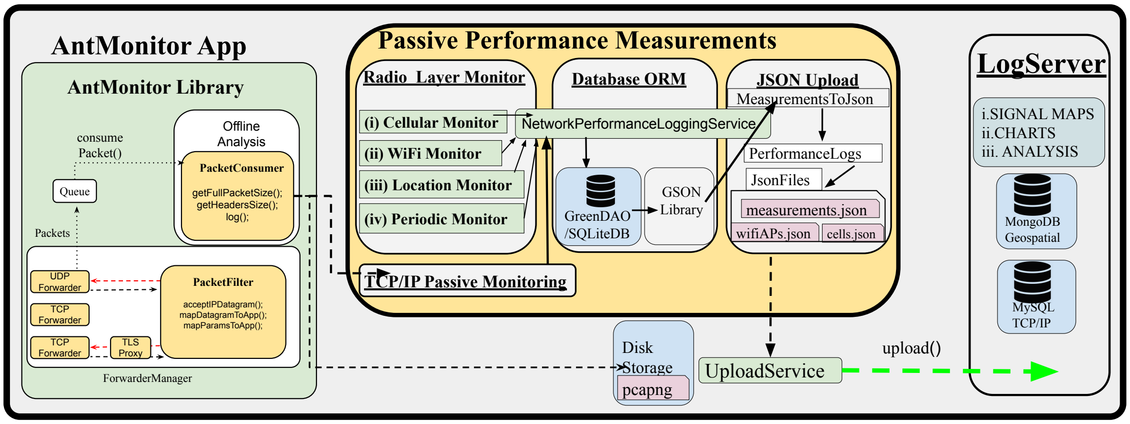


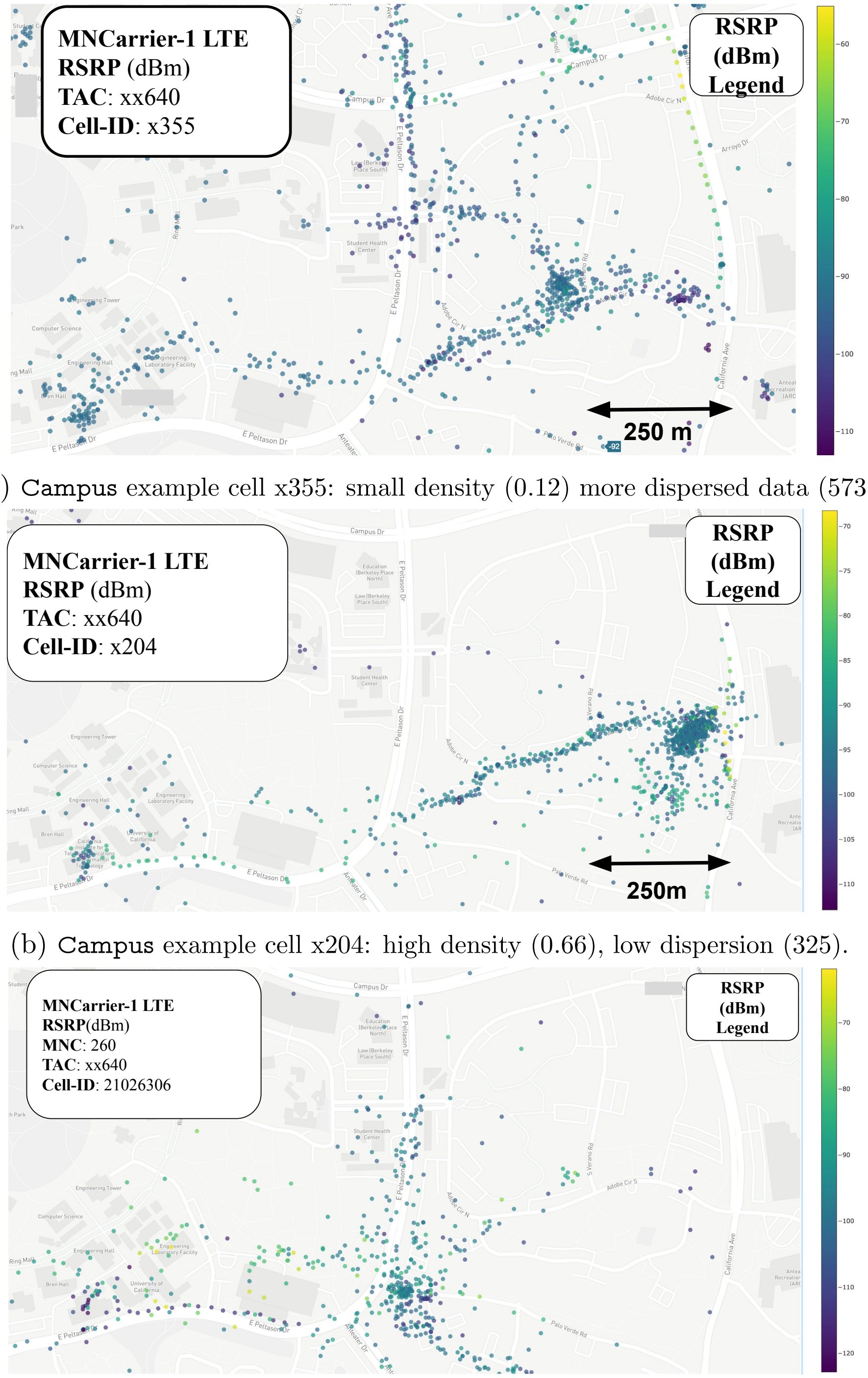
Figure 3.1: System overview, design and data flow in our passive network monitoring crowdsourcing too, implemented as part of `AntMonitor`. In this thesis, the cellular signal strength (*e.g.,* LTE RSRP) is used for mobile signal maps prediction.

belong to graduate students and faculty members, using 2 cellular providers. In terms of geographical area, it covers approximately $3km^2$, as the devices move between student housing, offices and other locations on campus. Some examples are depicted in Fig. 3.2.

Although small, this is a dense dataset, with multiple measurements over time on the same and nearby locations. Furthermore, the cells in this dataset exhibit a range of characteristics: (i) the number of measurements N per cell varies from a few thousand up to 50 thousand; (ii) the measurement density ($i.e.,\ \frac{N}{\text{sq}\ m^2}$) also varies from 0.01 to 0.6; (iii) the measurements in some cells are concentrated in a few locations while in some others they are dispersed. These properties (number of measurements, density and dispersion as well as the mean and variance of the LTE RSRP) and how the affect the signal maps prediction are reported later in this thesis; please see Section 4.5.2 and Table 4.4.

3.2.2 Crowdsourcing System for Data Collection

Fig. 3.1 presents the design of our measurement system. Our system is a fully functional end-to-end crowdsourcing system which allows the collection of passive network measurements such as wireless received signal strength (WiFI RSSI or LTE RSRP), various others LTE KPIs



(a) `Campus` example cell x355: small density (0.12) more dispersed data (573).

(b) `Campus` example cell x204: high density (0.66), low dispersion (325).

(c) `Campus` example cell x204: small density (0.011), more dispersed data (701).

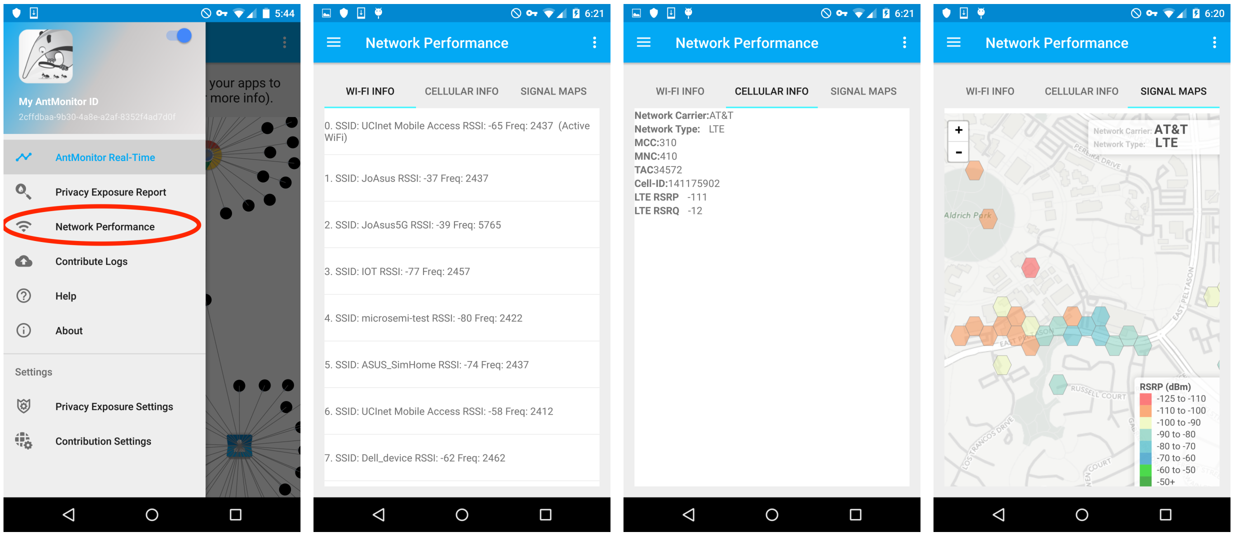
Figure 3.2: LTE RSRP Map Examples from `Campus dataset`. Color indicates RSRP value.

(*e.g.*, RSRQ), TCP/IP layer measurements such as throughput as well as other contextual information (location, time *etc.*). Our system handles the data transfer to a central web-server for permanent storage via a custom designed web-service via JSON. A comparison of our crowdsourcing system with state-of-the-art network measurements approaches is summarized in Table 3.2.

Measurements on the Device. On the device, we incorporate our network monitoring module in `AntMonitor` as shown in Fig. 3.1 which uses the Android APIs to obtain LTE information: cellular LTE RSRP, RSRQ, CQI, network carrier, radio access technology (RAT) to confirm that the network is LTE, and the relevant serving cell information cID as defined earlier. Each measurement is initiated by Android's notifications/callbacks for network and location changes (*e.g.,* RSS or cell status change) and is also piggy-backed on location change notifications from other apps, in order to achieve a low energy footprint. Rich contextual information is also recorded at the time of the measurement, including: timestamp, device hardware type (dev) and location via the Google Location API, which offers both precision and low energy consumption. Several of the app's screens such as the signal map for LTE RSRP on user's device are shown in Fig. 3.3. It should be noted at this point that these are relatively easy strategies to be implemented in order to minimize the battery overhead (therefore is expected to be present in real systems), however they can introduce sampling biases as we will see later in Section 3.3 and Chapter 5.

The various internal sub-modules (*e.g.,* cellular monitor, WiFi Monitor and location monitor) are attached to a background "Intent" Service offered by Android APIs for such operations. This "NetworkPerformanceLoggingService" is also responsible for coordinating the local caching of the data and the forwarding to the central web server.

Storing, Uploading and Processing Measurements. The measurements are saved locally in an SQLite database, by utilizing an object-relational mapping (ORM) library for automated conversion between Java objects and Sqlite relational tables. ORM is an



0. SSID: UCInet Mobile Access RSSI: -65 Freq: 2437 (Active WiFi)

1. SSID: JoAsus RSSI: -37 Freq: 2437

2. SSID: JoAsus5G RSSI: -39 Freq: 5765

3. SSID: IOT RSSI: -77 Freq: 2457

4. SSID: microsemi-test RSSI: -80 Freq: 2422

5. SSID: ASUS_SimHome RSSI: -74 Freq: 2437

6. SSID: UCInet Mobile Access RSSI: -58 Freq: 2412

7. SSID: Dell_device RSSI: -62 Freq: 2462

Network Carrier:AT&T
Network Type: LTE
MCC:310
MNC:410
TAC34572
Cell-ID:141175902
LTE RSRP -111
LTE RSRQ -12

Network Carrier: AT&T
Network Type: LTE

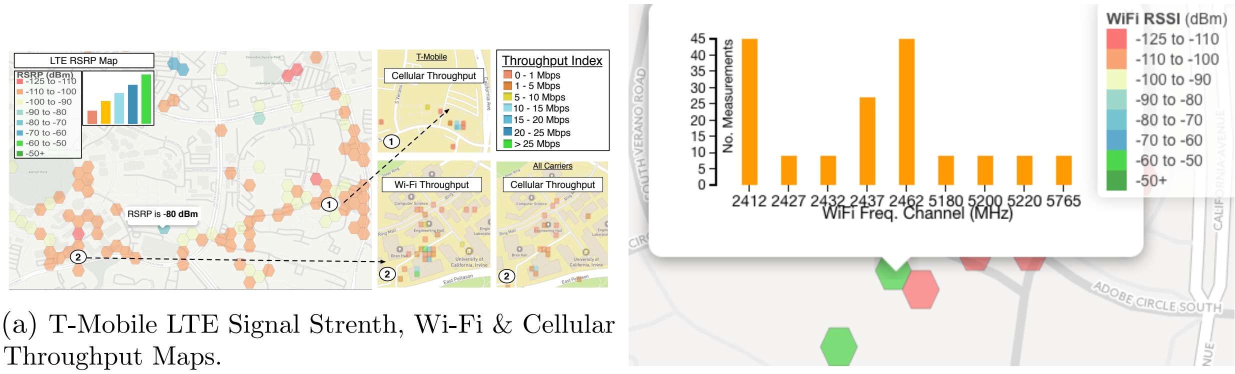
(a) Main Screen. (b) Nearby WiFi Info. (c) Cellular Connection. (d) UE's LTE RSRP Map.

Figure 3.3: AntMonitor passive network performance module user space app GUI examples.

middleware which automatically maps runtime Java objects to SQLite relational tables, thus provides easy and efficient data manipulation, without complex SQL statements. Our systems converts the collected data to Javascript Object Notation (JSON) format[4] and uploads them to a MongoDB on our server on the LogServer, per user's request or when the phone is charging and on WiFi.

MongoDB offers several advantages: it scales well, better than traditional database systems, it supports spatio-temporal operations, and allows schemaless storage, which is necessary given the heterogeneous parameters across devices. The data are stored for further off-line processing (*e.g.*, analysis, feature generation and machine learning for prediction for signal maps in the following chapters) and visualizations (*e.g.*, 3.4). We have anonymized the dataset by assigning a random id at the user's device, therefore, we cannot track back the original users. No personally identifiable information is stored or used in this thesis and the appropriately IRB exemption has been obtained.

[4] JSON is a lightweight data-interchange protocol widely used for communication by heterogeneous platforms.



(a) T-Mobile LTE Signal Strenth, Wi-Fi & Cellular Throughput Maps.

(b) WiFi RSSI & Freq. Channels Maps.

Figure 3.4: Performance maps from the university campus. Low RSRP (loc. 1) does not necessarily mean low cellular throughput (for the same carrier).

Signal Maps - LTE RSRP Measurements. This has been primarily the goal of our data campaign and our main use case; we collected the UCI `Campus dataset` in a period of 4 months, from 7 users in the UCIrvine campus. Fig. 3.2 depicts LTE RSRP maps for UCI campus for three distinct cells (*i.e.*, unique cID) with different characteristics. For example, Fig. 3.2a shows the LTE RSRP map for cell x355 which has small density and more dispersed data (*i.e.*, more uniformly distributed in space). On the contrary, Fig. 3.2b depicts a map for a cell that the collection was primarily performed on a specific location (higher density, less dispersed data). Apart from the LTE RSRP measurements RSS metrics from other wireless networked, when available, were collected as well, such as RSS data for 3G networks and RSSI, Frequency and WiFi SSID for WiFi network.

TCP/IP Layer & WiFi Passive Measurements: A secondary use case and a side benefit of our system is that we are capable of collecting *passively* TCP/IP layer measurements such as byte counts per TCP flow. First, we utilize our module to compute *passively* the smartphone's throughput and we compare it to a state-of-the-art *active* monitoring tool (`Speedtest`). Table 3.3 shows that the values are very close, but our passive approach does not incur any data overhead. Resources usage by these two methods is shown in Table 3.4. Fig. 3.4a also reports the average throughput of WiFi and LTE networks and compares it

Exp #	1	2	3	4	5	6	7
AM: W=5	21.24	29.26	22.83	27.01	30.75	26.84	26.14
Speedtest	19.96	28.42	22.39	28.74	31.66	26.98	27.22

Table 3.3: Throughput (Download Mbps): Active (using Speedtest) vs Passive (using AntMonitor: AM) measurements. First, we ran multiple Speedtests, with 5 min gaps, from the same location, and we list the throughput mentioned by Speedtest. Second, we computed the throughput using AntMonitor logs, over a window of 5 sec. Our approach is close to Speedtest but does not incur any measurement overhead. For a fair comparison in this table, we passively monitored the Speedtest packets using AntMonitor. In the wild, throughput computations can be made by counting the bytes of actual traffic sent over time.

to LTE RSRP. Interestingly, we observe that low RSRP does not necessarily result in low throughput. Fig. 3.4b depicts recorded WiFi measurements and the utilized WiFi frequency channels. Further measurements such as daily patterns of MB usage per user and per application are available in [6].

Metric	Data Overhead	Memory	CPU	Battery
Speedtest	50 MB	116 MB	14.7%	−0.5%
AntMonitor	0 MB	134 MB	43.4%	−0.7%

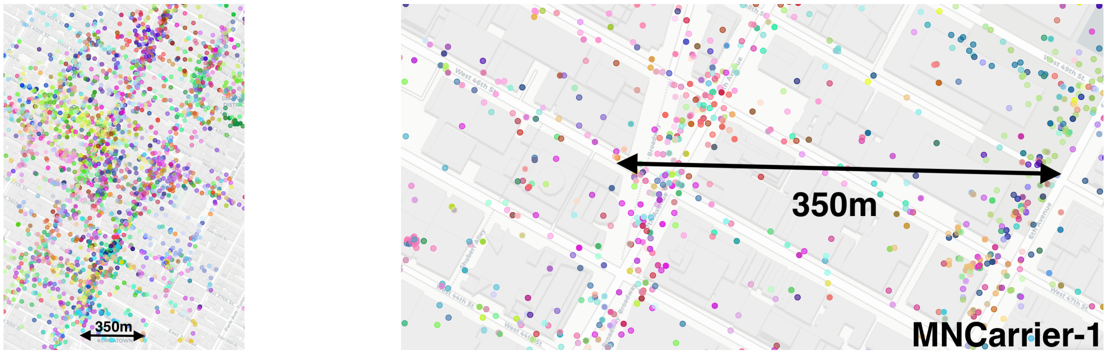
Table 3.4: Resources Utilization for AntMonitor and Speedtest per Exp.

3.3 NYC and LA datasets

3.3.1 Dataset Overview

We also use two much larger datasets generously provided to us by a Mobile Crowdsourcing Company: 10.9 million measurements in total, covering approx. 300km^2 and 1600km^2 in the metropolitan areas of NYC and LA, respectively, for a period of 3 months (Sep'17 - Nov'17). There are approx. $88,000$ and $111,000$ unique cell global identifiers (CGIs), as defined earlier, in the NYC and LA, respectively.[5] Other key characteristics are summarized in Table 3.1.

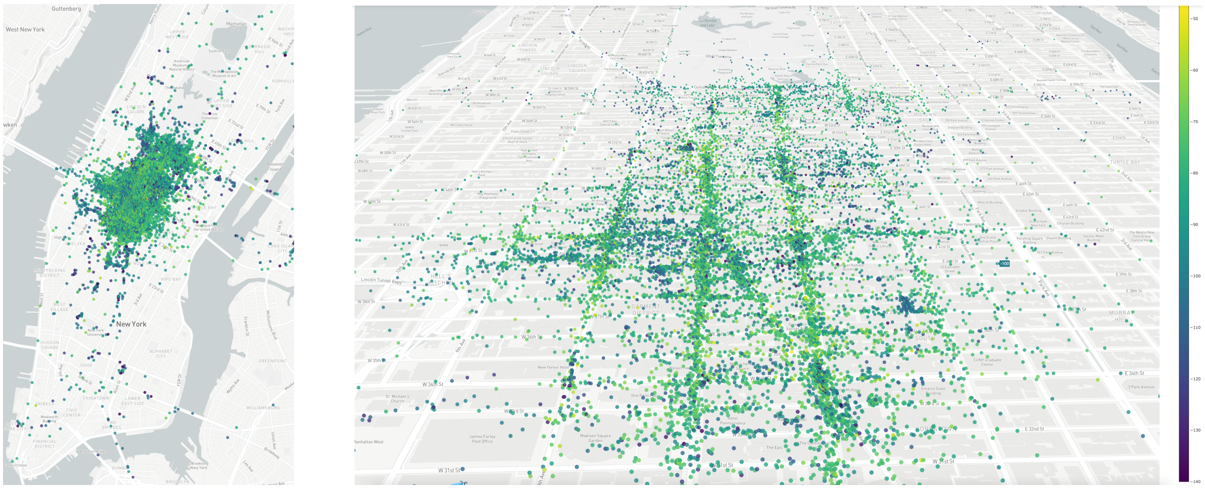
[5] It should be noted that many of these cells are either overlapping for extra capacity (*i.e.,* different network settings such as frequency *etc.*) and many of them are being served by the same cell tower (*i.e.,* different sectors). Moreover, cellular providers share their infrastructure with virtual providers (*i.e.,* MNVOs) which usually have unique MNCs and subsequently create new CGIs.



(a) NYC Manhattan LTE TA. (b) NYC: zooming in Manhattan Midtown (Time Square).

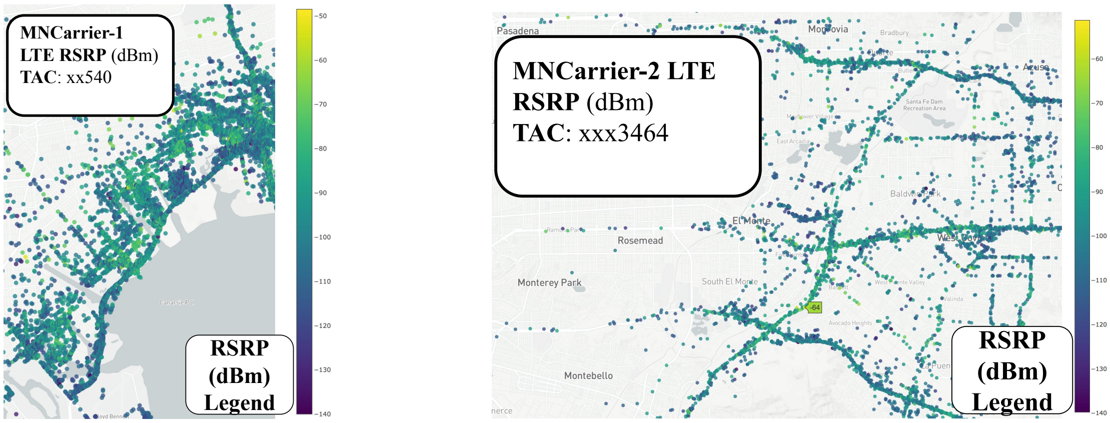
Figure 3.5: LTE RSRP Map Examples from NYC dataset, for a group of LTE cells in the Manhattan Midtown area. Different colors indicate different cell IDs (*i.e., cID*).

Examples of the data points locations from NYC Midtown Manhattan neighborhood are depicted in Fig. 3.5 and Fig. 3.6 respectively; the density of the particularly neighborhood lead the cellular operators to deploy many distinct cells as demonstrated in the figures. Signal maps examples are also shown in Fig. 3.7; Fig. 3.7a shows the RSRP values in Eastern NYC (Brooklyn) nearby JFK and Fig. 3.7b depicts the signal map, again for LTE RSRP, for the west part of San Gabriel Valley in LA metropolitan area. Please note that RSRP values are locally spatially correlated overall in terms of few blocks, however, there is also large variability across the map because of the random nature of the wireless channels.



(a) NYC Manhattan LTE TA-30000 Feet View. (b) NYC Coverage (RSRP) Maps: zooming in Manhattan Midtown (Time Square) for some of the available cells.

Figure 3.6: NYC dataset LTE RSRP Map Examples: Color indicates RSRP value



(a) East NYC nearby JFK: Example of Signal Map. (b) LA West San Gabriel Valley LA Signal Map.

Figure 3.7: NYC dataset LTE RSRP Map Examples: Color indicates RSRP value.

While these are large datasets and cover a big span of space and time, they are also relatively sparse in space; consider for example the density per cell (an average of 300 measurements per cell) or per cell tower (495 measurements per cell tower). There is also large heterogeneity across cells: we consider cells with more than 100 measurements and the maximum number of measurements per cell is 17424. Fig. 3.8 shows the CDF of the measurements per unique cID, as well as grouped by cell tower as defined above. Furthermore, there is sparsity in time: unlike the Campus dataset, there are fewer measurements for the same location. As we mentioned, no personal information or user identities are included in this dataset; our focus is on predicting signal strength and not on users.

To the best of our knowledge, these are the largest datasets used to date for signal maps prediction (*e.g.,* LTE RSRP), in terms of any metric (number of measurements, geographical scale, number of cells). As such, they provide novel insight into the problem at a scale that is relevant to operators and crowdsourcing companies, which is orders of magnitude larger than the scale previously considered in RSS prediction. Work in [19] uses 1500 locations samples from cellular networks for an area $\approxeq$ 15km^2, a university campus area in [59] and $\sim$ 1000 locations sampled at a $7km^2$ urban area in [39]. Work in [33] considers $20,000$ data points over approx. 20km^2 in Edinburgh. Work in [29] collected 10 million measurements but in

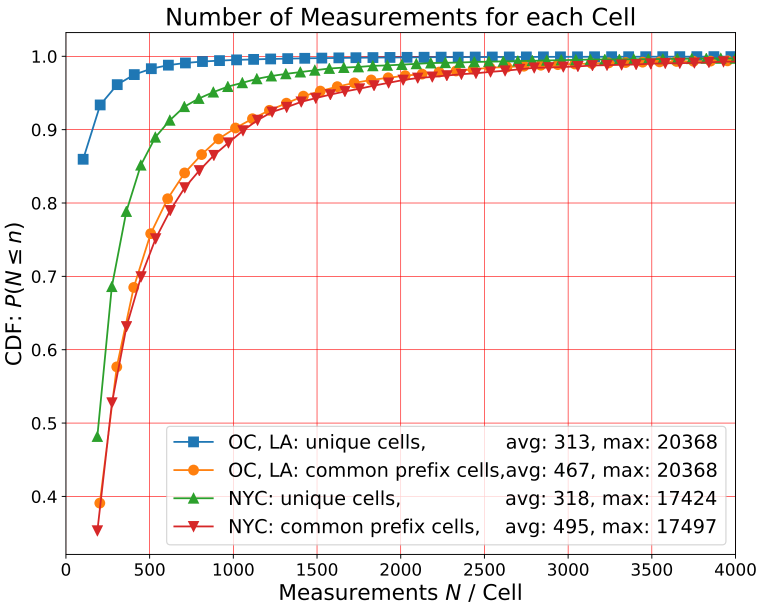


Figure 3.8: **`NYC and LA datasets`**: Number of measurements per cID and per cell tower (*i.e.,* cID with common prefix). We omit cells with less than 100 measurements.

much more limited geographical scale (3 neighborhoods in Dallas) and does not handle the effects of biased sampling as we do in this thesis, but rather focuses on minimizing the MSE.

3.3.2 Data Collection

This dataset was collected by a major mobile crowdsourcing and data analytics company and shared with us. RSS and other LTE KPIs have been collected through a measurement SDK, which is integrated into popular third party apps. The company crowdsources from a large user base, but they also try to collect measurements infrequently so as to not burden end-users, which explains the smaller overall density of the dataset compared to our **`Campus dataset`**, as it can be seen in Table 3.1. The observed data sparsity is a result of good sampling practices, *i.e.,* low overhead and battery usage for the users. Each location data point is accompanied by rich network and contextual information, except for device or other personal identifiers, which are not included in the dataset, for privacy-preserving reasons.

The details of the company's collection methodology are proprietary and not available to us. In Fig. 3.7b, we can see that commute traces (note the roads-highways trajectories) are over-sampled compared to nearby residential blocks where data are mostly absent. Similarly, for Eastern `NYC` in Fig. 3.7a the sampling density on the highway leading to JFK is much higher than nearby residential blocks (we refer to chapter 5 for more details on that). This sampling distribution seems similar to that observed in other crowdsourcing systems, *i.e.,* collect data when a notification/callback for a location change by *another* application takes place in Android OS. This strategy to collect data by piggy-backing on location changes notifications by other apps when the phone is power plugged, takes advantage of users commutes habits (*e.g.,* using Google Maps for GPS navigation) and offers a low energy footprint but can also lead to sampling biases.

The observation that even good measurement and sampling practices in crowdsourcing systems (*i.e.,* those minimizing user overhead, cost and battery usage) can lead to sampling bias and sparse measurements, further motivates our interest in developing techniques for predicting signal strength values from limited data (see Chapter 4). We also design techniques to handle the effects of sampling bias later (see Chapter 5).

3.4 Description of Datasets

For the purposes of signal maps prediction, we use the same subset of information from all datasets, *i.e.,* RSRP, RSRQ, CQI values and the corresponding contextual information - features defined earlier in Table 3.1. These features include LTE cell information, EARFCN (downlink LTE frequency channels), device hardware information, connectivity status, time and location. A comprehensive list of all the measurements' fields used in this thesis and their description are included in Table 3.5. It is worth mentioning that the two data sets have distinct attributes. On the one hand, although `Campus dataset` covers a smaller geographical

KPI	**GeoJson Schema Key**	**Description**
Timestamp	properties.time.timestamp	Timestamp of the measurement.
Timezone	properties.time.timezone	Timezone Recorded.
Latitude (l^x)	geometry.coordinates.1	Latitude reading by Android's Location APIs.
Longitude (l^y)	geometry.coordinates.0	Longitude reading by Android's Location APIs.
RSRP (y^P)	properties.lteMeasurement.rsrp	Reference Signal Received Power: The average received power in the reference LTE subcarriers.
RSRQ (y^I)	properties.lteMeasurement.rsrq	Reference Signal Received Quality: Interference indicator ≡ N x RSRP / RSSI.
CQI (y^C)	properties.lteMeasurement.cqi	Channel Quality Indicator for LTE connections, which considers several factors.
PCI	properties.lteMeasurement.cqi	Physical Cell Identifier: It is being utilized internally by the LTE protocol stack.
TA	properties.lteMeasurement.ta	Timing Advance: Signal's Time of Arrival reported by Android.
EARFCN ($freq_{dl}$)	properties.lteMeasurement.earfcn	LTE frequency band *a.k.a.* E-UTRA Absolute Radio Frequency Channel Number In LTE
MCC	properties.cell.mcc	Mobile Country Code.
MNC	properties.cell.mnc	Mobile Network Code.
TAC	properties.cell.tac	Tracking Area code: Unique identifier of a group of neighboring cells.
CID	properties.cell.cid	Cell-ID: Identifier of a cell in a TAC.
RAT	properties.connection.rat	Radio Access Technology: The current technology of the network (e.g. LTE).
Network Carrier	properties.connection.netCarrier	The name of the Cellular Provider.
Phone Connectivity	properties.connection.connectivity	Type of Internet Connectivity (Mobile or WiFi).
Device Model (dev)	properties.device.model	Hardware Name of the Device Model.
Android API	properties.device.api	Current version of Android API.
Android OS	properties.device.os	Name of the Android Operating System
Altitude	properties.locationMetaD.altitude	Altitude reading by Android's Location APIs.
Accuracy	properties.locationMetaD.accuracy	Accuracy of the location reported.
Speed (out)	properties.locationMetaD.speed	Moving speed of the device.
Bearing	properties.locationMetaD.bearing	Bearing of the device.

Table 3.5: Detailed Description of the Measurements of our Datasets: LTE KPIs (Key Performance Indicators), contextual information (time, location) and various other fields. Please note that the unique cell identifier cID consists of the concatenation of MCC, MNC, TAC, CID.

area and some fields contain limited or no data (*e.g.,* LTE CQI, EARFCN and location altitude), interestingly offers very dense samples on time and space. On the other hand, `NYC and LA datasets` cover a larger geographical area and offer a richer measurement collection including EARFCN, CQI, RSRQ *etc.* but the data are sparser.

Data Format. Measurements from both data sets are converted to GeoJSON format, which offers various advantages (lightweight JSON, compatibility with geospatial software, compact and intuitive representation of location information). A GeoJSON example with some of the KPIs fields (obfuscated) follows:

```
{”type”: ”Feature”, ”properties”: {
”timestamp”: ”2017−09−11T17:54:35EDT”,
”lteMeasurement”: {”rsrp”: −89,
                   ”rsrq”: −20, ”cqi”: 9,
                   ”pci”: 169, ”earfcn”: 9820},
”cell”: { ”ci”: xxxxx710, ”mnc”: 410, ”mcc”: 310,
          ”tac”: xx22,  ”networkType”: 4},
”device” : {”manufacturer”:”samsung”,
                   ”model”:”SM−G935P”, ”os”:”android70”},
”locationMetaData”: {”city”: ”New␣York”,
                    ”accuracy”: ”x”,”velocity”:”x”}},
”geometry”: {”type”: ”Point”, ”coords”: [−73.9xx, 40.7xx]}}
```

Listing 3.1: GeoJSON example with LTE KPIs and location, in MongoDB (obfuscated for presentation).

Properties of the Datasets. For each dataset, the following metrics describe characteristics that affected signal maps prediction, as shown later in this thesis.

- *Data Density:* This is the number of measurements per unit area, *i.e.,* $\frac{N}{m^2}$.
- *Cells Density:* Number of unique cells (*cid*s) per unit area, *i.e.,* $\frac{|C|}{\text{km}^2}$.
- *Dispersion:* In order to capture how concentrated or dispersed are the measurements in an area, we use the spatial distance deviation (SDD) metric, defined as the standard deviation of the distance of the data points from geometric mean center, *i.e.,* $(\overline{X},\ \overline{Y})$.

$$SDD = \sqrt{\frac{\sum_i^N (l_i^x - \overline{X})^2}{N} + \frac{\sum_i^N (l_i^y - \overline{Y})^2}{N}}$$

Higher SDD, means that geospatial points are more widely dispersed around the center.

OpenCellID. As we defined earlier in Table 3.1 and we will see later in the methodology, we need distance between the transmitting antenna and the receiver's location (where signal strength is measured or predicted), $||\mathbf{l}_{\text{BS}} - \mathbf{l}_j||_2$, in order to use it as a feature or in the prediction directly. To that end, we lookup the location of the base station, $\vec{l}_{\text{BS}}$, using the public APIs of a popular online crowdsourced database `opencellid.org`. This is the only external information we need in addition to the main RSS datasets.

3.5 Summary

In this chapter, we presented the LTE mobile network measurements datasets we use in this thesis, namely (i) the UCI `Campus dataset` and (ii) `NYC and LA datasets`. We also presented the crowdsourcing system we designed in order to collect `Campus` data and important lessons we learned throughout designing it. The first dataset is collected on a university campus of approx. 3km^2, contains cells with a wide range of characteristics and interestingly offers very dense samples on time and space. The second dataset consists of much larger datasets from `NYC` and `LA` metropolitan areas, which contain approx. 10 million LTE measurements and covering areas of approx. 300km^2 and 1600km^2 respectively. Although these datasets are much larger, they are also sparser in space and time and preliminary observations revealed the need to deal with sampling bias. Interestingly, it was a validation of what was expected considering our experience designing our own system; devices are typically collect and send data while plugged on power and GPS applications are pushing location updates (thus offer low battery footprint) leading to over-sampling roads and highways. We further described the common characteristics, contextual information and features we store for both of these LTE datasets.

Although the `NYC and LA datasets` are proprietary and cannot be released, we were lucky to be able to study them and obtain useful insights. We are in the process of releasing the `Campus dataset` (and the code we use to collect it) in order to allow the research community to experiment further with signal maps prediction or other tasks. To the best of our knowledge, the `Campus dataset` is among the largest publicly available data of RSS metrics for LTE networks. We also packaged our crowdsourcing system as an Android library and will make it open source for the community to maintain and evolve.

Chapter 4

City-Wide Mobile Coverage Maps Prediction with Random Forests

All Predictive Models are Wrong, but, some are useful and work.

George, Box

4.1 Overview

As we reviewed in the introduction and Chapter 2, mobile coverage maps are of great importance to cellular operators for network planning, however they are expensive to obtain, usually limited in scale, and possibly inaccurate in some locations. Apart from the mobile coverage maps by popular mobile analytics companies, there are myriad other applications for coverage maps. Examples include network management, maintenance, upgrades, and

The material in this chapter is based on the paper *"City-Wide Signal Strength Maps: Prediction with Random Forests"* by E. Alimpertis, A. Markopoulou, C. T. Butts and K. Psounis, published in the ACM Proceedings of World Wide Web Conference (WWW'19), May 13-17, 2019, San Fransisco, CA, USA. ACM, NY, USA, 7 pages, https://doi.org/10.1145/3308558.3313726. © 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

operations, *e.g.,* in order to determine if and where to deploy more cells, to identify problems and troubleshoot in self organizing networks SON, *e.g.,* [36, 41].

Our goal in this chapter is to improve the tradeoff between cost (number of measurements) and quality (*i.e.,* error) of signal maps via signal strength prediction from limited measurements. In general, as we reviewed in Chapter 2, there are two approaches for signal strength prediction: propagation models and geospatial interpolation. The latter is inherently limited to spatial features and does not take into account various critical aspects of the problem while the former requires extensively a priori modeling of the environment. Our approach falls in the broader data-driven category and we employ a powerful machine learning framework that naturally incorporates multiple features.

More specifically, the contributions of this chapter are the following:

1. Prediction framework based on Random Forests (`RFs`). We develop a powerful machine learning framework based on random-forests (`RFs`). We consider a rich set of features including, but not limited to, location, time, cell ID, device hardware, distance from the tower and frequency band; all of them affect the wireless properties and the calculation of the signal strength on the device. This is the first time that `RFs` have been applied to the coverage maps estimation problem. Prior work on data-driven prediction for signal maps was primarily based on geospatial interpolation techniques [50, 19, 59], which do not naturally extend beyond location features. To the best of our knowledge, this is the first time that location, time, device and network information are considered jointly for the problem of coverage maps prediction. We assess the feature importance and we find cell ID, location, time and device type to be the most important. We show that our `RFs`-based predictors can significantly improve the tradeoff between prediction error and number of measurements needed, compared to state-of-the-art data-driven predictors. They can achieve the lowest error of these baselines with 80% less measurements; or they can reduce the $RMSE$ (root mean square error) by 17% for the same number of measurements.

2. Device Hardware Information. Prior work has ignored important device, radio frequency and hardware's receiver information. First, each device calculates its signal strength (*e.g.,* LTE RSRP) differently (*i.e.,* proprietary algorithm in the device's cellular modem). Second, receiving sensitivity (the minimum RSRP for a feasible wireless communication) changes per device because each wireless receiver has different noise figure (NF) [5]. In this work we perform prediction per device (device hardware is used as a feature) and we incorporate different coverage thresholds per device (see Chapter 5), therefore we take into account both of the aforementioned phenomena.

3. Evaluation with large-scale real-world datasets. Our study leverages two types of real-world datasets: (i) a small but dense `Campus dataset` collected on a university campus; and (ii) several large but sparser `NYC and LA datasets`, provided by a mobile data analytics company. Examples are depicted in Fig. 1.4 and information about the datasets is provided in Table 3.1. We use these datasets to evaluate and contrast different prediction methods and gain insights into tuning our framework. For example, cell ID is an important feature in areas with high cell density, which is encountered in urban areas such as Manhattan Midtown; in contrast, cell ID should be used to train cell-specific `RFs` in suburban areas. Furthermore, time features are important in cells with less dispersed measurements, *i.e.,* concentrated in fewer locations. To the best of our knowledge, the `NYC and LA datasets` are among the largest used to date for RSRP (or other signal strength) prediction, in terms of any metric (number of measurements, geographical scale, number of cells *etc.*). They contain 10.9 million LTE data points in areas of 300km^2 and 1600km^2 for `NYC` and `LA` respectively, instead of at most tens of km^2 and tens of thousands of measurements in [33] or just three neighborhoods of Dallas [29] or smaller scale in [59, 50, 39, 19]. Thus, we provide novel insights into city-wide coverage maps prediction.

Outline. The rest of this chapter, is organized as follow. Section 4.2 recaps the coverage maps definitions and the problem statement for this chapter. Section 4.3 reviews the baseline prediction methods which reveals simultaneously fundamental properties of the wireless signal strength. In Section 4.4, we present our random forests-based approach and the rationale behind our modeling. In Section 4.5, we assess the feature importance for our framework and we provide evaluation results of our methodology compared to the state-of-the-art. Finally, Section 4.6, summarizes the findings of this chapter.

4.2 Problem Recap

We have already described and coverage maps problem domain and our goals in in Chapter 2 (see Sec. 2.1.3). To recap, a coverage map (*a.k.a.* signal map) is a collection of N measurements $(\mathbf{x_i}, y_i)$, $i = 1, 2...N$, where the label y_i is the signal strength measurement collected along with the features $\mathbf{x}_i$ that include the location, time, device, network information *etc.*

In this chapter, our goal is to develop a predictor for the missing values y_i, *i.e.,* to predict the signal (coverage) map value $y^P = y$ at a given location, time, and potentially considering additional contextual information (*i.e.,* the feature space $\mathbf{x}$; see recap in Table 3.1 and to be analyzed in Sec. 4.4.2), based on available measurement historical data either in the same cell cID or in the same LTE TA. In this chapter, we showcase prediction with LTE RSRP $y^P = y$, which is arguably the most important KPI for LTE networks assessments (*e.g.,* define coverage), however our predictors can be applied to any other signal strength (RSS) metric (*e.g.,* RSRQ or CQI in Chapter 5).

(1) Model Based (Radio Frequency Propagation Model)		1(a) $LDPL$ (Log Distance Path Loss Eq. 4.1)	1(b) $LDPL_{knn}$ (heterogeneous PLE)	1(c) WINNER I/II [16], COST 231 [25], Ray Tracing [76], Hata Model [66] *etc.*
Data Driven	(2) Geostatistics see Sec. 4.3.2	2(a) OK Ordinary Kriging	2(b) OKD OK Detrending	2(c) OKP, OKPD OK Partitioning Detrending
	(3) Random Forests	3(a) $RFs_{x,y}$ Spatial Features $\mathbf{x}= (l^x, l^y)$[61]	**3(b)** $RFs_{x,y,t}$ **Spatiotemporal** $\mathbf{x} = (l^x, l^y, d, h)$	**3(c)** RFs_{all} **Full Feats** $\mathbf{x} = (l^x, l^y, d, h, dev, cid, \lVert\mathbf{l}_{BS} - \mathbf{l}_j\rVert_2, freq_{dl}, out)$

Table 4.1: Overview of RSRP Prediction Methodologies evaluated in this chapter. Random Forests (RFs) methods proposed in this chapter are marked in bold. Methods in regular font are prior art, evaluated as baselines for comparison. Methods in light gray font are reviewed but not implemented in this thesis. Please also see Sec. 2.2 for a detailed review of prior work and more examples from each family of predictors and their limitations.

4.3 Background and Baseline Models

We begin by presenting the most representative state-of-the-art prediction methods, which will be used as baselines in this chapter. Table 4.1 summarizes the family of predictors for RSS which can be used to generate coverage maps as well as put our proposed work into perspective. There is a large literature on propagation models [16, 76, 25], which are reviewed in detail in Chapter 2 (see Sec. 2.2.1). They model the received signal strength given the location of receiver, transmitter and the propagation environment. As a representative baseline from the family of model-based predictors, we consider the Log Distance Path Loss (LDPL) propagation model, which is simple yet widely adopted in the literature. Additionally, it provides further understanding of the (i) low-level RSS statistical properties and (ii) the wireless network fundamentals that should be taken into account for the prediction task.

4.3.1 Model Based Prediction (LDPL)

The Log Distance Path Loss (LDPL) model predicts the power (in dBm) at location $\mathbf{l}_j$ at distance $||\mathbf{l}_{\text{BS}} - \mathbf{l}_j||_2$ from the transmitting basestation (BS) or cell tower, as a log-normal random variable (*i.e.,* normal in dBm) [60, 5]:

$$y_{cID}^{(t)}(\mathbf{l}_j) = P_0^{(t)} - 10 n_j \log_{10}\left(\frac{||\mathbf{l}_{\mathrm{BS}} - \mathbf{l}_j||_2}{d_0}\right) + \omega_j^{(t)}. \tag{4.1}$$

The most important parameter is n_j, *i.e.,* the path loss exponent (PLE), which has typical values between 2 and 6. $P_0^{(t)}$ is the received power at reference distance d_0, which is calculated by using the free-space path loss (Friis) transmission equation for the corresponding downlink frequency, gain and antenna directionality, and $\mathbf{l}_{\mathrm{BS}}$ the location of the transmitting antenna. In its simplest form, the equation assumes antenna reception gain and base station antenna gain equal to 0 dBi, but the application of an antenna directionality gain model as well as mobile gain model is also possible as shown in [5]. The log-normal shadowing is modeled by $\omega_j^{(t)} \sim \mathcal{N}(0, \sigma_j^2(t))$ (in dB), with variance $\sigma_j^2(t)$ assumed independent across different locations. The cell (identified by cell ID cID) affects several parameters in Eq. 4.1, including P_0, ω_j, the locations of transmitting ($\mathbf{l}_{\mathrm{BS}}$) and receiving ($\mathbf{l}_j$) antennas. The simplicity of this model lies in that it has only one parameter (the path loss exponent n_j) to be estimated from the measurements. Prior work [5] has shown that the PLE values and the time variant RSS variance ($\sigma_j^2(t)$) can be estimated by a large number of collected measurements. It should be noted, that in real world setups, base stations' transmission power changes according to the network load and conditions [5], contributing to the time varying component of the equation. We consider two cases.

`Homogeneous LDPL`: Much of the literature assumes that PLE n_j is the same across all locations. We can estimate it from Eq. (4.1) from all the training data points.

`Heterogeneous LDPL-knn`: Recent work (*e.g.,* [19, 5]) has considered that PLE changes across different locations. We considered various ways to partition the area into regions with different PLEs, and we present the one where we estimate $\widehat{n_j}$ via knn regression, from the k nearest neighbors, weighted according to their Euclidean distance (refer to as "`LDPL-knn`").

4.3.2 Geospatial Interpolation (OK-OKD)

State-of-the-art approaches in data-driven RSS prediction [19, 50, 59] have primarily relied on geospatial interpolation (*a.k.a.* geostatistics). However, this approach is inherently limited to predicting RSS from spatial features (l^x, l^y) and does not naturally extend to additional dimensions and contextual information. We refer to Chapter 2 (Sec. 2.2.2) for a detailed review of the related work. In this section, we present the best representatives of this family of predictors, namely ordinary kriging (OK) [50] and its variants OK detrending (OKD) [19], which are used as baselines for comparison in this chapter.

`Ordinary Kriging (OK)`: It predicts RSS at the testing location $\mathbf{l}_j = (l^x_j, l^y_j)$ as a weighted average of the K nearest measurements in the training set: $y_j = \sum_{i=1}^{K} \lambda_i y_i$. The weights λ_i are estimated by solving a system of linear equations that correlate the test with the training data via the semivariogram function $\gamma(h)$, which defines the variance between two different data points. Semivariogram $\gamma(h)$ must be estimated by the training data for different values of the lag h (*i.e.,* the distance between the data points) and each different environment requires different $\gamma(h)$; *i.e.,* distinct $\gamma(h)$ for `Campus` and `NYC and LA datasets`. The solution of the system is given by a Lagrange multiplier; more details for the derivation can be found in [19].

`Ordinary Kriging Partitioning (OKP)`: In [19], Voronoi-based partitioning was used to identify regions with the same PLE and apply a different OK model in each region. This is comparable to the heterogeneous propagation model. However, OKP solves separately a different optimization problem for each local subregion, which make them impractical for city wide signal maps.

`Ordinary Kriging Detrending (OKD) [19]`: While OK typically assumes the same mean value across locations[1], this does not necessarily hold for RSS values. OKD incorporates a simplified version of `LDPL` in the prediction in order to address this issue [19]. This can be thought as a hybrid approach of data-driven (geospatial) and model-driven (`LDPL`). It is the best representative of the geospatial predictors and serves as our baseline for comparison.

The basic steps of OKD are as follows: (i) OKD needs a transmitter location (we can use the strongest RSS algorirthm [73] or the `OpenCellID` online DB) and computes the perceived path-loss exponent n_i at each training data point ($n_i = P_0 - {y^P}_i / \log_{10} d_i$, where $d_i = ||\mathbf{l}_{\text{BS}} - \mathbf{l}_i||_2$). (ii) After computing the mean PLE $\widehat{n}$, OKD computes $L(y_i) = y_i - 10\widehat{n}\log_{10}(d_i)$ and the detrending component $\delta_i = y_i - L(y_i^P)$ for the training data points. (iii) Finally, OKD applies OK to predict δ_j by learning with the data δ_i and predicts δ_j.

4.4 Prediction with Random Forests (`RFs`)

4.4.1 Formulation

In this chapter, we leverage a state-of-the-art machine learning (ML) framework: Random Forests (`RFs`) regression. `RFs` is an ensemble of multiple decision trees [15], which provides a good trade-off between bias and variance by exploiting the idea of bagging. `RFs` first build multiple decision trees based on sub-samples of the training data and splits between nodes using a random sample of features. The random forest is the combination (average) of the individual trees. For regression, the objective is to minimize the MSE at the terminal leaf. A coverage map (signal) value y (*e.g.,* LTE RSRP) to be estimated can be modeled as follows,

[1] The first applications of geostatistics were interpolation for environmental measurements like humidity and temperature.

given a set of feature vectors $\mathbf{x}$.

$$y|\mathbf{x} \sim \mathcal{N}(RFs_{\mu}(\mathbf{x}), \sigma^2_{\mathbf{x}}) \tag{4.2}$$

where $RFs_{\mu}(\mathbf{x})$, $RFs_{\sigma}(\mathbf{x})$ are the mean and standard deviation respectively of the `RFs` predictor ($\equiv \widehat{f}_y(\mathbf{x})$). The total variance of the prediction is equal to $\sigma^2_{\mathbf{x}} = RFs_{\sigma}(\mathbf{x}) + \sigma^2_{RFs}$, where σ^2_{RFs} is the error (MSE) from the construction of the `RFs` itself (we refer to [38] for a detailed decomposition of the variance's terms). The final prediction $\widehat{y} = \widehat{f}_y(\mathbf{x}) = RFs_{\mu}(\mathbf{x})$ is essentially the MLE (maximum likelihood estimate) since we assume Gaussian distribution of the data[2]. Basically, the prediction is the mean of the training values at a terminal leaf node.

4.4.2 Features

Random Forests are a well-known and successful ML model, which have been used to facilitate UEs localization with RSRP measurements [61]. In this thesis, we exploit this powerful algorithm to create a rich framework for coverage maps (*i.e.,* predict missing LTE RSRP and other KPIs values) given the contextual information a cellular operators might be interested, modeled by the feature space $\mathbf{x}$. We already presented a summary of the features in the problem statement (Chapter 2 - Sec. 2.1) and in the datasets (Chapter 3 - Sec. 3.1); a coverage map value depends on a lot of different characteristics and the operators might be interested in some or all of them. Now, we present the rationale for incorporating each feature to the predictor as well as what are the important features on each scenario. For each measurement j in our data, we consider the following full set of features, available via the Android API:

$$\mathbf{x_j^{full}} = (l_j^x, l_j^y, d, h, cID, dev, out, ||\mathbf{l}_{BS} - \mathbf{l}_j||_2, freq_{dl}) \tag{4.3}$$

[2] As we already discussed in Sec. 4.3.1, signal strength values experience log-normal shadowing (*i.e.,* normal in dBm), therefore, the final conditional mean prediction of RFs can be the mean value of a Gaussian distributed r.v.

- *Location* $\mathbf{l}_j = (l_j^x, l_j^y)$. These are the spatial coordinates and the only ones considered by previous work on data-driven RSS prediction with geostatistics [19, 50] or in the context of localization [61, 49].

- *Time* features $\mathbf{t}_j = (d, h)$, where d denotes the weekday and h the hour of the day that the measurement was collected. Using h as a feature implies stationarity in hour-timescales, which is reasonable for signal strength statistics.

- *The cell ID*, cID. This is a natural feature since any signal strength value and KPI, such as LTE RSRP, is defined per serving cell cID (see Sec. 2.1.1 for the cID definition). The rationale is that the reception characteristics from one cell at one specific location might give information for the RSRP for other cells.

- *Device hardware* type, dev. This refers to the device model (*e.g.,* Galaxy9 or iPhone 11) and *not* to device identifiers. We consider this feature for several reasons. First, there are different noise figures (NF), *i.e.,* electronic interference, and reception characteristics across different devices. Second, the LTE KPIs calculation details differ across devices and manufacturers, since 3GPP just provides generic guidelines. Third, hardware manufacturing affects the mobile sensors output [4]. Fourth, each device has different receiving sensitivity, therefore, a different minimum RSRP threshold to be able to communicate over the wireless link.

- *The dowlink carrier frequency*, $freq_{dl}$. This is calculated by $EARFCN$ (E-UTRA Absolute Radio Frequency Channel Number). We consider this feature because radio propagation and signal attenuation heavily depend on $freq_{dl}$.

- *out* $\in \{0, 1\}$ is an approximate indicator of outdoors or indoors location, inferred from Android's GPS velocity sensor.

- *Euclidean distance* $||\mathbf{l}_{BS} - \mathbf{l}_j||_2$, of the receiver at location $\mathbf{l}_j$ from the transmitting BS (base station).

Among the above features, the cell ID is particularly important, as it will be demonstrated in Section 4.5.2. It turns out that when there is a large number of measurements with the same cID, it is advantageous to train a separate $\texttt{RFs}$ model per cID, using the remaining features:

$$\mathbf{x_j^{-cID}} = (l_j^x, l_j^y, d, h, dev, out, ||\mathbf{l}_{BS} - \mathbf{l}_j||_2, freq_{dl}).$$

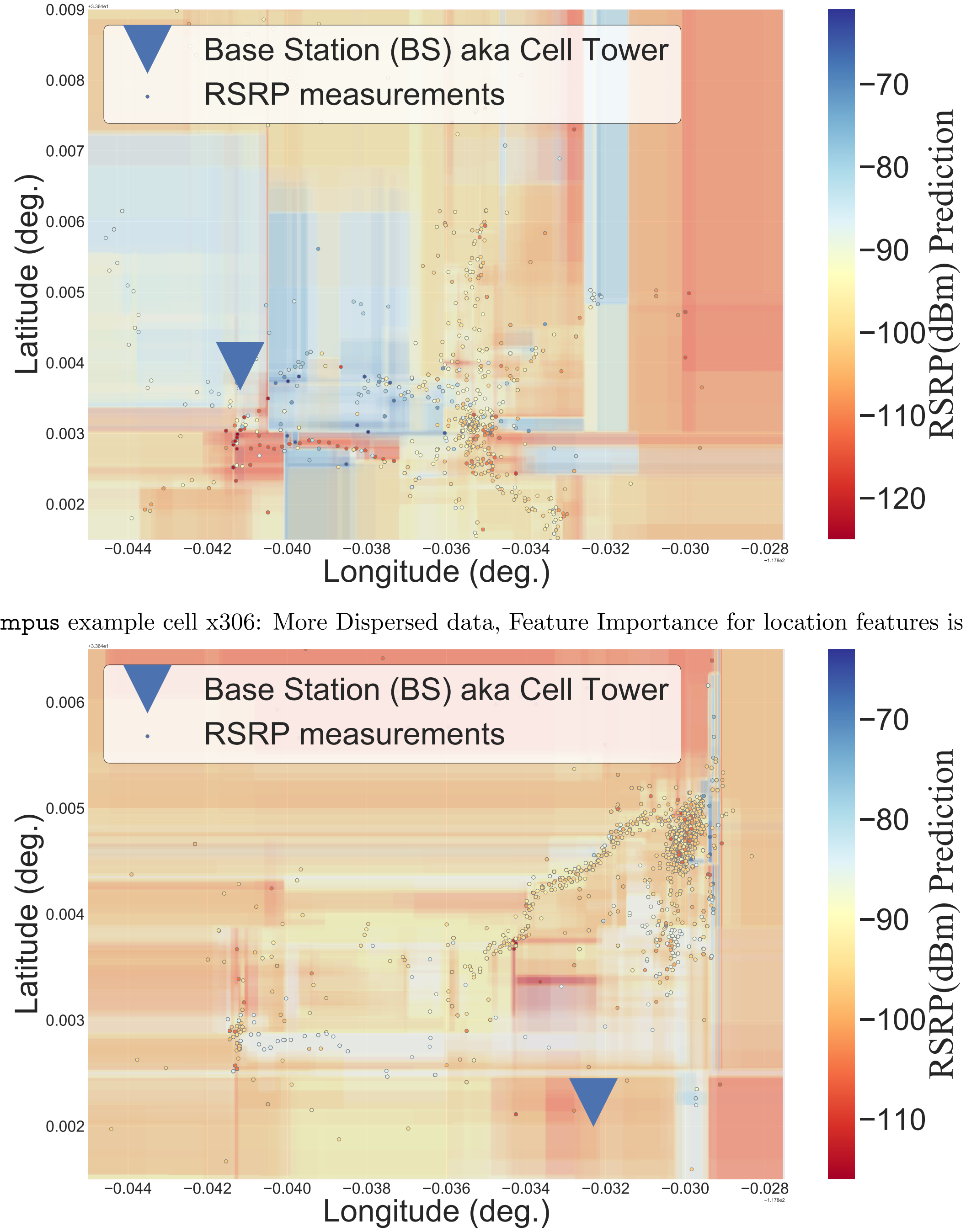
When there are a few measurements per cID, then we treat cID as one of the features in $\mathbf{x_j^{full}}$.

We denote as $\texttt{RFs}_{x,y}$, $\texttt{RFs}_{x,y,t}$, $\texttt{RFs}_{all}$ the $\texttt{RFs}$ predictors with only spatial (l^x, l^y), spatial (l^x, l^y) and temporal (d, h), and all features, respectively. In Section 4.5.2, we assess feature importance in different datasets, using tools inherent to the $\texttt{RFs}$ regression framework.

4.4.3 Why $\texttt{RFs}$ Prediction?

First, we selected $\texttt{RFs}$ regression because, $\texttt{RFs}$ can inherently incorporate all aforementioned features in Sec. 4.4.2, since geospatial interpolation [59, 78, 33, 19] does not naturally extend to arbitrary features. Second, $\texttt{RFs}$, by definition, partition the feature space with axis-parallel splits [51][3]. Examples of decision boundaries produced by $\texttt{RFs}_{x,y}$ (for LTE RSRP data) is depicted in Fig. 4.1. One can see the splits according to the spatial coordinates (lat, lng) and the produced areas agree with our knowledge of the placement and direction of antennas on campus. Essentially, these axis-parallel splits assume that measurements close in space, time most likely should be in the same tree node, which is a reasonable assumption for signal strength statistics. Automatically identifying these regions with spatially (and temporal) correlated RSRP comes for free to $\texttt{RFs}$ and is particularly important in RSRP prediction because wireless propagation has different properties across neighborhoods [61].

[3] For non-linear splits in space, *e.g.,* rivers, natural borders, hills, which would affect RSRP statistics, maps' terrain splits (shapefiles) can be used as extra features.



(a) `Campus` example cell x306: More Dispersed data, Feature Importance for location features is higher.

(b) `Campus` example cell x204. Less Dispersed data, Feature Importance for times features higher. Intuitively, we would expect the $\texttt{RFs}_{x,y,t}$ to perform better because where there are a lot of concentrated measurements, splits in time are needed to predict the dynamics of signal strength. Regardless, location features splits demonstrate the segmentation of space.

Figure 4.1: Example of decision boundaries chosen by $\texttt{RFs}_{x,y}$ for **(a)** `Campus` cell x306 and **(b)** x204. We can see that `RFs` can naturally identify spatially correlated measurements, *i.e.*, regions with similar wireless propagation characteristics. Color indicates RSRP value.

	Features			**Setup: Environment, Scale and Data**		
	Spatial	Time	Device & Network	Environment Agnostic	City-Wide	No Expensive LiDar Data
Log-Distance Path-Loss ($LDPL$) [5]	✓	✗	✗	✓	✗	✓
COST-231/ WINNER I-II/ Ray Tracing	✓	✗	✓	✗	✗	✓
Geostatistics SpecSense [19]	✓	✗	✗	✗	✗	✓
BCS [39]	✓	✓	✗	✗	✗	✓
RAIK-DNNs [29]	✓	✗	✗	✗	✓	✗
Our Work: Random Forests	✓	✓	✓	✓	✓	✓

Table 4.2: Signal Maps Approaches **Compared with Our Predictor** in this thesis.

In contrast, prior art (*e.g.*, OKP, [33, 19]) requires additional preprocessing for addressing this spatial heterogeneity; the area is splitted to disjoint areas with different interpolation parameters, with Voronoi diagrams, which is a problem with its own extra complexity. All the aforementioned advantages of our RFs prediction framework compared to prior-art and the incorporation of all features are summarized in Table 4.2.

4.5 Performance Evaluation

We evaluate all predictors of Section 4.4 (both state-of-the-art and our own RFs-based ones) over the datasets of Chapter 3. Along the way, we provide insights into the prediction performance and into tuning the framework depending on the characteristics of the dataset.

4.5.1 Setup

`RFs` **Setup.** `RFs` require less tuning compared to prior-art techniques (*e.g.,* estimating the parameters of the semivariogram [19], lag [19] and spatiotemporal correlation matrices per environment [39]). The most important hyper-parameters for `RFs` are the number of decision trees (*i.e.,* n_{trees}) and the maximum depth of each tree (*i.e.,* $\max_{depth}$). We used a grid search over the parameter values of the `RFs` estimator [58] in a small hold-out part of the data to select the best values. For the `Campus dataset`, we select $n_{trees} = 20$ and $\max_{depth} = 20$ via 5-Fold Cross-Validation (CV); larger $\max_{depth}$ values could result in overfitting of `RFs`. For the `NYC and LA datasets`,we select $n_{trees} = 1000$ and $\max_{depth} = 30$; more and deeper trees are required for larger datasets.

`RFs` **Model Granularity.** As argued in Sec. 4.4.2, one crucial design choice is what granularity we choose to build our `RFs` models: per cID or per LTE TA (defined in 2.1.1).

Training per cID: We can train a separate `RFs` model per cell (cID) using all features except cID ($\mathbf{x_j^{-cID}}$). This is natural since RSRP is defined per serving cell (see Sec. 2.1.1) but requires a large number of measurements per cell, which is the case in `Campus dataset` but not in `NYC and LA datasets`.

Training per LTE TA Another option is to train one `RFs` model per Tracking Area (LTE TA), and use cID as one of the features in ($\mathbf{x_j^{full}}$). This is particularly useful in the `NYC dataset`, where there are less measurements for the same cell unit area, insufficient to train a model per cID. However, in urban areas, there is very high cells density in a region and data points from different cells in the same LTE TA can still be useful.

In the next section, we consider the datasets and perform prediction at different granularities: (i) per cell (cID) (ii) per Tracking Area (LTE TA). Examples of representative LTA TAs used in our evaluation, are summarized in Table 4.3.

	NYC (MNC-1) Manhattan Midtown	NYC (MNC-1) E. Brooklyn	LA (MNC-2) Southern
No. Measurements	$\approxeq$ 63K	$\approxeq$ 104K	$\approxeq$ 20K
Area km^2	$1.8km^2$ (Fig. 1 (c-d))	44.8 km^2	220 km^2
Data Density $\frac{N}{m^2}$	$\approxeq$ 0.035	$\approxeq$ 0.002	$\approxeq$ 0.0001
No. Cells $\|C\|$	429	721	353
Cell Density $\frac{\|C\|}{km^2}$	238.3	16.1	1.6

Table 4.3: NYC and LA datasets: LTE TAs Examples.

Baselines' Setup. For LDPL *methods.* we do the following for the parameters of Eq. (4.1): we compute the distance from the base station using the online database from opencellid.org; breaking distance $d_0 = 1m$ [5]; $freq_{dl}$ is obtained from the $EARFCN$ measurement readily available via the Android API. [4]. In addition, for LDPL-knn: we select empirically $k = 100$ neighbors for the Campus dataset and $k = 10\%$ of the training data points in each cell for the NYC and LA datasets.

Geostatistics Predictors. The number of neighbors was empirically set to $k = 10$. For geospatial interpolation methods, a larger k did not show any significant improvement, and it would result in much higher computational cost. An exponential fitting function of the semivariogram function $\gamma(h)$ was selected [19]; the maximum lag (h) was set to 200m, as in [19], for the Campus and NYC environments, while it was set to 600m for the LA suburban environment. The approximated empirical semivariogram $\widehat{\gamma(h)}$ was calculated per $10m$ [19].

Splitting Data into Training and Testing. We select randomly 70% of the data as the training set $\mathcal{D}_{train} = \{\mathbf{X}_{train}, \mathbf{y}_{train}\}$ and 30% of the data as the testing set $\mathcal{D}_{test} = \{\mathbf{X}_{test}, \mathbf{y}_{test}\}$ for the problem of predicting missing signal maps values. The results are averaged over $S = 5$ independent random splits. These default choices are used unless otherwise stated. An exception is Fig. 4.5, where we vary the size of training set and we show that our RFs-based predictors degrade slower than baselines with decreasing training size.

[4] For the Campus dataset we got a limited number of $EARFCN$ measurements which indicated the most utilized frequencies in the area.

Evaluation Metrics. We evaluate the performance of the predictors in terms of absolute error (RMSE) and Relative Improvement (ARI) as well as feature importance in `RFs`.

Root Mean Square Error (RMSE): If $\widehat{y}$ is an estimator for y^P, then $RMSE(\widehat{y}) = \sqrt{MSE(\widehat{y})} = \sqrt{E((y-\widehat{y})^2)}$, in dB, since RSRP is reported in dBm. We report $RMSE$ for each predictor at different levels of granularity, namely: (i) per cID (ii) per LTE TA (in `NYC and LA`) or (iii) over the entire dataset (`Campus`).[5]

Absolute Relative Improvement (ARI): This captures the improvement of each predictor over the variance in the data: $ARI = 1 - \frac{1}{|\mathrm{C}|}\sum_{i\in\mathrm{C}}\frac{MSE_i}{Var_i}$, where |C| is the number of the different cells in the dataset, and Var_i is cell i's variance. Please note that (one of the simplest predictors would be the mean value over all data and its error would be the variance, therefore ARI encapsulates the improvement over the most minimal baseline.

Mean Decrease Impurity (MDI), a.k.a. Gini Importance: This essentially captures how often a feature is used to perform splits in `RFs`. It is defined as the total decrease in node impurity, weighted by the probability of reaching that node (approximated by the proportion of samples reaching that node), averaged over all trees in the ensemble [58].

Mean Decrease Accuracy (MDA), a.k.a. Permutation Importance: It measures the predictive power of each feature. The values of that feature are randomly permuted, *i.e.,* its predictive power is destroyed. Then we measure the decrease in the performance, when we predict with the remaining features and average over all trees in `RFs`.

[5] If we use `RFs` model per cell, denote $\widehat{y_j^{cid}}$ the prediction for the measurement j, with the dedicated `RFs` model for that specific cID. Then for each cell, $MSE_{cid} = \frac{1}{N_{cid}}\sum_j^{N_{cid}}(y_j^{cid} - \widehat{y_j^{cid}})^2$ while for all data points $MSE_{all} = \frac{1}{N}\sum_j^{N}(y_j^{cid} - \widehat{y_j^{cid}})^2$.

4.5.2 Results

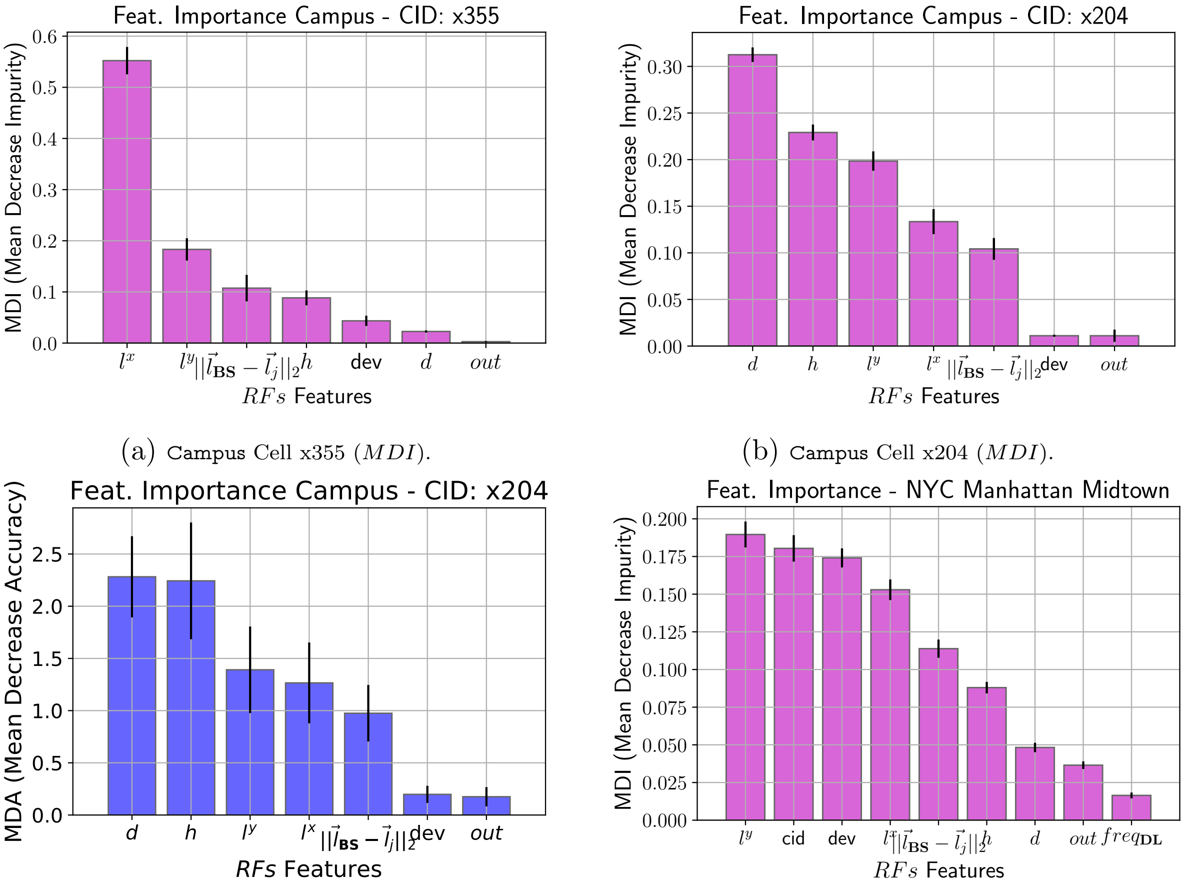
For the experimental evaluation, we report results with LTE RSRP coverage maps (as defined earlier) however the predictors could be used for any other RSS metric as shown in Chapter 5.

Feature Importance.

We begin our evaluation with the report of feature importance in Fig. 4.2.

a. `Campus dataset`*:* We train one `RFs` model per cID for the set of features $\mathbf{x} = (l_j^x, l_j^y, d, h, ||\mathbf{l}_{\text{BS}} - \mathbf{l}_j||_2, out, dev)$. We assess their importance w.r.t. MDI and MDA and representative results are shown on Fig. 4.2. We observe that, in cells with high data density and low dispersion, the most important are the time features (d, h) w.r.t. to both metrics. An example of such a cell is x204, which has $SDD = 325$, density=0.66 points/m^2 and is depicted in Fig. 3.2b). We see that (d, h) are the top features for this cell w.r.t. both MDI and MDA, as shown in Fig. 4.2b and Fig. 4.2c, respectively. For the rest of the thesis, we only report feature importance w.r.t. MDI. We also inspected the decision trees produced and these features are indeed being used at the higher levels of the decision trees. On the contrary, for more dispersed and less dense cells, such as cell x355 ($SDD = 573, 0.116N/m^2$, map in Fig. 3.2a), the location (l_j^x, l_j^y) is naturally the most important, as confirmed in Fig. 4.2a. Feature importance for dev and out are close to zero, which is expected because of the small number of devices in the `Campus dataset`. These results show that `RFs` can handle a diverse set of datasets with different characteristics, by splitting nodes according to the most important features.

b. `NYC and LA datasets`*:* In this case, $freq_{dl}$ is available and the datasets contain thousands of cells. We start with a `RFs` model per LTE TA. As a representative example, we report the feature importance, in Fig. 4.2d, for the LTE TA of a major mobile network carrier (MNCarrier-1) located in `NYC` Midtown Manhattan and already depicted in Fig. 3.5a-3.5b.



(a) Campus Cell x355 (MDI).

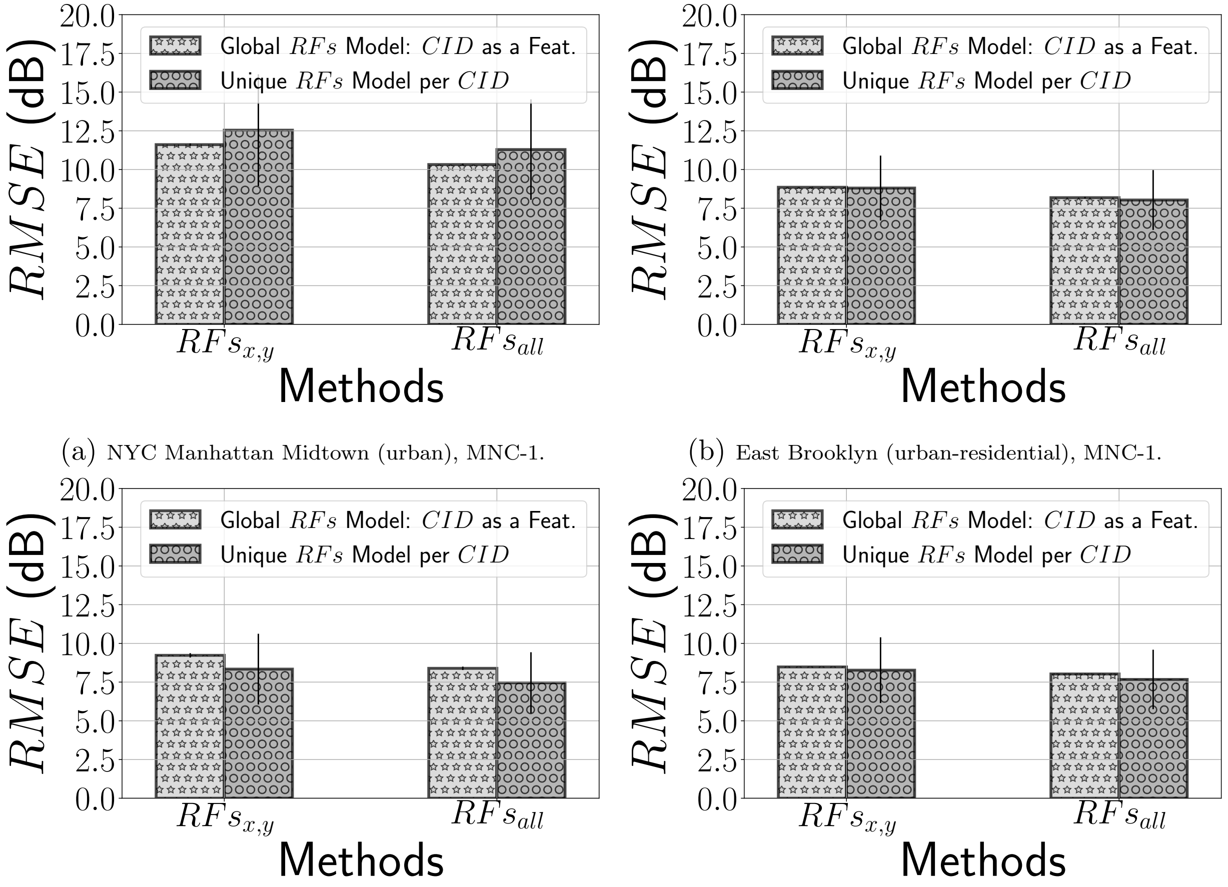
(b) Campus Cell x204 (MDI).

(c) Campus Cell x204 (MDA).

(d) NYC Manhattan Midtown LTE TA.

Figure 4.2: For `Campus dataset` (a), (b), (c): Feature Importance for two distinct cells (`RFs` built per distinct cell). Cells' data are depicted in Fig. 3.2. For `NYC dataset`, (d) shows the MDI score for one LTE TA for MNC-1. LTE TA's data are shown in 3.5.

The most important features turn out to be the spatial features (l^x_j, l^y_j) as well as the cell cID and the device dev. This is because the data are sparser and the whole LTE TA is served by geographically adjacent or overlapping cells; although RSRP is defined per serving cell (Sec. 2.1.1), the receptions characteristics at a specific location give information for the statistics of the RSRP for other cells and the `RFs` predictor is capable of encapsulating this information. The device hardware, dev, seems also important, because of (i) the heterogeneity in the devices reporting data in `NYC dataset` and (ii) the different RSRP calculation algorithm per device (see Sec. 2.1.1). We also investigated whether we should train a separate `RFs` per cID, or cID should be used as one of the features in a single `RFs`.



(a) NYC Manhattan Midtown (urban), MNC-1.

(b) East Brooklyn (urban-residential), MNC-1.

(c) Southern LA (suburban), MNC-2.

(d) Southern LA (suburban), MNC-1.

Figure 4.3: RMSE in `NYC and LA datasets`. This figure makes multiple comparisons: (1) urban vs suburban LTE TAs; (2) cID as feature vs. training a different `RFs` model per cID (*i.e.,* granularity of the models); (3) providers MNC-1 vs. MNC-2.

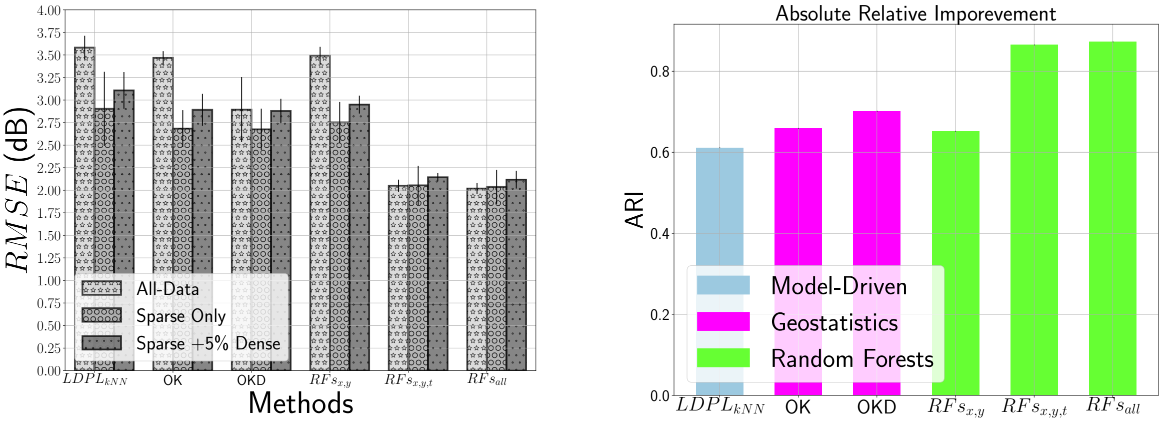
For a representative urban LTE TA (Manhattan Midtown), in Fig. 4.3a we calculate the $RMSE$ for two cases: (i) when cID is used as a feature in a single `RFs` per LTE TA and (ii) when a separate `RFs` model is produced per cell. Interestingly, the prediction is better when cID is utilized as a feature. Given the sparsity of the data and the high overlap of the cells, `RFs` benefit from the features of the additional measurements. Manhattan Midtown has a cells density of 238 per km^2 at it can be seen in Table 4.3: the cell size does not exceed the size of a few blocks or sometimes there are even multiple cells within a skyscrapper. On the contrary, for the suburban `LA dataset`, where the cells are not so densely deployed, a unique `RFs` model per cell performs better than `RFs` per LTE TA, as shown in Fig. 4.3c.

	Cell Characteristics					$RMSE$ (dB)						
cID	N	$\frac{N}{\text{sq } m^2}$	SDD	$\mathbb{E}[y^P]$	σ^2	LDPL hom	LDPL $_{kNN}$	OK	OKD	$RFs_{x,y}$	$RFs_{x,y,t}$	RFs_{all}
x312	10140	0.015	941	-120.6	12.0	17.5	1.63	1.70	1.37	1.58	0.93	**0.92**
x914	3215	0.007	791	-94.5	96.3	13.3	3.47	3.59	2.28	3.43	1.71	**1.67**
x034	1564	0.010	441	-101.2	337.5	19.5	7.82	7.44	5.12	7.56	3.82	**3.84**
x901	16051	0.162	355	-107.9	82.3	8.9	4.60	4.72	3.04	4.54	1.73	**1.66**
x204	55566	0.666	325	-96.0	23.9	6.9	3.84	3.85	2.99	3.83	2.30	**2.27**
x922	3996	0.107	218	-102.7	29.5	5.6	3.1	3.16	2.01	3.10	1.92	**1.82**
x902	34193	0.187	481	-111.5	8.1	21.0	2.60	2.47	1.64	2.50	1.37	**1.37**
x470	7699	0.034	533	-107.3	16.9	24.8	3.64	2.73	1.87	2.78	1.26	**1.26**
x915	4733	0.042	376	-110.6	203.9	14.3	7.54	7.39	4.25	7.31	3.29	**3.15**
x808	12153	0.035	666	-105.1	7.7	4.40	2.41	2.42	1.60	2.34	1.75	**1.59**
x460	4077	0.040	361	-88.0	32.8	11.2	2.35	2.28	1.56	2.31	1.84	**1.84**
x306	4076	0.011	701	-99.2	133.3	18.3	4.85	4.30	2.80	3.94	3.1	**3.06**
x355	30084	0.116	573	-94.3	42.6	9.3	2.42	2.31	1.85	2.26	1.79	**1.79**

Table 4.4: Campus dataset: Comparing Predictors per cell.

Very interestingly, prediction in Brooklyn is somewhere in the middle of these two cases and the performance per cell is approx. equal to the performance of the prediction per LTE TA (See Fig. 4.3b); Brooklyn neighborhood is denser than the suburban LA but there are definitely a lot of residential areas where there is not much overlapping of the cells.

Likewise in the Campus dataset (higher data density than NYC) RFs model per cID did better than using as a feature in a single RFs model for the entire LTE TA; even if there is overlapping for some of the cells there are usually from different cellular operators therefore a lot of the wireless network setup is fundamentally different: different transmitters' locations and power control *etc.* Most importantly, there are so many measurements from the same cell (up to approx. 50 thousands for the cell x204) that the model has a lot of information already. In general, RFs trained per cID is usually a better option, but cID should be used as a feature in urban areas with high cells density. Furthermore, the features with the least score could be omitted for computational costs and to avoid overfitting.



(a) $RMSE$ under various scenarios (lower is better). (b) Absolute Relative Improvement (higher is better).

Figure 4.4: Comparison of all predictors over the entire Campus dataset (all points, all cells), for all the methods (see Tab. 4.1). Left (a) $RMSE$(dB) under various scenarios, Right (b) ARI over all data points. Our Approaches ($\texttt{RFs}_{x,y,t}$, $\texttt{RFs}_{all}$) outperform prior art in all scenarios.

Comparing Coverage (RSRP) Maps Predictors. We compare the performance of the RFs prediction framework against state-of-the-art geospatial interpolation techniques (OK and OKD) as well as model-driven techniques (LDPL-knn and LDPL-hom).

a. Campus dataset*:* Table 4.4 reports the $RMSE$ for all predictors for each cell in the Campus dataset, and for the default 70-30% split. Fig. 4.4 compares all methods but calculating RMSE over the entire Campus dataset, instead of per cell. In both cases, we can see that our RFs-based predictors outperform model (LDPL) and other data-driven (OK, OKD) predictors, as long as they use more features than just location.

Fig. 4.5 shows the $RMSE$ as a function of the training size (as % of all measurements in the dataset). First, the performance of OK and $\texttt{RFs}_{x,y}$ is almost identical, as it can be seen for $RMSE$ over all measurements (Fig. 4.5 and Fig. 4.4) and $RMSE$ per cell (Table 4.4). This result can be explained by the fact that both predictors are essentially a weighted average of their nearby measurements, although they achieve that in a different way: OK finds the weights by solving an optimization problem while $\texttt{RFs}_{x,y}$ uses multiple data splits in multiple decision trees which are averaged at the end. Essentially, the RFs is a weighted

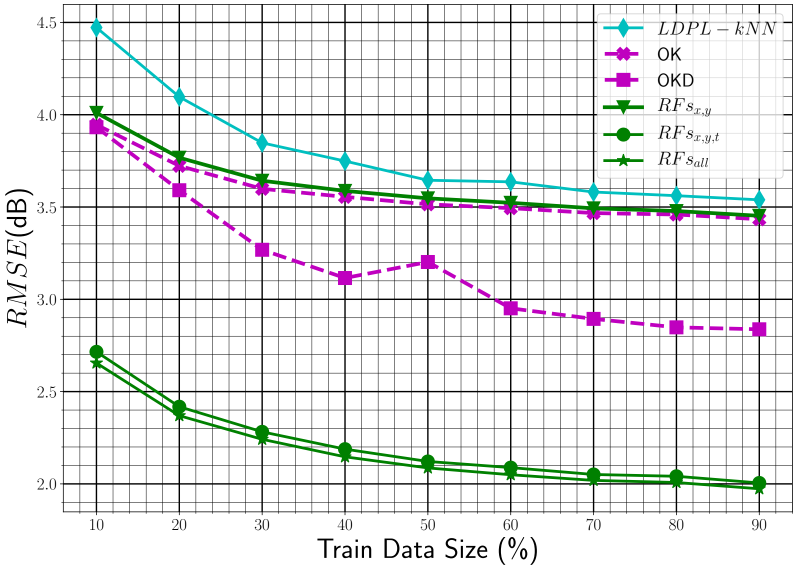
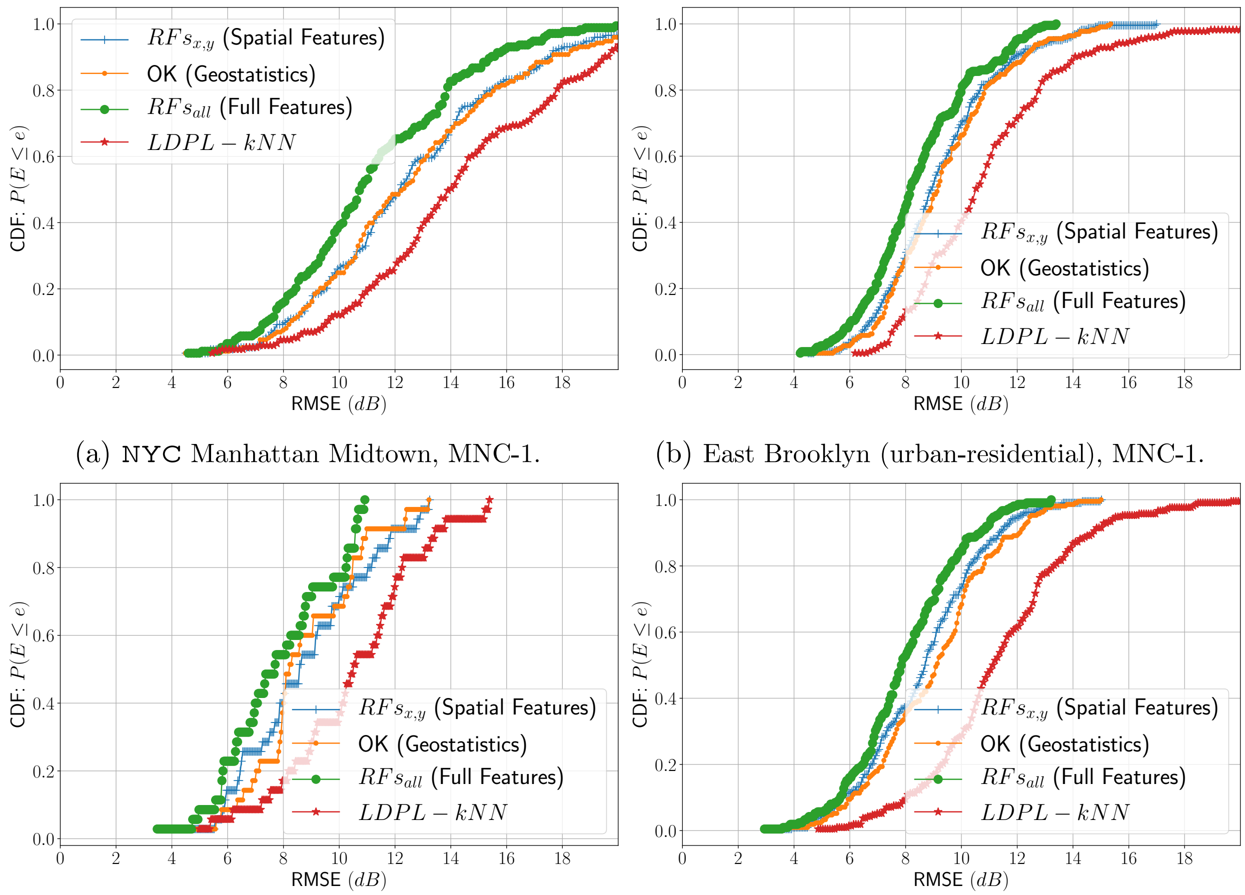


Figure 4.5: **Campus dataset**: $RMSE$ vs. Training Size Trade-off. Our methodology (**RFs** with more than spatial features, *i.e.,* $\mathtt{RFs}_{x,y,t}$, $\mathtt{RFs}_{all}$) significantly improves the RMSE-cost tradeoff: it can reduce $RMSE$ by 17% for the same number of measurements compared to state-of-the-art data-driven predictors (OKD); *or* it can achieve the lowest error possible by OKD ($\simeq$ 2.8dB) with 10% instead of 90% (and 80% reduction) of the measurements.

neighborhood scheme as shown in [48].

Second and more important, considering additional features can significantly reduce the error. For the **Campus dataset**, when time features $\mathbf{t} = (d, h)$ are added, $\mathtt{RFs}_{x,y,t}$ significantly outperforms OKD: it decreases $RMSE$ from 0.7 up to 1.2 dB. Alternatively, in terms of training size, $\mathtt{RFs}_{x,y,t}$ needs only 10% of the data for training, in order to achieve OKD's lowest error ($\simeq$ 2.8dB) with 90% of the measurement data for training. Our methodology achieves the lowest error of state-of-the-art geospatial predictors with 80% less measurements. The absolute relative improvement of $\mathtt{RFs}_{x,y,t}$ compared to OKD is 13%, as shown in Fig. 4.4b. ARI, defined in Sec. 4.5.1 can be considered as an overall improvement score, which mitigates the effect of the different' variance and scale of the collected measurements per each distinct cell (reported in Table 4.4).

b. **NYC and LA datasets***:* Fig. 4.6 shows the error for the same four LTE TAs used in Fig. 4.3, namely for **NYC** Manhattan Midtown (urban), for Eastern **NYC** Brooklyn (urban-



(a) NYC Manhattan Midtown, MNC-1.

(b) East Brooklyn (urban-residential), MNC-1.

(c) Southern LA (suburban), MNC-2.

(d) Southern LA (suburban), MNC-1.

Figure 4.6: **NYC and LA datasets**: CDFs for $RMSE$ per cID for four different LTE TAs, for two major Cellular Operators (*a.k.a.* MNCarrier). $\texttt{RFs}_{all}$ offer 2dB gain over the baselines for the 90th percentile.

residential) and for southern LA (suburban), where RFs have been trained per cID. CDFs of the error per cID of the same LTE TA are plotted for different predictors. Again, OK performance is very close to $\texttt{RFs}_{x,y}$, because they both exploit spatial features. However, $\texttt{RFs}_{all}$ with the rich set of features improves by approx. 2dB over the baselines for the 90th percentile. Interestingly, the feature dev is now important (see discussion in Fig. 4.2d), which is expected in this crowdsourced data with high heterogeneity of devices reporting RSRP.

Limitations of Geospatial Interplolation. There are multiple reasons why $\texttt{RFs}_{all}$ outperform geospatial interpolation (*a.k.a.* geostatistics) predictors. First, geostatistics techniques such as OK, work optimally when the random process is second-order stationary, which means (i) constant mean and (ii) the semivariogram of OK depends only on its lag (*i.e.,* the distance between two locations). However, the LTE RSRP's (and other RSS metrics) mean and variance depend on a plethora of phenomena such as complex wireless-propagation environment, time varying cells' parameters (*e.g.,* transmitted power [5]) *etc.*; apparently the correlation between two locations does not depend only on the distance. As we already shown in Fig. 4.1, RFs can easily capture all this complexity from the data, instead of modeling everything a priori. Second, even the more advanced OKD cannot naturally incorporate the influence of additional features (*e.g.,* time, device type, etc.), as shown in the numerical results. For example, $\texttt{RFs}_{x,y,t}$ predicts a time-varying value for the measurements at the same location in Fig. 3.2b, while $\texttt{RFs}_{x,y}$ or OK/OKD produce just a flat line over time or for NYC dataset cannot harvest the information from the heterogeneity of the devices (*i.e.,* different calculation of RSRP per device as we in Sec. 2.1.1 and 4.4.2). Last but not least, geostatistics methods have inherently *technical* limitations. For example, the methods cannot consider multiple measurements on the *same* location. More specifically, OK equations assume that the matrix of the measurements (to be precise, the semivariogram of the measurements) is an invertible matrix. However, if there are multiple measurements on the same location with the same value, this does not hold, since matrix with duplicate rows yield a matrix with determinant zero and therefore the matrix is not invertible. Given that we could have multiple measurements on the same GPS coordinates, we loose important information if we omit them or just take the average.

Assessing location density and overfitting. In the Campus dataset, we observed that a significant fraction of the data comes from a few locations, *i.e.,* from participating grad students' home and work. In other words, many data points were reported from the same or nearby locations over time, which begs the question whether this leads to overfitting of RFs to those oversampled location. We investigated this question and found that our

	Data Characteristics		RMSE(dB)					
cID	N	SDD	LDPL-knn	OK	OKD	RFs x, y	RFs x, y, t	RFs all
x312	4852	1240	1.66	1.49	1.46	1.62	1.06	0.91
x914	858	922	5.08	4.94	5.09	5.04	3.38	3.33
x034	514	532	6.94	6.52	6.59	6.6	5.52	5.25
x901	1549	218	3.07	2.79	2.86	2.9	1.9	1.97
x204	13099	535	2.53	2.48	2.46	2.57	1.93	1.93
x922	1927	309	3.62	3.66	3.56	3.66	2.13	2.17
x902	7589	245	2.45	2.06	1.89	2.08	0.92	0.92
x470	1357	431	3.72	0.75	1.51	0.79	0.48	0.52
x915	785	345	5.17	4.81	4.78	4.94	4.27	4.34
x808	5655	972	2.43	2.36	2.41	2.46	1.95	1.82
x460	1176	347	3.35	3.38	3.47	3.43	3.23	3.23
x306	1382	1131	5.84	5.15	5.13	5.34	4.14	4.3
x355	15356	790	2.68	2.54	2.53	2.58	2.04	2.03

Table 4.5: Campus dataset: Comparing Predictors per cell, considering only sparse measurements (*i.e.,* after removing measurements which create dense clusters).

RFs predictors neither get an "artificial" performance boost nor overfit. To that end, we utilize HDBSCAN [17], a state-of-the-art clustering algorithm, to identify very dense (spatially) clusters of measurements (cluster size 5% of the cell's data). We refer to data from those locations as "dense"; we remove them and we refer to the remaining ones as "sparse-only" data. Fig. 4.4a reports the $RMSE$ of different methods when training and testing is based on (i) all-data, (ii) sparse-only data and (iii) sparse-only data with a 5% randomly sampled from the dense data. It can be clearly seen that our $\texttt{RFs}_{x,y,t}$ and $\texttt{RFs}_{all}$ have similar performance in all scenarios and consistently outperform baselines. Please note that OK and LDPL-knn's errors slightly decrease for "sparse-only"; OK cannot handle repeated locations and LDPL-knn may overfit, but are still higher than our error. Table 4.5 reports the error per cell for sparse-only data, and our proposed predictors outperform baselines in a cell-by-cell basis.

Lessons from different datasets. When possible, we already provided insights w.r.t. the characteristics of the datasets (*e.g.,* different density and dispersion) and their effect on feature selection and selection of training block of RFs (cID or LTE TA). We would like to further discuss the effect of the wireless propagation environment in the prediction error. On the one hand, the Campus dataset has an average error of 2.2 dB while on the other hand the NYC LTE TA for Manhattan Midtown (see Table 4.3) has an average $RMSE$ of approximately 10dB (see Fig. 4.3a). The former is a suburban campus with very dense measurements in a small area, while the latter exhibits very harsh wireless propagation conditions because of Manhattan's skyscrapers, large number of people *etc.* It should be noted that the data density (number of measurements per m^2) is comparable (*e.g.,* $\simeq 0.035$ for both NYC Midtown LTE TA and cell x808 in Campus), the Campus dataset has 180 thousand measurements for 13 cells, while LTE TA for NYC Midtown has approx. 63K measurements for 429 cells, thus less measurements per cell. For less harsh propagation environments such as Brooklyn (Fig. 4.3b) or suburban LA (Fig. 4.3d), the error decreases to approx 7.5dB, *i.e.,* within the range of one signal bar.

Importance of RSRP Prediction and Magnitude of Error. We argue that the reduction in prediction error ($RMSE$ is on the order of a few dB for the Campus dataset and 7-11dB in NYC and LA datasets) is significant for cellular operators, notably in areas with low (*i.e.,* borderline) coverage. It should be emphasized, that RSRP values are not only used to determine signal bars (*e.g.,* coverage maps), but also determine the performance of voLTE (voice over LTE), which gives us a crucial real world example. Work in [43] showed that the probability of a call drop in VoLTE is approx. $2-5\%$ for RSRP $\in [-105, -110)$dBm, increases to $5-10\%$ for $[-115, -120)$ and more than 15% for RSRP ≤ -120dBm. Apparently, dropped calls are one of the main customers' complaints regarding operators, therefore, a reduction in error by a few dB from our methods is critical to accurately identify such regions.

Most importantly, by carefully inspecting the real world implications of RSRP for the users and the operators, we recognize the non-linear relationship between the signal strength and the Quality of Service (QoS) and realize that *certain* values of signal strength matter more than others. Our methods already perform well for these values (*e.g.,* $\texttt{RFs}_{all}$ has low error of $\approxeq$ 1.5 dB for cells x470, x902, see Tab. 4.4 and $\mathbb{E}[y^P]$), nonetheless, might waste predictive power treating all RSS values equally (*e.g.,* -105 to -120 dBm *vs.* -80 to -90dBm), since our `RFs` regression method, as well as existing literature, minimizes the standard mean squared error (MSE). This will be the topic of the next chapter of this thesis, which among other improvements shows that by identifying quality functions based on signal strength, can improve coverage maps for the low reception regime even further.

4.6 Summary

In this chapter, we developed a machine learning framework for cellular signal strength (LTE RSRP) prediction, which is important for creating mobile coverage maps in a cost-efficient way, crucial for future 5G and IoT deployments. We used the powerful tool of random forests (`RFs`), which we adapted in this context for the first time by evaluating different features readily available by Android APIs. We demonstrated the following contributions:

- We conclusively showed that the `RFs`-based predictors outperform state-of-the art data-driven predictors (geospatial interpolation) in all scenarios, when more features beyond just location are considered. We showed that the most important features were primarily cID, location, time and device type, which none of them can be naturally incorporated to geostatistics.

- We can significantly improve the tradeoff between prediction error and number of measurements needed compared to the state-of-the-art, *i.e.,* require 80% less data for the same error, or reduce the relative error by 17% for the same number of measurements.

- We showed how device hardware information is very important because of (i) the different signal strength calculation per device and (ii) the different NF per device; for example, device hardware *dev* was one of the most important features in `NYC dataset`.

- The datasets under study such as the dense `Campus` and `NYC and LA` datasets with approx. 11 million LTE Measurements, are among the largest used in terms of any metric (number of measurements, geographical scale, number of cells *etc.*) in this context and provide unique insight into city-wide signal map prediction.

Moreover, for city-wide signal maps we should train a separate `RFs` model per cell, when there is a large number of data points per cell, otherwise we should use cID as a feature. Overall our `RFs`-based predictors offer (i) superior performance, (ii) better performance *vs.* measurements tradeoff and (iii) extensibility to any RSS metric and different features.

In this chapter, we focused on minimizing the mean squared error (MSE) for the prediction task, however, there are *certain* values of signal strength that might matter more than others (*e.g.,* low coverage areas). Moreover, this chapter evaluated the `RFs` predictors on unobserved test data, which might not correspond to the real target data distribution (*e.g.,* we already got a glimpse of the sampling biases in chapter 3). The next chapter addresses both of these issues: we demonstrate (i) how to optimize the coverage map via quality functions and (ii) mitigate the sampling biases via weight functions.

Chapter 5

Quality and Weight Functions

Data Is Useless Without Context

Nate Silver, "The Signal and the Noise"

5.1 Overview

We already discussed the need for coverage (signal strength) map prediction techniques in order to improve the accuracy of these maps based on limited data. These include propagation models [76, 16] as well as data-driven approaches [33, 19, 39] and combinations thereof [59]. Increasingly sophisticated machine learning models have been developed that try to capture various spatial, temporal and other characteristics of signal strength [61, 29] including our work in Chapter 4.

Although there are complicated models for the signal strength itself, as we discussed in Sec. 4.5.2, not all signal strength values matter the same and to the best of our knowledge all prior work focused solely on minimizing the mean square error (MSE) for the signal strength prediction. However, this strategy, of treating all values equally, neither necessarily maps directly to cellular operators' objectives nor reflect the users' experience. First, the operator may be more interested in predicting *quality functions* (such as the number of signal bars and the call drop probability), which depend on but are different from measurable KPIs. For example, the operator may be more interested in predicting accurately good vs. poor coverage than in minimizing the MSE of signal strength. Second, an operator, may be interested in some locations more than others, *e.g.,* locations chosen uniformly at random, locations with dense user population, or specific locations of interest (*e.g.,* to 911 dispatchers or to beat competition), while relying on sampling distributions that are different from target distributions of interest. For example, Fig. 3.7a depicts the signal map for highways leading to JFK, which are over-sampled compared to nearby residential blocks (as shown on Fig. 5.8).

In this chapter, we develop a principled machine learning framework that provides cellular operators (and mobile analytics companies) with knobs to tackle the mismatch between (1) operators' quality functions and raw signal strength as well as (2) sampling and target distributions. Our focus is not on improving the machine learning model itself (although we adopt state-of-the-art random forest-based prediction as our running example), but on the above two orthogonal aspects. These can, in principle, be combined with any prediction model and have not been addressed by existing literature.

More specifically, we make the following contributions.

First, we identify **quality functions** based on signal strength, such as mobile coverage indicators and call drop probability (CDP), which are not directly optimized by learning on signal strength. While prior work minimizes naively the MSE for signal strength [29, 33, 39, 59], we train models directly on these functions and we show that we can improve the relative error up to 32% in the high CDP regime of greatest concern to cellular operators, recall from 76% to 92% for predictions of coverage loss (where false negatives are costly to operators) and balanced accuracy from 87% to 94%. Working directly with the quality function, our methodology optimizes directly the function of interest and allow operators to put more emphasis in the values and use cases of signal maps that matter most. Alternatively, if signal strength (*e.g.,* RSRP) prediction is needed, we can leverage the CDP QoS optimization framework to improve the signal strength prediction *itself* (up to 3dB in RMSE improvement), in its low values regime.

Second, we introduce **weight functions** that can express the importance operators give to particular locations. This reweighting is rooted at the framework of importance sampling and allows us to obtain unbiased error metrics in settings for which the available data is not sampled proportionally to the target distribution of interest (*a.k.a.* dataset shift problem where the train and the test distribution differ significantly [67]). We demonstrate two intuitive weight function classes, respectively encoding (i) uniform loss with respect to spatial area; and (ii) loss proportional to user population density. Training models with reweighted errors shows an average improvement of 5% and up to 20% for oversampled regions.

Combining both techniques shows improvement up to 5.5%, compared to the base problem prediction, for the estimation of CDP adjusted with population and uniform distributions.

Finally, we leverage the two **real-world LTE datasets**, which are presented earlier in Chapter 3 to evaluate the performance of our framework: (i) the small but dense Campus dataset, we collected on our own university campus; and (ii) the large but sparser city-wide (in NYC and LA) datasets, provided by a mobile data analytics company. The city-wide scale of the latter dataset, allows us to gain valuable insights into the sampling strategies of mobile analytics companies and reveal significant sampling biases, that we handle in this chapter.

This chapter in perspective. In the previous Chapter 4, we proposed random forests for predicting signal strength (RSRP) based on a number of features, including but not limited to location and time (see Section 4.4 and 4.4.2). In this chapter, we build on this random forest-based predictor as our running example. However, our focus is no longer on evaluating

	Notations	Definitions-Description
Data	$\mathbf{x}$	Measurement's Features
	$y = \{y^P, y^I, y^C\}$	Label - KPI (Key Performance Indicator)
	y^P	RSRP: Received Signal Reference Power
	y^I	RSRQ: Received Signal Reference Quality (Interference)
	y^C	CQI: Channel Quality Indicator
Network Quality Functions	$Q(y)$	Network Quality Function (e.g $Q_c(y^P)$, $Q_{cdp}(y^P)$
	$Q_c(y^P)$	Mobile Coverage Indicator
	$Q_{cdp}(y)$	Call Drop Probability
Error / Loss Scores	$L(\widehat{y}, y)$	Loss function of its arguments (*e.g.,* squared loss)
	ε_p	Reweighted Error Metric for target distribution $p(\mathbf{x})$
Importance Sampling Framework	$p(\mathbf{x})$	Target distribution
	$s(\mathbf{x})$	Sampling Distribution
	$P(\mathbf{x})$	Population Distribution
	$u(\mathbf{x})$	Unifom Distribution
	$W(\mathbf{x})$	Weighting Function
	w_u	Importance Ratio for Uniform Target Distribution
	w_P	Importance Ratio for Population Target Distribution

Table 5.1: Notation used in this chapter.

the underlying RFs algorithm itself. Instead we can *leverage any* ML-based predictor to allow a cellular provider to express operational objectives, which have not received attention in prior literature, namely (1) quality functions that are not directly optimized by learning the signal strength itself and (2) importance sampling to address the mismatch between the sampling and target distribution (*i.e.,* the dataset shift problem [67]).

Outline. The rest of this chapter is organized as follows. Section 5.1.1 recaps the formulation of the coverage maps prediction for both signal maps values and QoS values. Section 5.2 presents the quality functions framework for coverage indicators and call drop probability. Section 5.3.2 develops the weight error functions and the corresponding importance sampling framework. Section 5.4 presents evaluation results. Section 5.5 concludes the chapter.

Training Options	y **Domain**	Q **domain**
same $w = 1$ for all points	$\widehat{y} \to Q(\widehat{y})$	$Q(y) \to \widehat{Q}(y)$
training weights $\mathbf{w}$ (Table 3)	$\widehat{y_w} \to Q(\widehat{y_w})$	$Q(y) \to \widehat{Q}^w(y)$

Table 5.2: Overview of Prediction Methodologies. One can perform prediction on the y (signal) or on the Q (quality) domain. One can assign the same or different weights to different points. These are orthogonal to each other and to the prediction model used.

5.1.1 Problem Formulation

In Chapter 2 (see Sec. 2.1.3), we formulated coverage maps (for both signal strength/KPIs y and QoS Q) and prediction ($\widehat{y}$ and $\widehat{Q}$). To recap briefly, a coverage map is a collection of N measurements $(\mathbf{x_i}, y_i)$, $i = 1, 2...N$ where the label y_i is a KPI measurement collected along with the features $\mathbf{x}_i$ which specifies the location, time, device, network information *etc.* for the signal value to be mapped. A coverage map in the QoS domain is defined as quality-transformed observations $(\mathbf{x}_i, Q(y_i))$ instead of raw signal strength observations $(\mathbf{x}_i, y_i)$. We elaborated on $(\mathbf{x}_i, y_i)$ in Sec. 2.1.1 and 4.4.2 where we refer for details. In this chapter, we discuss in detail network quality functions $Q(y)$ (*e.g.,* call drop probability or coverage 0-1 indicator) and our goal is to predict directly in the Q domain ($\widehat{Q}(y)$). We also show the relationship between $y \leftrightarrow Q(y)$ and how we can leverage prediction in the Q domain, *i.e.,* $Q^{-1}\left(\widehat{Q}(y)\right)$, in order to improve prediction back in the signal strength y domain for the values that matter the most for the operators

In Section 5.3.2, we present important sampling to allow the operator to specify what points are important to predict, via weight functions W. Prior work focused on sophisticated models to predict $\widehat{y}$, then computed $Q(\widehat{y})$. Our contribution lies in proposing two other aspects of prediction (*i.e.,* network quality function and importance sampling) that provide knobs to allow operators to express their operational objectives and optimize signal map prediction accordingly. The two ideas are orthogonal to each other and to the prediction model used, thus can be used independently or jointly. Tables 5.2 and 5.1 summarize the methodologies and notation, respectively.

5.2 Quality of Service (QoS) Functions

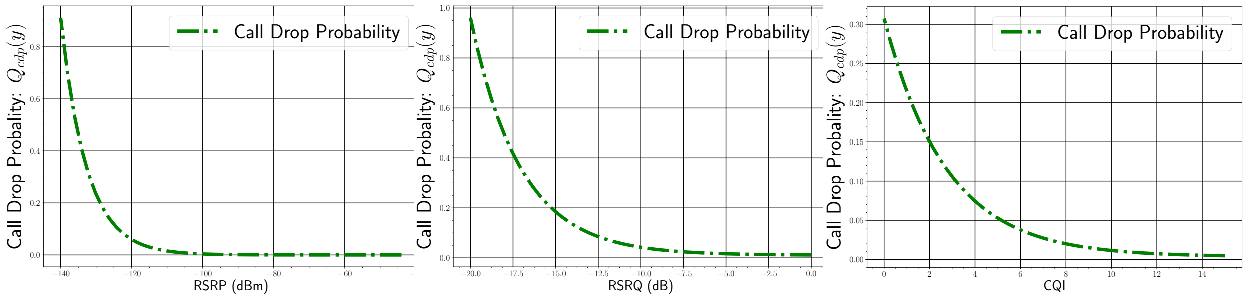
5.2.1 Background

We introduced QoS in Section 2.1.1 and here we present a more detailed taxonomy, considering the wide range of use cases. QoS literature are focused on (i) measuring the QoS itself and (ii) take actions to ensure an adequately level of QoS in the network. First, for QoS measurements, as reviewed in Chapter 2 (Sec. 2.1.2) and seen above, the terms KPIs and QoS are sometimes being used interchangeably. Similarly, LTE RSRP is used as a proxy in [43], to determine the call drop probability. Second, QoS may also refer to any technology that manages data traffic to reduce packet loss, latency *etc.* on the network, *i.e.,* tries to offer adequate network service, particularly the performance as seen by the users. A good example is the QoS Class Identifier (QCI) in LTE [27], for different queue buffers according to the traffic type.

Similarly, we use the term QoS to relate to *user experience*, rather than a sophisticated performance metric (we already reserved the term KPIs for them). Thus, we define a quality of service (QoS) function as follows:

A *QoS function*, Q, is a real-valued function of KPI y that reflects an application-specific outcome that depends upon signal strength (*i.e.,* KPI) y. Such examples of QoS of interest to cellular operators include, but are not limited to, the call drop probability $Q_{cdp}(y)$, the number of signal bars $Q_B(y)$, and the mobile coverage indicator $Q_c(y)$.

There is a large body of work related to QoS, however, to the best of our knowledge, this thesis is the first to consider how to leverage QoS in order to improve a signal strength map itself. Our rationale is simple: we use a cellular operators objective and user experience proxies that might not be reflected to the typical MSE minimization in the signal strength domain. We reviewed such an example in Chapter 4 where the same error for $y = -110$dBm would miss-characterize low coverage area *vs.* for $y = -80$dBm would not really matter. Thus, for



(a) $Q_{cdp}(y^P)$ *vs.* RSRP.

(b) $Q_{cdp}(y^I)$ *vs.* RSRQ.

(c) $Q_{cdp}(y^C)$ *vs.* RSRP.

Figure 5.1: Call Drop Probability (CDP) $Q(y)$ as a function of KPIs.

certain tasks, we can operate to the QoS domain and then transform to signal strength domain. This chapter demonstrates how our approach, which exploits the nonlinear dependence of quality-of-service on raw signal strength, is superior for many practical applications.

5.2.2 Call Drop Probability (CDP)

One of the most important cellular network quality metrics is the call drop probability (CDP), *a.k.a.* connection drop rate (see above). We here model CDP with the exponential function, $Q_{cdp}(y) = ae^{-by} + c$, with parameters a, b, c estimated using empirical data from the literature [43], [42]. Examples of CDP vs. KPIs are shown on Fig. 5.1. It is immediately apparent that nearly all of the variation in CDP occurs at signal strengths below -100 dBm, implying that signal strength errors at high dBm will have far less impact on predictions of Q_{cdp} than errors of equal size at low dBm. As a continuous outcome, the call drop probability estimation $\widehat{Q}_{cdp}(y)$ can be treated as a ML regression problem.

5.2.3 Mobile Coverage Indicator and Signal Bars

Absolute RSRP values y are translated to the widely used network performance bars $Q_B(y)$ on mobiles' screens. Mobile analytics companies usually produce 5-colors map to visualize signal bars [55]. Despite variation across devices, typical LTE RSRP values for iOS and Android devices are:

$$Q_B(y) = \begin{cases} 0 & \text{if } y^P <= -124 \text{ dBm} \\ 1 & \text{if } y^P \in [-123, -115]) \text{ dBm} \\ 2 & \text{if } y^P \in [-114, -105]) \text{ dBm} \\ 3 & \text{if } y^P \in [-104, -85]) \text{ dBm} \\ 4 & \text{otherwise } (\textit{i.e.,} \text{ excellent reception}) \end{cases} \tag{5.1}$$

As with the other Q functions described here, better accuracy for producing such maps can be obtained by seeking to directly reduce error in Q_B, rather than y.

Mobile Coverage Indicator. Detecting areas with weak/no signal (*i.e.,* bad coverage) is a major problem for cellular operators. This is essentially a binary classification (per [34]). We define the mobile coverage indicator as a function of RSRP for LTE [34]:

$$Q_c(y) = \begin{cases} 0 & \text{if } y^P <= -115 \text{ dBm, } \textit{i.e.,} \text{ 0 or 1 bar} \\ 1 & \text{otherwise} \end{cases} \tag{5.2}$$

The rationale behind this indicator is that the call drop probability begins to increase exponentially and the service deteriorates significantly below -115dBm [43]; at this threshold, a mobile phone is on the edge of very bad service. We want to detect areas of bad coverage because undetected $Q_c(y) = 0$ could impact the operator's reputation, revenue and overall performance (*e.g.,* a cell upgrade or a SON/SDN configuration could solve the problem).

For the reader's convenience, we should re-iterate some of our terminology. We use the term mobile coverage maps as prior art does ([61, 49]) to refer to both continuous signal strength (*e.g.,* LTE RSRP) values as well as other forms of such maps such as the 5-colors map for signal bars *etc.* When we need to refer to both we could distinguish them by referring as "signal maps" for the continuous version and "coverage indicator" for the binary version.

5.2.4 Prediction in the Original domain *vs.* the Quality Domain

$\widehat{\mathbf{Q}}(\mathbf{y})$ **vs.** $\mathbf{Q}(\widehat{\mathbf{y}})$: In this thesis, we show that prediction can be improved by training models directly on QoS observations $Q(y)$ and predicting $\widehat{Q}(y)$ instead of using the proxy $Q(\widehat{y})$; in other words, we minimize the error of $f_Q(\mathbf{x})$ instead of minimizing the error of $f_y(\mathbf{x})$. This is equivalent to changing the loss function used in estimation from one that treats errors at all y values equally to one that emphasizes errors with practical consequences for cellular operators such as mis-identifying bad coverage areas or failing to predict areas with high call drop probability (*e.g.,* see Fig. 5.1, $y^P \leq -100$dBm). Our experimental results in Section 5.4 show how the prediction is improved for these regions that matter the most; *e.g.,* we can improve the prediction of bad coverage areas ($Q_c(y) = 0$) for the binary coverage indicator problem and the prediction for below -100dBm for the continuous signal map.

Prediction of $\widehat{Q}$ using Random Forests. We use `RFs` to predict $\widehat{Q}$, similarly to predicting $\widehat{y}$ in the previous chapter. Given a QoS metric $Q(y)$ that is a *deterministic* function of y, we can model $\widehat{Q}$(y) prediction similarly: $Q(y)|\mathbf{x} \sim \mathcal{N}(RFs'_\mu(\mathbf{x}), \sigma^2_\mathbf{x})$, where RFs'_μ is trained on quality-transformed observations $(\mathbf{x}_i, Q(y_i))$ instead of raw signal strength observations $(\mathbf{x}_i, y_i)$. This simple procedural modification (using $\widehat{Q}(y)$ in place of $Q(\hat{y}(\mathbf{x}))$) can improve prediction.

5.3 Weight Functions for Cellular Operators

Cellular Operators Objectives. We have argued above that different values of y are not equally important for operators from a QoS perspective, therefore we invoke the QoS function Q to modify out losses so as to place more weight on errors with greater practical consequence. Similarly, not all inputs $\mathbf{x}$ are necessarily equally important. For example, an operator may be particularly interested in accurate predictions in some locations, *e.g.,* 911 locations or hospitals, areas with high revenue or competitive advantage and areas with dense user population. Therefore, selection of appropriate (i) ML training loss functions and (ii) evaluation error metrics should be aligned with operators' use cases and objectives.

Prior work [19, 29] has primarily minimized MSE for predicted signal strength via cross-validation (CV), *i.e.,* report the error on held-out test data, after training on a sample of signal strength measurements. This implicitly assumes that *all observations are equally important* for both learning and evaluation, and further, that the importance of error minimization in some subset of $\mathbb{X}$ is proportional to the number of observations in it. These are strong assumptions that are often violated in practice. For example, an operator might consider all locations within an areal unit having equal importance. If, however, the available data is distributed according to population (which is highly uneven), then the weighting implicitly used in the analysis will be far from the desired (uniform) distribution. Conversely, an operator interested in population-weighted error may encounter problems when using data intensively collected by a small subset of users with residential locations or commuting patterns that are not reflective of the customer base. Such mismatches between the *sampling distribution* of signal strength observations in $\mathbb{X}$ and the *target distribution* that captures the operator's desired loss function lead to prediction bias, which can be corrected by techniques borrowed from importance sampling. It should be noted that ML literature uses the term "dataset shift" [67] to describe this mismatch between the training and the test data.

5.3.1 Importance Sampling

Importance sampling [69] is a general technique for estimating properties (*e.g.,* the expected value) of a particular distribution, when the available samples are generated from a different distribution than the distribution of interest. We start with a function $\varepsilon(\mathbf{x})$ and suppose that our problem is to calculate the quantity $\mu = \mathbb{E}_p\left[\varepsilon(\mathbf{x})\right] = \int_{\mathcal{D}} \varepsilon(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$, where p is a probability density function on $\mathcal{D} \subseteq \mathbb{R}^d$. We can write the following:

$$\mu = \int_{\mathcal{D}} \varepsilon(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \int_{\mathcal{D}} \frac{\varepsilon(\boldsymbol{x})p(\boldsymbol{x})}{s(\boldsymbol{x})} s(\boldsymbol{x})d\boldsymbol{x} = \mathbb{E}_s\left(\frac{\varepsilon(\mathbf{x})p(\mathbf{x})}{s(\mathbf{x})}\right)$$

where we assume that $s(\cdot)$ is a positive pdf over $\mathbb{R}^d$ and $\mathbb{E}_s(\cdot)$ denotes expectation for $\mathbf{x} \sim s$. Our goal remains to find the original $\mathbb{E}_p\left[\varepsilon(\mathbf{x})\right]$. In essence, by making a multiplicative adjustment to ε we compensate for sampling from s instead of p, with this factor $\frac{p(\boldsymbol{x})}{s(\boldsymbol{x})}$ being known as importance ratio, which expresses the relative weight given to a data point under the target distribution p versus the sampling distribution s (*a.k.a.* importance distribution).

It can be proven [69] that an unbiased estimator of $\mu = \mathbb{E}_p\left[\varepsilon(\mathbf{x})\right]$ is given by the importance sampling estimate:

$$\widehat{\mu}_s = \frac{1}{N}\sum_{i=1}^{N} \frac{\varepsilon(\mathbf{x}_i)p(\mathbf{x}_i)}{s(\mathbf{x}_i)}, \ \mathbf{x}_i \sim s \tag{5.3}$$

In order to use Eq. (5.3) we need to be able to compute $\frac{p(\mathbf{x})}{s(\mathbf{x})}$ at any $\mathbf{x}_i$ we might sample. When p or s has an unknown normalization constant, then we can utilize a ratio estimate.

5.3.2 Reweigthed Errors with Importance Sampling

We are interested in fitting to and assessing performance via an error metric corresponding to an operator-defined *objective*, which is some measure of expected prediction error (i) integrated over some space $\mathbb{X}$ (*e.g.,* geography, time, frequency band), as per 4.4.2 (ii) with some weight function that says how much the operator cares about different points in that space; the space of interest is our feature space $\mathbf{x}$ as described in 4.4.2 and 3.4. The expected prediction error over the target data distribution of interest $p(\mathbf{x})$ can be written as:

$$\varepsilon_p = \varepsilon\left(\mathbf{x}, W, \widehat{y}, y\right) = \int_{\mathbb{X}} W(\mathbf{x})\mathbb{E}\left[L\left(\widehat{f}(\mathbf{x}) - f(\mathbf{x})\right)\right] d\mathbf{x} \tag{5.4}$$

where,

- $W(\mathbf{x}) \rightarrow \mathbb{R}^{+}$ is the weighting function for the cellular operators‘s objective of interest.
- $L(\cdot)$ is the loss function, *e.g.,* the square of its arguments in this thesis.
- $f(\mathbf{x}) = y \rightarrow \mathbb{R}^{y}$ to facilitate our notation (please remember that we already denote the predictor of y with $\equiv \widehat{f}_y(\mathbf{x}) = \widehat{y}$).
- The following condition is true: $\int_{\mathbb{X}} W\left(\mathbf{x}\right) d\mathbf{x} < \infty$.

For simplicity of notation, we write the above integral generically over set $\mathbb{X}$. In practice, this integral will typically be over the various dimensions of the input space (*e.g.,* $\int_{\mathbf{l}} \int_{\mathbf{t}} \cdots d\mathbf{t} d\mathbf{l}$, where $\mathbf{l} = (l^x, l^y)$ denotes the location vector and $\mathbf{t}$ the time vector.

If we knew $\mathbb{E}\left[L((\widehat{f}(\mathbf{x}) - f(\mathbf{x})))\right]$, we could directly evaluate this integral, however we do not. For a given predictor (*e.g.,* `RFs` in our case, but it could be any nonparametric prediction function) and outcome, we do not actually know the expected prediction loss. Ideally, $f(\mathbf{x})$ would be given by an oracle, which knows the “true” underlying signal strength map phenomenon $Y|\mathbf{x}$ (see Sec. 2.1.2 for the signal maps definition).

Nevertheless, we can sample from smartphones measurements $(\mathbf{x}_i, y_i)$ and compare our predictions to true values under *e.g.,* cross-validation (CV). However, the mean CV error itself will not in general give us ε_p, because CV is based on the sampling distribution of the data $s(\mathbf{x})$, which may look nothing like $W(\mathbf{x}), \mathbf{x} \sim p(\mathbf{x})$ (which we can interpret as target distribution, though it may not be normalized). In order to deal with the miss-match of the sampling and the target probabilities, we turn to importance sampling techniques (introduced above in Sec. 5.3.1). If we know that our *sampled* data represents iid draws from the sampling distribution (PDF) $s(\mathbf{x})$, we then have:

$$\begin{aligned}
\varepsilon_p(\mathbf{x}, W, \widehat{y}, y) &= \int_{\mathbb{X}} W(\mathbf{x}) \mathbb{E}\left[L(\widehat{f}(\mathbf{x}) - f(\mathbf{x}))\right] d\mathbf{x},\ \mathbf{x} \sim p(\mathbf{x}) && (5.5)\\
&= \int_{\mathbb{X}} W(\mathbf{x}) \mathbb{E}\left[L(\widehat{f}(\mathbf{x}) - f(\mathbf{x}))\right] \frac{s(\mathbf{x})}{s(\mathbf{x})} d\mathbf{x},\ \mathbf{x} \sim p(\mathbf{x}) && (5.6)\\
&= \mathbb{E}_s\left[\frac{W(\mathbf{x}')}{s(\mathbf{x}')} \mathbb{E}\left[L(\widehat{f}(\mathbf{x}') - f(\mathbf{x}'))\right]\right],\ \mathbf{x}' \sim s(\mathbf{x}), && (5.7)
\end{aligned}$$

where $\mathbf{x}'$ represents the (random) test data originating from $s(\mathbf{x})$. By using the law of total expectation and importance sampling estimation [69] from Eq. (5.3), we can provide an estimate for Eq. (5.4):

$$\widehat{\varepsilon_s}(\mathbf{x}, W, \widehat{y}, y) = \frac{1}{N} \sum_{i=1}^{N} \frac{W(\mathbf{x}_i)}{s(\mathbf{x}_i)} \left(\widehat{f}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2, \mathbf{x}_i \sim s(\mathbf{x}_i) \quad (5.8)$$

where N is the number of observations (sampled data points), $s(\mathbf{x}_i)$ is the sampling distribution for data point i, $p(\mathbf{x}_i)$ is the target distribution for data point i and the adjustment factor $\frac{W(\mathbf{x}_i)}{s(\mathbf{x}_i)}$ is the importance ratio.

Thus, we are able to estimate an error weighted by $W(\mathbf{x})$, $\mathbf{x} \sim p(\mathbf{x})$, with data generated from the distribution $s(\mathbf{x})$. More precisely, by making a multiplicative adjustment to the squared error term $(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2$ we compensate for sampling from the distribution $s(\mathbf{x})$ instead of the actual distribution $p(\mathbf{x})$. We note that the above development treats s as

Target Distribution **(Cellular Operator Objective)**	**Importance Ratio** **(Weights w_i)**
Uniform distribution $u(\mathbf{x})$	$w_u \propto \frac{1}{s(\mathbf{x})}$
Population distribution $P(\mathbf{x})$	$w_P \propto \frac{P(\mathbf{x})}{s(\mathbf{x})}$
Operator's custom target distr. $p(\mathbf{x})$	$w_o \propto \frac{p(\mathbf{x})}{s(\mathbf{x})}$

Table 5.3: Examples of importance sampling for operators.

known, which is not always the case; as we show below, however, kernel density estimation (KDE) can be used to estimate s when it is not known a priori.

This development sheds light on what naive CV does. We can provide an interpretation of naive CV by observing what W would yield the standard CV loss, $\varepsilon_{CV} = \frac{1}{N}\sum_i \left(\widehat{f}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2$. If we observe carefully our importance sampling estimate of ε_p from eq. 5.8, our importance ratio $\frac{W(\mathbf{x})}{s(\mathbf{x})}$ must be equal to 1 in this case, leading to $\varepsilon_{CV} = \int_{\mathbf{x}} s(\mathbf{x})\mathbb{E}\left(L(\widehat{f}(\mathbf{x}) - f(\mathbf{x}))\right) d\mathbf{x}$, with $\mathbf{x}$ as before being the distribution of the observed data on the space of interest. Seen in this way, it is obvious that this ε_{CV} function is biased by the sampling distribution and a naive metric of dubious interest (*i.e.,* we get biased errors unless we have been very lucky about how we picked our data and s matches the p distribution).

5.3.3 Weight Functions Examples

In sharp contrast to the common practice of reporting just the naive ε_{CV}, we provide operators with explicit choice of the target distribution, to express which parts of the space are important for prediction. Some intuitive examples are summarized in Table 5.3 and described below.

(1) ε_u uniform over $\mathbb{X}$. This is equivalent to the expected loss evaluated at a random location in $\mathbb{X}$, and reflects settings where the operator is equally concerned with performance over all portions of the target area. To obtain this objective function, we need $W(\mathbf{x})$ proportional to a constant, *i.e.,* the uniform distribution $u(\mathbf{x}_i)$. This leads to an importance ratio $w_u \sim \frac{1}{s(\mathbf{x})}$: we want to weight each data point inversely by how often its region of the space is sampled, *i.e.,* the inverse of the weights implicitly used by naive estimation.

(2) ε_P **proportional to population density.** An intuitive target for operators is loss averaged over the user population, denoted by $P(\mathbf{x})$ at point $\mathbf{x}$ of the input space. We then want $W(\mathbf{x}) \sim P(\mathbf{x})$, thus importance ratio $w_P \sim \frac{P(\mathbf{x})}{s(\mathbf{x})}$. This means that observations from parts of the user population that are rarely sampled need to be given more weight and those that are oversampled should be given less weight. It should be noted that if our sample is representative of our user population, then the naive error estimator is already an approximation of the target. However, if some groups of users are under or oversampled then the naive estimator may not perform well. Data collection from mobile analytics companies can be biased and does not necessarily match the operators' user population.

The former can be generalized further. We always weight inversely by s and directly by W to calculate the importance ratio w. Different choices of W and L determine respectively *where* we want to weight our performance and *how* we judge different kinds of errors. We can also select $\mathbb{X}$ as another tacit choice: do we only use space? time? frequency? device (*e.g.,* some devices might have been under-sampled)? Operators should take care that their targets of interest are well-represented in the data they purchase or collect, to avoid prediction biases.

Estimating the sampling distribution $s(\mathbf{x})$**.** Our observed signal strength data may have come from a known or unknown sampling design $s(\mathbf{x})$. In the latter case, s must be inferred. In Section 5.4 we estimate $s(\mathbf{x})$ via adaptive bandwidth KDE [47] on the 2D spatial space and therefore the importance ratio is $w_u \propto \frac{u(\mathbf{l})}{s(\mathbf{l})}$. Our experimental results show that the main source of bias is location of devices therefore we assume a uniform sampling over the rest of the feature space $\mathbf{x}\in \mathbb{X}$. Our approach can be extended to arbitrary input (features) space $\mathbf{x} \in \mathbb{X}$.

5.3.4 Weighted Random Forests

The reweighted error metric we introduced above from Eq. (5.8) can be used for both (i) the training procedure as well as (ii) the evaluation and final error reporting. In this thesis, we utilize `RFs` (as described earlier in this chapter and in Chapter 4) for prediction of signal strength, therefore, we present shortly how we can implicitly incorporate the reweighted score in the training of the `RFs` algorithm.

For growing an individual tree, the `RFs` algorithm splits each node utilizing a random set of features. The criterion of each split is to maximize the Information Grain (for classification), or to minimize the MSE (for regression). For N training samples for the signal map of interest, weighted `RFs` [20] adjust MSE for each split [58] according to the samples weight vector $\mathbf{w} = (w_1, \cdots, w_N)$, (*i.e.,* implicitly turning loss function to a $wMSE$) while the default setting would be $w_i = 1$.

5.4 Performance Evaluation

5.4.1 Setup

Random Forest Setup. We train `RFs`, for predicting the KPI $\widehat{y}$ (then calculating $Q(\widehat{y})$) or $\widehat{Q}(y)$ directly. Basically, we utilize the $\texttt{RFs}_{all}$ model [7], which is developed in Chapter 4 as the underlying predictor, but it could be replaced by other ML model. Our focus is *not* on evaluating `RFs` in this chapter, but rather our two proposed improvements on top of `RFs` and on how to leverage any ML model. We refer to Sec. 4.4.2 for a features $\mathbf{x}$ recap.

`RFs` Hyperparameters Selection. The most important hyper-parameters for `RFs` are the number of decision trees (*i.e.,* n_{trees}) and the maximum depth of each tree (*i.e.,* $\max_{depth}$).

We follow the same selection procedure as Sec. 4.5.1. For the Campus dataset, we select $n_{trees} = 20$ and $\max_{depth} = 20$; larger $\max_{depth}$ values could result in overfitting. For the NYC and LA datasets, we select $n_{trees} = 1000$ and $\max_{depth} = 30$; more and deeper trees are required for larger datasets.

RFs Model Granularity. One important design choice is what granularity we choose to build our RFs models. As we extensively demonstrated in Chapter 4, using a model per cell (*i.e.,* train a separate RFs model per cell cID with $\mathbf{x_j^{-cID}} = \{x : x \in \mathbf{x_j^{full}}, \sim x \notin \{cID\}\}$) is beneficial when there is a large number of measurements per cID. On the other hand, in many cases in sparser data such as NYC and LA datasets it is better to train a model per LTE TA using $\mathbf{x_j^{full}}$. In this chapter, we utilize models per cID for the Campus dataset and per LTE TA models for NYC and LA datasets as Chapter 4 and our work in [7], unless otherwise stated.

Default Training RFs *vs.* **Weighted Training RFs$_w$.** We want to improve the reweighted prediction error ε_p according to operators objectives (Section 5.3). Thus, we train weighted random forests RFs$_w$ model as described in 5.3.4, with $w_i = \{w_{u_i}, w_{P_i}\}$ proportionally to the target distribution (see Table 5.3). In essence, the ML training weights are set equal to the importance ratio of each sample. We compare RFs$_w$ with the default RFs, where all samples are weighted equally. Please note that this applies to both predicting signal strength values y and quality $Q(y)$ (See Table 5.2 for a summary).

Splitting Data into Training and Testing. We select randomly 70% of the data as the training set $\mathcal{D}_{train} = \{\mathbf{X}_{train}, \mathbf{y}_{train}\}$ and 30% of the data as the testing set $\mathcal{D}_{test} = \{\mathbf{X}_{test}, \mathbf{y}_{test}\}$ for the problem of predicting missing signal map values (*i.e.,* KPIs $y = \{y^P,\ y^I,\ y^C\}$ or QoS $Q_c(y)$, $Q_{cdp}(\mathrm{y})$). The reported results are averaged over $S = 10$ random splits.

Evaluation Metrics - Coverage Classification. We evaluate the performance of the QoS coverage indicator $Q_c(y)$ in terms of binary classification metrics, *i.e.,* recall, precision, F1 score and balanced accuracy. Recall, as we will see, is very important for the cellular operators.

Recall: For a class of interest is a measure of completeness, *a.k.a.* the ratio of relevant instances $\bigcup$ retrieved instances over the relevant instances (*i.e.,* what's fraction of the relevant instances were actually retrieved). In other words, it is defined as $R = \frac{T_p}{T_p+F_n}$ where T_p is the true positive rate and F_n is the false negative rate, *for the class of interest.*

Precision: It is a measure of exactness or quality, *a.k.a.* the ratio of relevant instances over the retrieved instances. In other words, $Pr = \frac{T_p}{T_p+F_p}$ where F_p is the number of false positives.

F1 Score: (*a.k.a.* F-score) It is an overall accuracy measure and can be interpreted as a weighted average of the precision and recall. The relative contribution of precision and recall to the F1 score are equal. $F1 = 2 \times (Pr \times R)/(Pr + R)$.

Balanced Accuracy: The balanced accuracy in binary classification problems to deal with unbalanced datasets. It is defined as the average recall obtained on each class.

Evaluation of Regression. For signal strength maps[1] with continuous signal strength-KPIs values (*i.e.,* $y = \{y^P, y^I, y^C\}$) we define the following evaluation metrics.

Root Mean Squared Error (RMSE). Similarly to the evaluation Sec. 4.5 in Chapter 4, if $\widehat{y}$ is an estimator for y, then $RMSE(\widehat{y}) = \sqrt{MSE(\widehat{y})} = \sqrt{E((y-\widehat{y})^2)}$, in dB for RSRP y^P(since RSRP is reported in dBm) and RSRQ $y^I$and unitless for CQI y^C.

Reweighted Error ε_p for target distribution $p(\mathbf{x})$. According to importance sampling estimate from eq. 5.8, $\varepsilon_p = \frac{1}{N}\sum_{i=1}^{N} w_i\,(\widehat{y_i} - y_i)^2$, with $w_i = \{w_{u_i}, w_{P_i}\} \propto \{\frac{1}{s(\mathbf{l}_i)}, \frac{P(\mathbf{l}_i)}{s(\mathbf{l}_i)}\}$, as defined in

[1] See Sec. 2.1.2 for more about terminology. Coverage maps with continuous $y \equiv$ Signal maps, and coverage maps with QoS $Q_c(y) \equiv$ Coverage Indicator.

Table 5.3, where w_u corresponding to the importance ratio for error in a random location in $\mathbb{X}$ and w_P the weighting proportional to population density. In tis thesis, we use only location density $s(\mathbf{l})$ to calculate the (i) uniform error ε_u or (ii) ε_P, over the space $\mathbb{X}$, however, our methodology is applicable to an arbitrary space $\mathbb{X}$.

5.4.2 QoS Domain Coverage Maps

Coverage Indicator QoS Domain $Q_c(y)$. This setup is a typical binary classification problem, where class 0 corresponds to bad coverage (*i.e.,* coverage hole) and class 1 corresponds to good coverage. As a baseline, we train the `RFs` regression models with the features we described in Sec. 4.4.2 in order to predict $\widehat{y}$ and compute the proxy $Q(\widehat{y})$. We compare that with our approach, which is to train `RFs` classifiers, with the same features, on quality-transformed observations $(\mathbf{x}_i, Q(y_i))$ and predicting $\widehat{Q}(y)$. Please note that for coverage indicator we employ $y = y^P$ since it is defined on RSRP and `RFs` use the default training $(\forall i, w_i = 1)$.

Ideally from operators' perspective is to maximize the Recall for class-0 R_0 because the higher R_0 means fewer false negatives for the class-0, which can be translated to the statement that our algorithm did not classify a bad coverage $(Q(y) = 0)$ as a good coverage area $(\hat{Q}(y) = 1)$. In this setup, coverage holes (class-0) misclassified as good coverage areas (class-1) would impact reputation, revenue, and overall performance (*e.g.,* the need for a cell upgrade may not be detected).

`Campus dataset`: Fig. 5.2 illustrates the improvement in UCI data set from utilizing our predictor $\widehat{Q}(y)$ instead of the naive proxy $Q_c(\widehat{y})$ for bad coverage spots. For this example, we discover 1939 bad coverage sites that the baseline did not detect (16% of the total 12418 bad coverage points). Moreover, Fig. 5.2c shows how the bad coverage spots which were mis-classified as good coverage spots have been reduced by our predictor $\widehat{Q_c}(y)$, especially

Train Data	161634
Test Data	12418

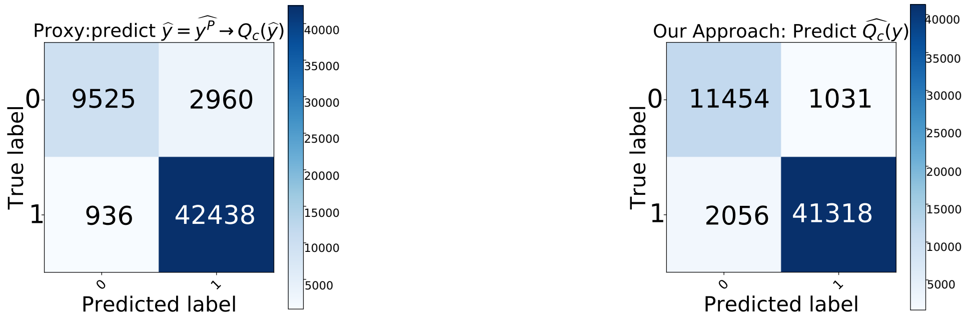
(a) Bad Coverage Spots on UCI Campus.

(b) Campus: Baseline-Proxy- Prediction $Q_c(\widehat{y})$.

Baseline	Our Approach	No.
BAD	BAD	9454
BAD	GOOD	34
GOOD	GOOD	991
GOOD	BAD	1939

(c) Campus: Our Approach $\widehat{Q_c}(y)$.

Figure 5.2: LTE Coverage Map for UC Irvine area (Campus dataset). Display only Test Data. (a) Bad Coverage in Test Data (b) Baseline -Proxy- Prediction $Q_c(\widehat{y})$ (c) Our Model Prediction. It can be seen that (c) has more red points than (b), implying better classification. For this example, we find 1939 data points which the baseline would not detect (16% of the total 12418 bad coverage points). Best viewed in color.



(a) Baseline-Proxy $Q_c(\widehat{y})$

(b) Our method $\widehat{Q_c}(y)$.

Figure 5.3: `Campus dataset`: Confusion matrix for coverage, $Q_c(y)$. The points incorrectly classified as "good coverage" by the baseline $Q_c(\widehat{y})$ are shifted to the "bad coverage" class under our model $\widehat{Q_c}(y)$.

	Recall		Precision		F-1		Accuracy	**Balanced**
$Q_c(\widehat{y})$ Class Label	**0**	1	0	1	0	1		**Accuracy**
$Q_c(\widehat{y})$	0.762	0.978	0.910	0.934	0.830	0.956	0.930	0.870
$\widehat{Q_c}(y)$	**0.917**	0.952	0.847	0.975	**0.881**	**0.963**	**0.944**	**0.935**

Table 5.4: Campus Dataset Coverage $Q_c(y)$ results: (i) Recall for Class-0 (No-Coverage) 76% → 92%, (ii) Accuracy and (iii) Balanced Accuracy Improve. The improved Recall (R_0) is of immense importance for Cellular Providers; higher R_0 means less false negatives for $Q_c(y)$ (*i.e.,* miss-classifications of bad coverage to good coverage).

in areas of densely sampled data and commute traces (note the road and path trajectories). The confusion matrix for these results is shown in Fig. 5.3, where we can see again the shift of points incorrectly classified as "good coverage" by the baseline $Q_c(\widehat{y})$ predictor to the "bad coverage" class under the improved predictor.

The overall classification results, summarized in the terms of the binary classification metrics, are shown in Table 5.4. We see an improvement of 16% for Recall R_0, per Fig. 5.2, as well an improvement in balanced accuracy from 87% to 94%. These improvements do not come at the expense of F1 for class-1 and overall Accuracy, which improve by approx. 1% while F1-score for class-0 improved by 5%.

`NYC and LA datasets`: Table 5.5 lists the classification results for some characteristic examples of `NYC and LA datasets`. We observe a similar increase up to 12% in terms of R_0

	Recall		**Precision**		**F-1**		**Accuracy**	**Balanced**
$Q_c(\widehat{y})$ **Class Label**	**0**	**1**	**0**	**1**	**0**	**1**		**Accuracy**
MNC-1, LTE-TA: x561, NYC Manhattan Midtown								
$Q_c(\widehat{y})$	0.49	0.96	0.71	0.91	0.58	0.94	0.90	0.73
$\widehat{Q_c}(y)$	**0.61**	0.94	0.66	**0.93**	**0.63**	**0.94**	**0.90**	**0.78**
MNC-1, LTE-TA: x552, Eastern Brooklyn								
$Q_c(\widehat{y})$	0.55	0.98	0.80	0.93	0.65	0.96	0.93	0.77
$\widehat{Q_c}(y)$	**0.67**	0.95	0.70	**0.95**	**0.68**	**0.96**	**0.93**	**0.81**
MNC-1, LTE-TA: x641, LA, Covina - Hacienda Heights								
$Q_c(\widehat{y})$	0.58	0.90	0.73	0.82	0.65	0.86	0.80	0.74
$\widehat{Q_c}(y)$	**0.70**	0.86	0.70	**0.86**	**0.70**	**0.86**	**0.81**	**0.78**

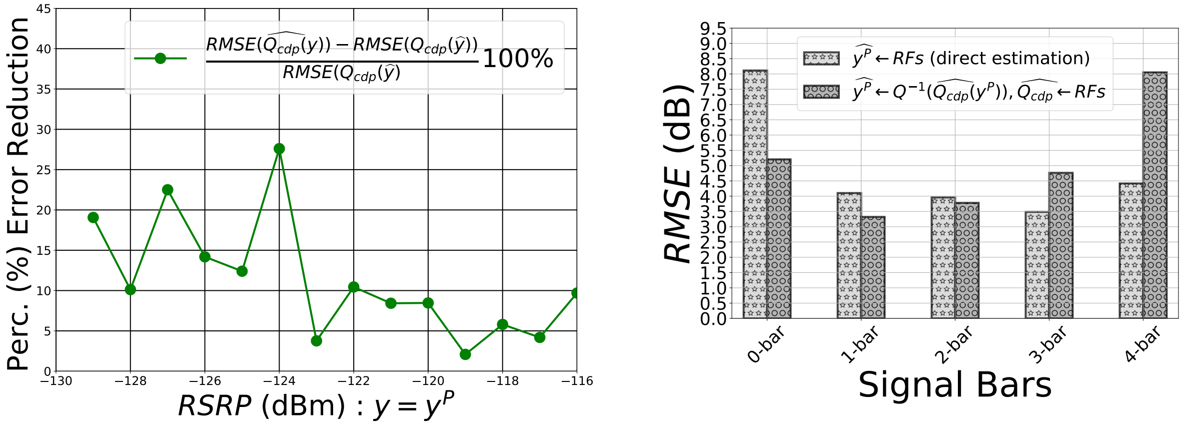
Table 5.5: `NYC and LA datasets` Coverage $Q_c(y)$ results. Recall R_0 improves up to 12%. Operators would ideally minimize the false negatives of class-0. Similar results in other LTE TAs omitted due to space limitations.

for our predictor $\widehat{Q_c}(y)$ compared to the baseline proxy.

In summary, using $\widehat{Q}_c(y)$ instead of $Q_c(\widehat{y})$ optimizes the function of interest instead of naively optimizing the MSE for the regression prediction for the entire range of y. The result is better performance for the outcomes of primary interest.

Call Drop Probability (CDP) Domain $Q_{cdp}(y)$. CDP estimation is a continuous value prediction problem (*i.e.,* regression) on the [0,1] interval. As with the coverage domain, we train `RFs` models with the features described in Section 4.4.2 in order to predict $\widehat{y}$ and use the proxy $Q_{cdp}(\widehat{y})$ as a baseline. We compare that with our approach, which is to train `RFs`, using the same features, on quality-transformed observations $(\mathbf{x}_i, Q(y_i))$ and predict $\widehat{Q_{cdp}}(y)$. We evaluate performance in terms of relative reduction in $RMSE$.

`Campus dataset`*:* We report results for estimating CDP, both by using the proxy baseline as well as predicting CDP directly. Fig. 5.4 shows the relative reduction in $RMSE$ of CDP estimation *vs.* RSRP, which confirms the validity of our design option. Our technique $\widehat{Q}_{cdp}(y)$ improves estimation up to 27% in terms of relative error, in lower reception regime (*i.e.,* bars 0-1, $y^P \leq -115$dBm), where the error function being minimized is highly sensitive to predictive performance in that regime.

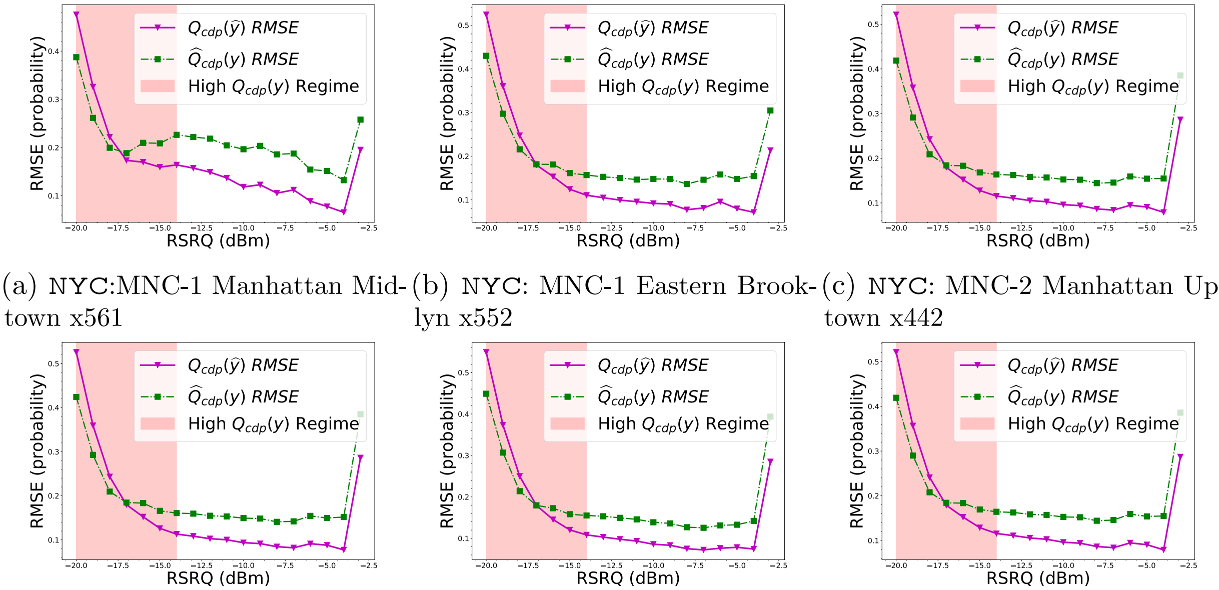


(a) $Q_{cdp}(y)$ $RMSE$ Relative Reduction.

(b) $Q_{cdp}(y) \leftrightarrow$ RSRP y

Figure 5.4: Call Drop Probability $Q_{cdp}(y)$ results for Campus dataset. **(a)** $Q_{cdp}(y)$ $RMSE$ *vs.* RSRP. Our methodology provides an improvement of 27% error reduction for the CDP estimation for the lower RSRP values (bar 0-1). **(b)** $Q^{-1}(\widehat{Q}_{cdp})$ $RMSE$ *vs.* Predicting Directly our RSRP values. We observe that the improvement has been shifted towards the lower RSRP accordingly to the new QoS function $Q_{cdp}(y)$ that was used for training.

From $Q_{cdp}(y)$ to the RSRP $y^P = y$ domain and vice versa. It is very important to highlight that even if our methodology minimizes the QoS error and predicts value in the QoS domain, it is a very elegant way to handle the input in order to improve significantly *signal strength prediction itself* for values that *matter the most* for cellular operators. Fig. 5.4b demonstrates such an example of how we can improve the continuous RSRP y^Pcoverage map (*a.k.a.* signal map) *itself* for the Campus dataset. In order to demonstrate how our technique focuses the sensitivity of the prediction model on the RSRP range with higher call drops, we compare the prediction error of $\widehat{y}$ values (RSRP) themselves, *vs.* inverting $\widehat{Q}_{cdp}(y)$ back to the original y^P space. We group the error by signal bars and we observe that, as designed, the change in learning objective shifts model effort to reducing error where it is most critical (lower signal strength range). We basically exploit the fact that we can tolerate higher uncertainty at high RSRP (where a large error has little impact on CDP). We can hence view our procedure as allowing us to train on an *application-specific loss function*, without requiring us to modify our underlying learning algorithm.



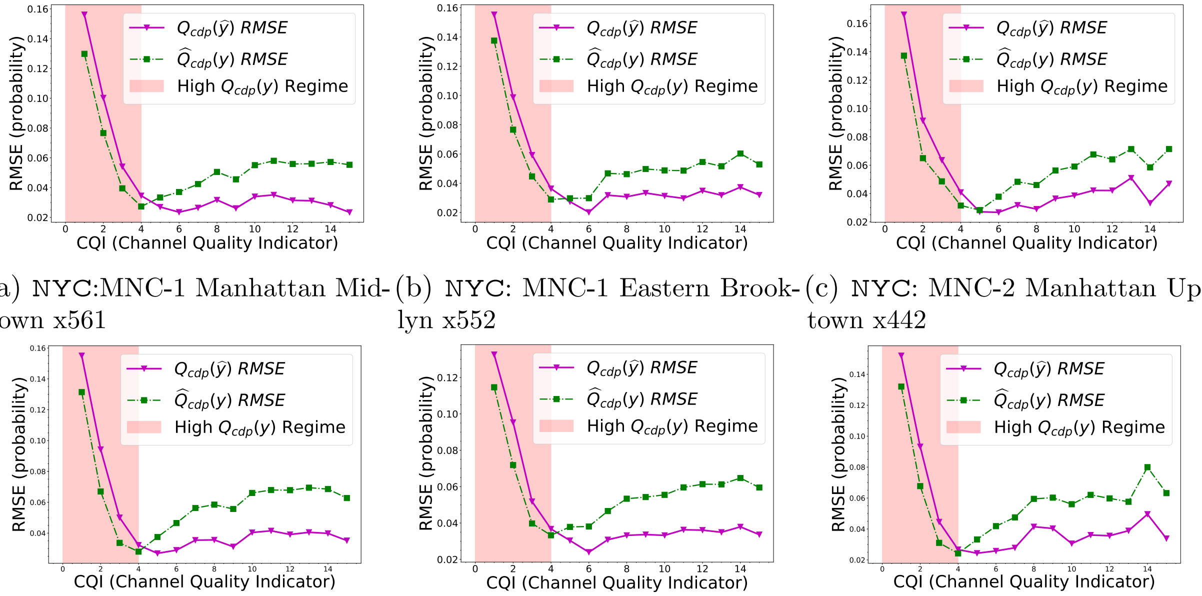
(a) NYC:MNC-1 Manhattan Midtown x561 (b) NYC: MNC-1 Eastern Brooklyn x552 (c) NYC: MNC-2 Manhattan Uptown x442

(d) LA: MNC-1 Suburb Southern LA x211 (e) LA: Mnc-1 LA Downtown - Hollywood x470 (f) LA: MNC-2 Covina Hacienda Heights x641

Figure 5.5: Call Drop Probability $Q_{cdp}(y)$ Estimation for NYC and LA datasets. Results for RSRQ ($y = y^I$), models per cID. Please note that our methodology outperforms the baseline-proxy $Q_{cdp}(\widehat{y})$ in the high CDP regime, where it really matter for the cellular operators.

CDP Estimation for the NYC and LA datasets. We also present results for CDP estimation for NYC and LA datasets; we demonstrate CDP estimation from RSRQ, y^I, and CQI, y^C, data apart from RSRP y^P. Fig. 5.5 and Fig. 5.6 show $RMSE$ of CDP estimation with RSRQ, y^I, and CQI, y^C, respectively. We note that the different KPIs and the use of per-cID models in one case (RSRQ y^I) do not change the improvements from our technique. We improve in the low KPI y regime up to 0.1 in absolute error value (in the probability domain); in terms of relative error our method $\widehat{Q}_{cdp}(y)$ performs up to 32% better than the baseline $Q_{cdp}(\widehat{y})$ for CDP estimation.

Fig. 5.7 summarizes the performance of CDP estimation for the different LTE TAs (*i.e.,* areas) available in our dataset. We plot the log-ratio of the $RMSE$ of the baseline *vs.* our approach. Values greater than 1 indicate improvement for our model as in the other examples, we see that our procedure successfully focuses improvement where it is needed for CDP



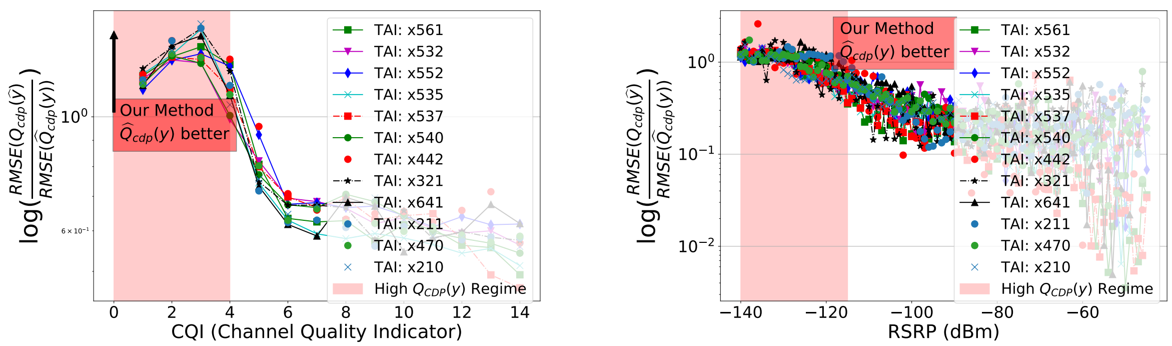
(a) NYC:MNC-1 Manhattan Midtown x561 (b) NYC: MNC-1 Eastern Brooklyn x552 (c) NYC: MNC-2 Manhattan Uptown x442

(d) LA: MNC-1 Suburb Southern LA x211 (e) LA: Mnc-1 LA Downtown - Hollywood x470 (f) LA: MNC-2 Covina Hacienda Heights x641

Figure 5.6: Call Drop Probability $Q_{cdp}(y)$ estimation for NYC and LA datasets. Results for CQI ($y = y^C$), models per LTE TA. Similarly, we offer improvement in the high CDP / low CQI regime.

prediction, rather than wasting statistical power on the high signal strength regime. Similarly to the Campus dataset, we could utilize this improvement in the QoS domain by inverting the CDP estimation $Q^{-1}\left(\widehat{Q}_{cdp}(y)\right)$ back to the original KPI domain, improving the signal map itself.

Why minimizing MSE can be naive. In signal strength prediction, an error of few dB (*e.g.,* 5 dB) will not reflect much change in QoS when the user's received signal strength is high (*e.g.,* -50 to -60 dBm, see Fig. 5.1). The UE experiences excellent QoS in that regime, and hence even moderately large errors in predicted RSRP would not greatly impact predictions of QoS. By turns, an error of 5dB would substantially affect QoS prediction in the weak reception regime (*e.g.,* for -120dBm *vs.*-125dBm you can notice the large difference in CDP in Fig. 5.1). For a signal map which reflects both user QoS experience and operators objectives, it can hence be worth "trading" greater RSRP error in the high-strength regime



(a) CDP with CQI $y = y^C$.

(b) CDP with RSRP $y = y^I$

Figure 5.7: NYC and LA datasets Call Drop Probability $Q_{cdp}(y)$ estimation. $\log \frac{RMSE(Q_{cdp}(\widehat{y}))}{RMSE(\widehat{Q}_{cdp}(y))})$

for lower error in the low-strength regime, as we demonstrated. Working directly with $Q(y)$ alters our application loss function so as to focus performance where it is most needed (but without requiring us to modify the **RFs** procedure to change its nominal loss function). The result is improved performance for QoS outcomes, here up to 32% for the values that matter more to cellular operators.

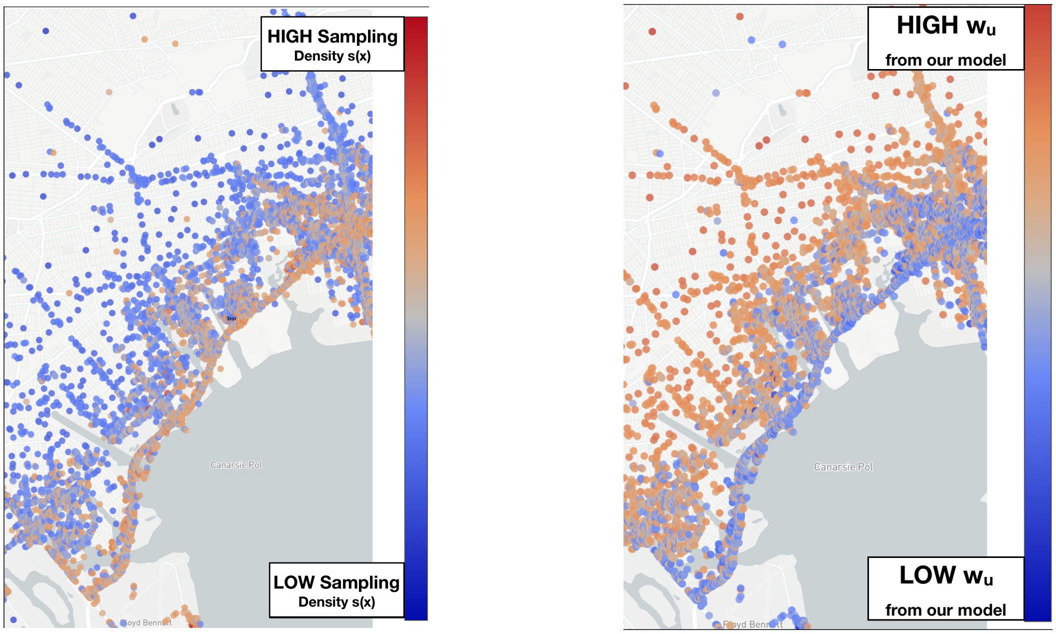
5.4.3 Reweighted Error for Coverage RSRP Maps

In this section, we evaluate our framework in terms of the reweighted error ε_p; we start with the prediction of RSRP signal strength values $\widehat{y}$; we compare a default setting **RFs** *vs.* **RFs**$_w$ (*i.e.*, the ML w_i are set to importance ratio as described in 5.4.1).

ε_u **over Uniform Spatial Distribution.** Campus dataset: In order to calculate the importance ratio $w_u = 1/s(\mathbf{l})$ we estimate $s(\mathbf{l})$ with adaptive bandwidth KDE over the spatial dimensions as we describe in 5.3.3 (an exception is for two cells with very densely sampled data (see Table 5.6) where we used fixed kernel bandwidth estimation over both space and time dimensions).

Cell Characteristics		**Default** $RFs \to \widehat{y}$	$RFs_{w_u} \to \widehat{y}_w$	**Improvement**	
cID	N	$\sqrt{\varepsilon_u}$	$\sqrt{\varepsilon_u}$	**Diff.**	**Diff. (%)**
x922	3955	0.86	**0.69**	0.17	**19.6**
x808	12153	1.54	**1.25**	0.28	**18.5**
x470	7688	0.71	**0.59**	0.12	**17.0**
x460	4069	1.66	**1.44**	0.22	**13.1**
x355	29608	1.77	**1.57**	0.20	**11.5**
x306	4027	2.21	**2.03**	0.18	**8.1**
x901	16049	0.94	**0.91**	0.03	**3.4**
x902*	34164	1.93	**1.90**	0.03	**1.5**
x914	3041	1.66	**1.64**	0.02	**1.0**
x915	4725	1.81	**1.80**	0.00	**0.2**
x312	9727	0.64	**0.65**	-0.01	**-0.6**
x204*	55413	0.91	**0.94**	-0.03	**-3.2**
x034	1554	2.43	**2.68**	-0.24	**-10.0**
All	**186173**	**1.34**	**1.28**	**0.06**	**4.89**

Table 5.6: Campus dataset RSRP y prediction: ε_u Error (*i.e.,* reweighted according to the uniform distribution): (i) Train on Default RFs *vs.* (ii) train on $\mathtt{RFs}_w$ $w_i = w_u \propto \frac{1}{s(\mathbf{l})}$. Models per *cID*. For each *cID* and training case, we pick the best performing adaptive bandwidth KDE for estimating $s(\mathbf{l})$. Our methodology shows improvement up to approx. 20%. For cells * with extremely high sampling density in few locations, we utilize fixed bandwidth estimation both in space and time (see Table 4.4 for the density of the data).



(a) East NYC: Sampling $s(\mathbf{l})$.

(b) Importance ratio $\log(w_u)$.

Figure 5.8: NYC dataset Comparison of the actual data sampling and the correction our model (w_u) to optimize uniform error ε_u. (a) Real sampling estimated by adaptive bandwidth KDE. (b) w_u from importance sampling. It can be clearly seen that the collected data from the Mobile analytics companies oversample devices during commute (GPS Apps push locations updates - power plugged) and undersample other residential areas. We also observed this common engineering practice when we designed our own crowdsourcing system in Chapter 3. Best viewed in color.

Table 5.6 reports the error ε_u for both the default RFs predictor as well as the $\texttt{RFs}_w$. We observe an improvement of up to 20% for ε_u for cells which have been oversampled in just few locations; the average relative improvement is approx. 5%, which demonstrates the significant benefits for readjusting the training loss when the error we want to optimize is different than the typical MSE.

5.4.4 Driving with GPS enabled nearby JFK Airport

The NYC dataset allows us to demonstrate in large scale the mismatch between the sampling distribution and the target distribution (*a.k.a.* dataset shift problem [67]) and how our methodology has real world implications for mobile analytics companies (as we suspected in the preliminary discussion in Chapter 3 about the datasets and our crowdsourcing system). Fig. 5.8a depicts the sampling distribution $s(\mathbf{l})$ in spatial dimensions (estimated by adaptive bandwidth KDE as we described in 5.3.2-5.3.3), in East `NYC` nearby JFK airport. It can be observed that the data are primarily being collected on the highway (Belt Pkwy) adjacent to the sea; the sampling density is much higher compared to nearby residential blocks. Although the specifics of the data collection for `NYC dataset` are proprietary, we hypothesize that the data collection is more frequent when the devices are plugged to power and the users utilize a location navigator app which pushes location updates to other applications. This is a good common practice in crowdsourcing systems [6], as we extensively describe in Chapter 3, to minimize the impact on users' devices. Fig. 5.8b illustrates the importance ratio weights w_u and how our model readjusts for the sampling-target distribution mismatch. Similar patterns in data collection are observed throughout many different areas in `NYC and LA datasets` (*e.g.,* 405 highway in Long Beach area x210). In other words, this mismatch of the collected data with the target distribution is not a bug, but rather is a feature of good crowdsourcing systems, an observation which further motivates our research for dealing with this mismatch.

Table 5.7 reports the error ε_u for different LTE TAs in `NYC and LA datasets`. The average performance improvement by training $\texttt{RFs}_{w_u}$ is approx. 3%, with up to 5% in some areas. We also examined the area x532 where the benefit of our method was small and as expected the spatial distribution was indeed approx. uniform. At the other extreme, regions with highly biased data collection (*i.e.,* x540 East `NYC` near JFK and x210 Long Beach in `LA`) show the highest error reduction (here 3.6% and 5.3% respectively).

Overall, we achieve higher gain from reweighting on `Campus dataset`, as it is collected from a small number of users and hence more unevenly sampled. In general, we expect that this feature will be common for small-scale data sets as well as setups with biased sampling because of mobile analytics companies practices, making reweighting especially important to correct for sampling bias.

LTE-TA Characteristics			**Default** $RFs \rightarrow \widehat{y}$	$RFs_{w_u} \rightarrow \widehat{y}_w$	**Improvement**	
TAI	N	**Area/Neighborhood**	$\sqrt{\varepsilon_u}$	$\sqrt{\varepsilon_u}$	**Diff.**	**Diff. (%)**
x210	197521	Long Beach Lakewood	4.06	3.84	0.21	**5.3**
x552	97942	Eastern Brooklyn	5.38	5.12	0.26	**4.9**
x540	136105	E. Long Island	5.01	4.83	0.18	**3.6**
x535	121159	W. Queens	5.36	5.17	0.19	**3.6**
x641	10663	Covina Hacienda Heights	1.8	1.74	0.06	**3.5**
x561	62448	Manhattan Midtown	5.64	5.46	0.18	**3.2**
x470	198252	LA Downtown Hollywood	4.56	4.43	0.13	**2.8**
x211	77049	Suburban S. LA	4.06	3.96	0.1	**2.4**
x442	14538	Manhattan Uptown Queens - Bronx	3.23	3.19	0.05	**1.5**
x537	37247	Manhattan Midtown East	7.62	7.53	0.09	**1.1**
x321	5111	Eastern Brooklyn	3.87	3.83	0.04	**1.1**
x532	136508	Brooklyn	5.46	5.43	0.03	**0.5**
ALL	**1094543**	**NYC & LA**	**4.88**	**4.72**	**0.16**	**3.16**

Table 5.7: `NYC and LA datasets` RSRP y prediction: ε_u Error (*i.e.,* reweighted according to the uniform distribution): (i) Train on Default `RFs` *vs.* (ii) train on $\texttt{RFs}_w$ $w_i = w_u \propto \frac{1}{s(\mathbf{l})}$. Models per LTE TA. We use adaptive bandwidth KDE for estimating $s(\mathbf{l})$[47]. Our methodology shows improvement up to 5.3%.

Reweighted error to target Population Density ε_P. An alternative to uniform weighting is to weight errors by local population density, resulting in a metric that places more emphasis on accuracy in regions where more potential users reside. To that end, we utilize public APIs to retrieve the census data and estimate the population density $P(\mathbf{l}_i)$. Table 5.8 includes the reweighted ε_P for RSRP data by using the default `RFs` *vs.* the weighted train $\texttt{RFs}_{w_P}$; we see performance improvement up to 5.7%. Please note that cellular operators could also utilize similar users' location activity data (*e.g.,* [22]) from other sources. For example, some locations in census such as big parks or airports, do not have high assigned population and they experience high user traffic.

Reweighted Error for QoS functions. So far, we have separately evaluated the improvement from (1) predicting QoS directly and (2) re-weighting by importance ratio. We can also combine our two contributions and calculate the reweighted error ε_p (how we handle the

LTE-TA Characteristics			**Default** $RFs \to \widehat{y}$	$RFs_{w_P} \to \widehat{y}_w$	**Improvement**	
TAI	N	**Area/Neighborhood**	$\sqrt{\varepsilon_P}$	$\sqrt{\varepsilon_P}$	**Diff.**	**Diff. (%)**
x561	63303	Manhattan Midtown	7.23	6.82	0.41	**5.7**
x321	7014	Eastern Brooklyn	4.94	4.8	0.14	**2.8**
x535	122071	W. Queens	6.03	5.87	0.15	**2.5**
x552	98240	Eastern Brooklyn	5.35	5.29	0.06	**1.2**
x532	137962	Brooklyn	6.24	6.22	0.02	**0.3**
x537	37964	Manhattan Midtown-East	8.82	8.81	0.01	**0.1**
x540	138495	E. Long Island	5.09	5.09	0.00	**0.0**
x442	16372	Manhattan Uptown Queens - Bronx	3.97	4.21	-0.24	**-6.1**
ALL	**621421**	**NYC**	**5.98**	**5.90**	**0.08**	**1.35**

Table 5.8: NYC and LA datasets RSRP y prediction: ε_P Error (*i.e.*, reweighted according to the population distribution): (i) Train on Default RFs *vs.* (ii) train on $\texttt{RFs}_w$ $w_i = w_P \propto \frac{P(\mathbf{l})}{s(\mathbf{l})}$. Models per LTE TA. We use adaptive bandwidth KDE for estimating $s(\mathbf{l})$[47]. Our methodology shows improvement up to 5.7%.

input space) for a QoS function (how we handle the output space) of interest. We only show results for Q_{cdp}(y), although this can be extended to $Q_c(y)$. As summarized in Table 5.2, there are four cases to be compared. First, $Q_{cdp}(\widehat{y})$ is the baseline, where we first predict the signal map value y of interest and then get an estimate of the CDP. Second, $\widehat{Q}_{cdp}(y)$ is our prediction directly on the function of interest. Third, we can train a weighted $\texttt{RFs}_w$ for y, and get $Q_{cdp}(\widehat{y}_w)$. Last, we can have $\widehat{Q}^w_{cdp}(y)$ which is the weighted trained model $\texttt{RFs}_w$ for estimating CDP. In essence, reweighting the samples according to the target distribution is orthogonal to adjusting the outcome of interest $Q(y)$ of interest, and the two approaches can be combined.

Table 5.9 reports the errors for uniform loss over a spatial area, and shows improvements up to 5.5%. Interestingly, the baseline performance deteriorates when we train on the adjusted weights. It tries to minimize MSE for y, therefore the weights can either have very little or even negative effect for mapping back to CDP space. Similar results are observed for error proportional to user population density (see Table 5.10). This again demonstrates the importance of choosing the loss function, here jointly controlled by w and Q, to optimize performance for a specific prediction problem.

	Training Options	y **domain** → $Q(\widehat{y})$		$Q(y)$ **domain**	
KPI: CQI **All LTE-TA regions**	$w_i = 1, \forall i$	$Q_{cdp}(\widehat{y})$	0.018	$\widehat{Q}_{cdp}(y)$	0.0169
	$w_i = w_u \propto \frac{1}{s(\mathbf{l}_i)}$	$Q_{cdp}(\widehat{y_w})$	0,018	$\widehat{Q}^{w}_{cdp}(y)$	0.0160
	Relative Difference		0.5%		5.5%
KPI: RSRP **All LTE-TA regions**	$w_i = 1, \forall i$	$Q_{cdp}(\widehat{y})$	0.028	$\widehat{Q}_{cdp}(y)$	0.023
	$w_i = w_u \propto \frac{1}{s(\mathbf{l}_i)}$	$Q^{w}_{cdp}(\widehat{y})$	0.029	$\widehat{Q}^{w}_{cdp}(y)$	0.022
	Relative Difference		-2%		2.3%

Table 5.9: NYC and LA datasets ε_u Error (*i.e.*, reweighted according to the uniform distribution) results on the Q domain. Predicting $\widehat{y}$ with weights and then converting to $Q(y)$ does not help because information is lost from the transformation. Predicting $\widehat{Q}(y)$ after training with the importance sampling weights can further improve the error up to 5%.

	Training Options	y **domain** → $Q(\widehat{y})$		$Q(y)$ **domain**	
KPI: CQI **All LTE-TA regions**	$w_i = 1, \forall i$	$Q_{cdp}(\widehat{y})$	0.0107	$\widehat{Q}_{cdp}(y)$	0.0088
	$w_i = w_P \propto \frac{P(\mathbf{l}_i)}{s(\mathbf{l}_i)}$	$Q_{cdp}(\widehat{y}_w)$	0,0109	$\widehat{Q}^{w}_{cdp}(y)$	0.0085
	Relative Difference		-2.3%		3.07%
KPI: RSRP **All LTE-TA regions**	$w_i = 1, \forall i$	$Q_{cdp}(\widehat{y})$	0.0045	$\widehat{Q}_{cdp}(y)$	0.0036
	$w_i = w_P \propto \frac{P(\mathbf{l}_i)}{s(\mathbf{l}_i)}$	$Q_{cdp}(\widehat{y}_w)$	0.0047	$\widehat{Q}^{w}_{cdp}(y)$	0.0034
	Relative Difference		-5%		4%

Table 5.10: NYC and LA datasets ε_P Error (*i.e.*, reweighted according to the population distribution) results on the Q domain. Predicting $\widehat{y}$ with weights and then converting to $Q(y)$ does not help because information is lost from the transformation. Predicting $\widehat{Q}(y)$ after training with the importance sampling weights can further improve the error up to 5%.

5.4.5 Applicability to 5G and beyond

Obtaining signal (or other KPI, and most importantly coverage) maps in an accurate and cost-efficient way is a fundamental need in 5G and our method is directly applicable and useful in that context. A major trend in 5G is to use a large numbers of small cells. Having accurate estimates of signal strength, coverage and other KPIs, will be necessary for knowing where to deploy more cells, and how to control 5G parameters. Our framework can naturally handle prediction over small cells, *e.g.,* similarly to what we did with the NYC and LA datasets, where prediction was not per cell, but across an area covered by multiple cells, (*i.e.,* LTE TA) with cID used as a feature. Furthermore, small cells will introduce even higher sampling biases and our importance sampling framework handles an mitigates these train-target distribution mismatches. Moreover, our error metrics, are (i) integrated over $\mathbb{X}$ (*e.g.,* geography, time, frequency band, device) with (ii) weight functions $W(\mathbf{x})$ which can express complex operator objectives in various 5G setups (*e.g.,* IoT, self driving cars).

5.5 Summary

We presented a principled ML framework for cellular coverage map prediction. Instead of evaluating prediction of signal strength itself w.r.t. conventional MSE, we introduced QoS functions (*e.g.,* call drop probability, signal bars, coverage) and importance ratio re-weighting (*e.g.,* for uniform, population, or arbitrary target distributions) that allows a cellular operator to express its operational objectives and optimize prediction (for both classification and regression tasks) accordingly. We demonstrated improvements for both quality and weight functions on the same real-world and large scale LTE data sets we have used in this thesis.

Coverage indicator (or signal bars) QoS maps can be used directly by both mobile analytics and cellular operators for coverage maps and relevant applications, since a class of quality can provide enough information for certain tasks of interest (*e.g.,* visualizations, network decisions in SDN/SON, detect coverage holes *etc.*). We trained models directly on these QoS functions and showed an improvement up to 32% in terms of the relative error in the high CDP regime, which is of greatest concern to cellular operators, as well as improved recall from 76% to 92% for predictions of coverage loss (where false negatives are costly to operators). However, if signal strength (*e.g.,* RSRP) prediction is needed, our CDP QoS optimization framework provides an elegant way to improve the signal strength prediction *itself* (up to 3dB in RMSE improvement), in its low values regime, where it matters more for the cellular operators.

Our importance ratio re-weighting framework, apart from expressing the operational objective of interest, handles and mitigates the dataset shift problem [67], *i.e.,* the mismatch of the available training data distribution with the target (test) distribution. The dataset shift is a prominent problem in the ML area and we hope our work can offer a framework to mitigate it specifically for mobile coverage maps. As we demonstrated in this chapter the best practices of mobile analytics companies could introduce significant sampling biases and our technique allows an improvement of up to 20% for the uniform spatial error when we train models with reweighted loss.

We also showed how both adjustments operate together by implicitly changing the loss function optimized by the ML method, providing a direct and easily implemented way to work with complex, operator-specific loss functions without modifying the underlying learning algorithm. Thus, our methodology is also applicable to the upcoming complex 5G deployments(*e.g.,* dense small cells, IoT, self driving cars), where additionally coverage and other KPIs estimates would be of immense of interest.

Chapter 6

Data Shapley Valuation for Coverage Maps Prediction

If Your Data Is Bad,
Your Machine Learning Tools Are Useless

Thomas C. Redman

6.1 Overview

In this chapter, we apply, for the first time, the problem of data Shapley valuation for mobile coverage maps. Although there have been significant ML developments in the last years, only recent literature addressed valuation of training data points in the context of medical tasks classification [35], but not in the context of mobile data. In this chapter, we study the unique aspects of coverage maps prediction and we address the absence of data valuations tools. Assessing the data Shapley values of training data points enables a number applications: (1) it enables "cleaning" of the data and improving prediction by removing negative values, (2) data minimization and privacy-utility tradeoffs by removing low valued data and (3) it can be an important tool for pricing for mobile crowdsourced data.

We define jointly a specific prediction task and the performance-error metric of interest under the umbrella of data Shapley in order to quantify the data valuation. This holistic approach is necessary since there is no universal value for data points, but the value depends on the particular use of the data in ML. We demonstrate data valuation for various operators metrics instead of the standard accuracy and MSE in classification and regression respectively. We also show how our reweighted errors, from Chapter 5, fit naturally the data Shapley framework. We built on and extend the framework provided by [35], which itself builds on the fundamentals of Shapley value from economics. More specifically, we make the following contributions:

1. We study and implement a wide range of different performance metrics instead of the standard accuracy and MSE in classification and regression respectively. We calculate the data Shapley valuation for the mobile coverage QoS $\widehat{Q_c}(y)$ classification for evaluation metrics such as recall for coverage loss, R_0, which is of immense importance for cellular operators. We analyze the distribution of data Shapley values in our datasets and we apply it to remove data points with negative/low Shapley values, which simultaneously improves prediction and achieves data minimization. For instance, we are able to remove up to 65% of the low valued training data points and simultaneously improve the recall of coverage loss from 64% to 99%. Furthermore, we identify how the dataset shift problem [67] (*a.k.a.* mismatch of the training and the target distribution) can affect the performance after a certain threshold of removing training data.

2. We implement novel reweighted performance scores for data Shapley based on the principles of importance sampling. First, we compare data Shapley and importance sampling and we recognize similarities (*e.g.,* both can inform us where the data are scarce and valuable for our objective) and differences (*e.g.,* importance sampling does not

quantify the contribution of training data points) between the two frameworks. Second, we leverage the importance ratio weights as an input to the data Shapley framework creating a powerful framework that provides training data valuation according to the cellular operators objectives.

In a nutshell, there is no a universal data valuation for crowdsourced mobile measurements, thus, application specific performance metrics must be carefully considered jointly with the prediction task when assigning a value to a data point.

Outline. The rest of this chapter is organized as follows. Section 6.2.1 introduces the formulation and the fundamentals of data Shapley. Section 6.3, presents the application specific error metrics with data Shapley for mobile coverage maps prediction. Section 6.3.3 demonstrates data minimization results. This chapter is finally concluded by Section 6.4.

6.2 Data Shapley Background

The Shapley value [65] is a solution concept in cooperative game theory[1]. The Shapley value assigns a numerical (monetary) valuation for the contribution among the different participants (players) in a cooperative game. The Shapley value is characterized by a collection of desirable properties and has motivated the research for data Shapley [35], which quantifies the contribution of each data point in algorithmic prediction.

[1] It was named in honor of Lloyd Shapley, who introduced it in 1951 and won the Nobel Prize in Economics for it in 2012.

6.2.1 Formulation

Data Shapley is a framework developed in [35], which attempts to provide an answer to the question: "How do we quantify the value of the data in an algorithmic predictions and decisions?" Data Shapley can provide us a valuation of the data (*i.e.,* assign an arithmetic value to each data point) in the setting of supervised ML. What is an equitable measure of the value of each train data point (*a.k.a.* datum) $(\mathbf{x}, y_i)$ to the training algorithm $\mathcal{A}$? In order to answer that, we have to take a closer look to the essential ingredients of a supervised ML algorithm: (a) training data, (b) learning algorithm, (c) performance metric. The prediction is a function that *depends jointly* on all of them, therefore, each one of them affects the equitable measure assigned to our data. We follow the exposition and the organization of [35]; the notation of the aforementioned components is as follows:

(a) Data: The dataset of the ML setting follows the typical setup we have already seen in this thesis: $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_1^N$. We denote with $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ the training and the test data respectively; in the Data Shapley framework we also need $\mathcal{D}_{eval}$ for the final evaluation, *i.e.,* the heldout data.

(b) Learning Algorithm $\mathcal{A}$: A *black box* for the data Shapley setting, which takes as input $\mathcal{D}_{train}$ and produces as ouput a predictor $\widehat{y} = \widehat{f}_y(\mathbf{x})$. For example, the algorithm could be a logistic regression or the `RFs` predictors we have developed throughout this thesis.

(C) Performance Score V: We can treat it as a black box that takes an input $\widehat{f}$, the error metric - valuation we want to apply to each data point, the test data $\mathcal{D}_{test}$ and outputs a performance score V. We denote with $V(S, \mathcal{A}) = V(S)$ the performance score of a predictor trained on train data S using the learning algorithm $\mathcal{A}$. Please note, that the performance score can be completely different than the loss function of the learning algorithm itself. For example, we can train `RFs` regression with the typical MSE minimization and then report also $RMSE$ as we did in Chapter 4, however, we could train a model on a different loss

function than the evaluation functions as we did in some cases in Chapter 5. Data Shapley provides us a powerful framework for evaluating the different quality functions and weights functions (which essentially are other forms of error metrics according to a target distribution of interest).

Goal. The goal is to compute the data Shapley value $\phi_i(\mathcal{D}, \mathcal{A}, V) \equiv \phi_i(V) = \phi_i \in \mathbb{R}\ \forall i, (\mathbf{x}_i, y_i) \in \mathcal{D}_{train}$, which follows the equitable valuation properties (described in Sec. 6.2.2).

Leave-one-out Value. A simple way to get a proxy of a data point value is through the leave-one-out method, which calculates the datum value by leaving it out and estimates the performance score, *i.e.,* $\phi_i^{\mathrm{LOO}} = V(D) - V(D - \{i\})$. However, leave-one-out does not satisfy the equitable valuations, inspired by the original Shapley value and described next. Intuitively, a data point interacts and influences the training process, which creates the predictor function, in conjunction with the other training points. Thus, these conditions should be taken into account for data valuation.

6.2.2 Equitable Valuation Conditions

We follow the organization and the notation of [35]. The equitable properties of data Shapley are defined as follow:

1. **Null Property.** If a datum $(\mathbf{x}_i, y_i)$ does not change the performance it should be given $\phi_i = 0$. Thus, $\forall S \subseteq \mathcal{D} - \{i\}, V(S) = V(S \cup \{i\})$

2. **Symmetry Property.** If two distinct datums i, j contribute equally to the performance score V then they should have equal data Shapley values. In other words, $\forall i, j \in \mathcal{D}$ such as $S \subseteq \mathcal{D} - \{i, j\}$ and $V(S \cup \{i\}) = V(S \cup \{j\}) \Rightarrow \phi_i = \phi_j$.

3. **Summation and linearity.** It is very typical in Machine Learning to have an overall performance score which is the sum of seperate performance scores. A typical example is the mean squared error (MSE) that is the summation of the weighted equally (*i.e.,* the mean) squared losses of individual points. When V consists of the summation of individual scores, then, the overall value of a datum should be the sum of its values for each score. In other words, for data Shapley we should have: $\phi_i(V+W) = \phi_i(V)+\phi_i(W)$ for performance scores V, W. It should be re-emphasized, that the data Shapley value is defined for the train data points according to the performance scores on the test data. Thus, $V = -\sum_{k \in test} l_k$ with l_k to be the predictor's loss on the kth test point; the data Shapley value for quantifying the value of the ith source for predicting the kth test point is denoted with $\phi_i(V_k)$. If datum i contributes values $\phi_i(V1)$ and $\phi_i(V_2)$ to the predictors of the test points 1 and 2 respectively, then we expect the data Shapley value of i in predicting both test points, *i.e.,* when $V = V_1 + V_2$, to be $\phi_i(V_1) + \phi_i(V_2)$.

6.2.3 Data Shapley Approximation

According to [35] the data Shapley value which complies to the above three properties, must have the form:

$$\phi = C \sum_{\mathcal{D}-\{i\}} \frac{V(S \cup \{i\}) - V(S)}{\binom{n-1}{|S|}} \tag{6.1}$$

In other words, the data Shapley value must he the average of the leave-one-out value (*a.k.a.* marginal contributions) of all possible training subsets of data in S. It should be noted that an exhaustive computation of Eq. (6.1) is very expensive computationally. An approximation - truncated Monte Carlo algorithm (TMC-Shapley) is provided by [35] and is being used in this thesis.

6.3 Data Shapley Applications for Coverage Maps

6.3.1 Prediction Tasks & Error Metrics for Coverage Maps

In the context of mobile coverage maps, this thesis has already considered prediction tasks for which the notions of “data valuation” and “some measurements are more valuable than others” were implied but not explicitly formulated. First, we took a glimpse of how a small portion of the data may bring the majority of the error reduction in the prediction. For example, in Fig. 4.5, with the 10% of the train data size the error was approx. equal to 2.7dB; for 10% → 50% the error decreased to 2.1dB, however, for a larger training set the improvement was negligible. Second, in Chapter 5 we introduced weight functions, via the importance sampling framework, that specified operators objectives. The weights *correct* the data distribution in order to *match* the target distribution of the objective of interest. For example, if we calculate the spatial uniform error ε_u (see Sec. 5.3), instead of the naive cross-validation error ε_{CV}, under-sampled regions would be assigned higher weights.

Data Shapley and importance sampling share many common characteristics but also have some distinct differences. We argue that they can complement each other and create a very powerful framework. Data Shapley requires three main components, namely (a) a dataset, (b) a training algorithm and (c) a performance score (evaluation metric) and *quantifies the contribution of individual training data points to a learning task.* Apart from the data valuation itself, according to [35], data Shapley has other benefits too: 1) it gives more insights into the importance of each data point than the common leave-one-out score; 2) it can identify outliers and corrupted data; 3) it can inform how to acquire future data to improve the predictor. On the other hand, importance sampling is being used to as a technique to modify the predictor **(b)** (*i.e.,* define a weighted loss function) and to *define* **(c)** the evaluation metric, when the train and the target distribution miss-match. However, it does **not** provide a valuation of the training data for the learning task (as data Shapley does) since it does not consider which datums helped the most for the final test error. Although they do not provide direct answers to (1) and (2), please note that importance weights are directly comparable to (3) above, since they indicate where the data are scarce and valuable for our objective.

Basically, data Shapley can leverage importance ratio weights as an input (for the predictor's loss function and/or the performance score itself) creating a powerful framework which provides training data valuation according to operators' objectives. Data Shapley inherently *requires* a performance score to evaluate the test data, therefore, our importance sampling framework with the family of weight functions provide the context for the data Shapley value. There is no a universal data valuation and for different learning tasks (*i.e.,* objectives) some data points might be more valuable than other.

In Section 6.3.3 we showcase data minimization for coverage indicator QoS prediction. Some of the questions we are interested in are: (i) Does a subset of data points bring significant benefits to the prediction? (ii) Can we remove low valued data and improve simultaneously the performance, the privacy-utility tradeoff and save in storage? In Section 6.3.4 we showcase how our weight functions can be used as evaluation metrics for data Shapley for mobile coverage maps. We provide a data valuation scheme for potential transactions between mobile analytics companies and cellular operatorsaccording to different objectives and scenarios.

Thus, for coverage maps we consider and evaluate the data Shapley valuation for the following prediction tasks and their corresponding evaluation metrics. First, we start with the classification of the mobile coverage indicator $Q_c(y)$ (see Sec. 5.2 for its formulation) and we calculate data Shapley values for the evaluation metric of (1a) accuracy and (1b) Recall or the class-0 (*i.e.,* R_0). Particularly R_0 is of immense importance for cellular operators as we studied in Chapter 5. For coverage maps regression (signal strength values) we calculate data Shapley with mean squared error (MSE) and the reweigthed uniform error ε_u as defined in 5.

6.3.2 Data Minimization Setup

For data minimization, we showcase results for mobile coverage QoS maps (*i.e.,* $Q_c(y)$ with $y = y^P$), which were introduced in Chapter 5. This setup is a typical binary classification problem, where class 0 corresponds to bad coverage and class 1 encodes good coverage. As we demonstrated in Chapter 5, minimizing directly the error of $\widehat{Q}(y)$ provides significant benefits compared to predicting coverage with the proxy $Q_c(\widehat{y})$.

Metric of Interest: Recall for coverage loss R_0. For a class of interest Recall is a measure of completeness, *a.k.a.* the ratio of relevant instances $\bigcup$ retrieved instances over the relevant instances (*i.e.,* what's fraction of the relevant instances were actually retrieved). In other words, it is defined as $R = \frac{T_p}{T_p+F_n}$ where T_p is the true positive rate and F_n is the false negative rate, *for the class of interest.* Ideally from operators' perspective is to maximize the Recall for class-0 R_0 because the higher R_0 means fewer false negatives for the class-0, which can be translated to the statement that our algorithm did not classify a bad coverage ($Q(y) = 0$) as a good coverage area ($\hat{Q}(y) = 1$). In this setup, coverage holes (class-0) misclassified as good coverage areas (class-1) would impact reputation, revenue, and overall performance (*e.g.,* the need for a cell upgrade may not be detected).

Prediction with RFs. As a briefly recap, we can model $Q_c(y)$ to be estimated as $Q_c(y)|$ $\mathbf{x}\sim \mathcal{N}(RFs'_{\mu}(\mathbf{x}), \sigma^2_{\mathbf{x}})$; therefore the final prediction is given by $\widehat{Q_c}(y) = \widehat{f_Q}(\mathbf{x}) = RFs'_{\mu}(\mathbf{x})$; for further details we refer back to Sec. 5.2.4.

RFs Setup. We follow the same hyperaparameters selection procedure and setup as in Chapter 4 and 5. For the `Campus dataset`, we select $n_{trees} = 20$ and $\max_{depth} = 20$ and we build-train models per cID.

Data Shapley - TMC-Shapley Setup. We adapt the TMC-Shapley's library, provided by [35], in order to estimate the data Shapley values ϕ_i of each training data point $(\mathbf{x}_i, y_i)$. Although the particular library implements `RFs` classification with accuracy as the evaluation metric, it neither provides `RFs` regression nor recall or other custom evaluation metrics as we do. Thus, we augment it with the recall R_0 evaluation metric for classification, `RFs` regression implementation and our reweighted-spatial uniform error ε_u, as defined in Chapter 5.

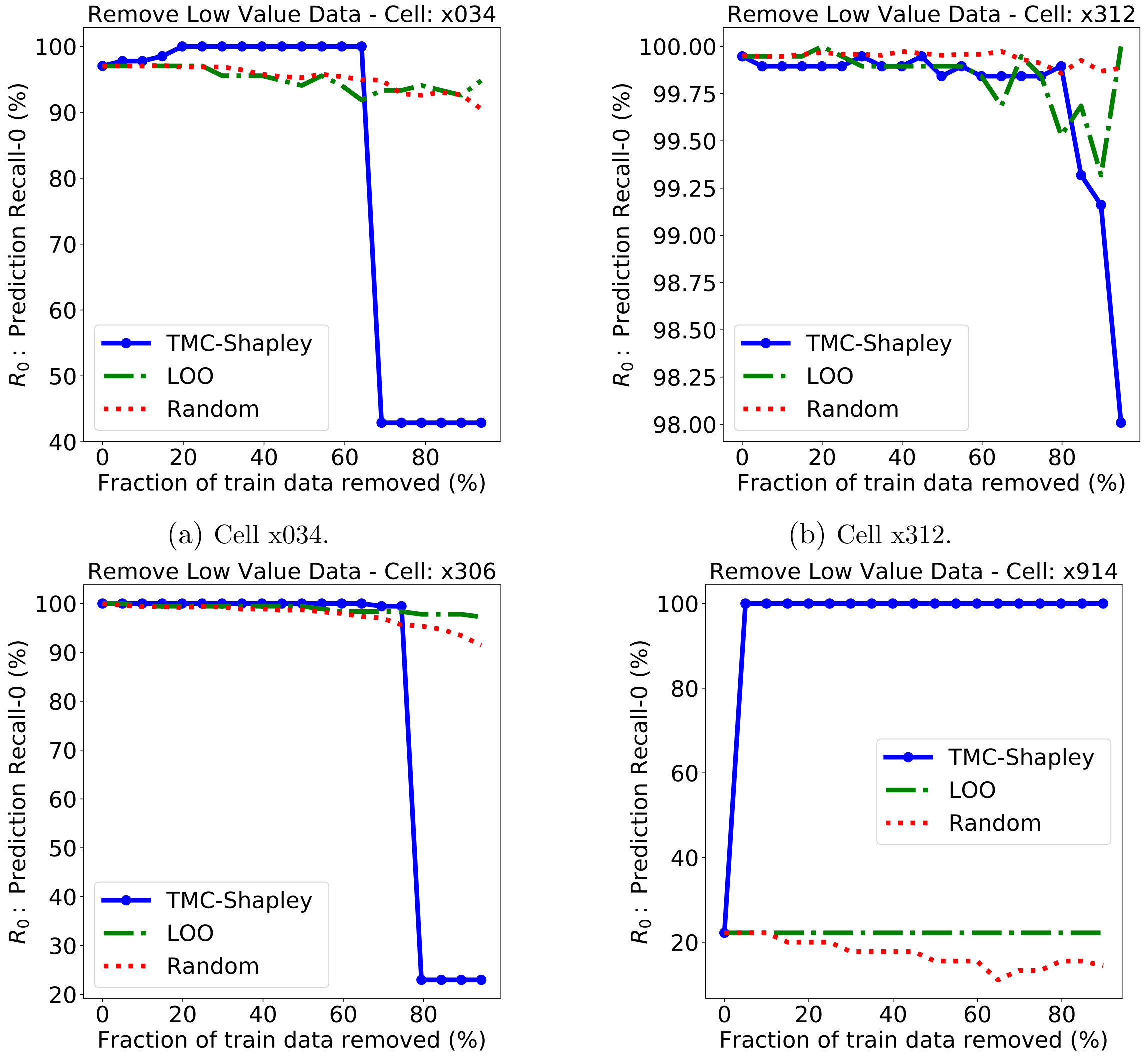
Moreover the TMC-Shapley algorithm is a Monte-carlo approximation, therefore, it generates Monte-carlo approximation until the average $\widehat{\phi_i}$ value has converged. Work in [35] suggests a convergence (stopping) criterion of $\frac{1}{n}\sum_{i=1}^{n}\frac{|\phi_i^t - \phi_i^{t-100}|}{|\phi_i^t|} < 0.05$ and they claim that the algorithm usually convergences with up to $3N_{train}$ iterations. However, our datasets are significantly larger than the data utilized in [35], which is in range of 1000-3000 data points; for example, the cell with the smallest number of measurements in `Campus dataset` contains approx. 1500 measurements and the majority of cells contain significant larger number of measurements, as can be seen in Table 4.4. Thus, we relax the convergence criterion to save execution time and we set a 30% convergence rate if we we exceed $2N_{train}$ iterations.

Splitting Data into Training $\mathcal{D}_{train}$, Testing $\mathcal{D}_{test}$ and Held-out Sets $\mathcal{D}_{\textbf{held-out}}$. We select randomly 60% of the data as the training set $\mathcal{D}_{train} = \{\mathbf{X}_{train}, y_{train}\}$, 20% as the testing set $\mathcal{D}_{test} = \{\mathbf{X}_{test}, y_{test}\}$ and 20% for the held-out data $\mathcal{D}_{\text{held-out}} = \{\mathbf{X}_{\text{held-out}}, y_{\text{held-out}}\}$. Please note the difference between the typical train-test split (as we did in Chapter 4 and 5) and the split here; data Shapley values ϕ_i are being calculated per training point $(\mathbf{x}_i, y_i)$ by calculating the performance score V of the prediction on $\mathcal{D}_{test}$. We use the $\mathcal{D}_{\text{held-out}}$ dataset to report the final data minimization results, *i.e.,* use some completely unseen data in order to report recall R_0 and accuracy A, while removing training data. $\mathcal{D}_{test}$ should not be used for the final evaluation since it was used to calculate the data Shapley in the first place.

Removing Low Value Data and Baselines. We utilize the TMC-Shapley algorithm to calculate the data Shapley values ϕ_i per training data point $(\mathbf{x}_i, y_i)$, for the problem of coverage classification (*i.e.,* $\widehat{Q_c}(y)$ as outlined in Sec. 6.3.1). For the data minimization process, we remove batches of 5% of the data points $\mathcal{D}_{train}$ starting from the least valuable (*i.e.,* lowest ϕ_i). At each step (*i.e.,* removal of a 5% batch), we re-train the $\texttt{RFs}_{all}$ model with the remaining $\mathcal{D}_{train}$ and we calculate the performance of the prediction on the $\mathcal{D}_{\text{held-out}}$ data. In the same way, we setup two natural baselines. First, Leave-one-out (LOO), defined in Sec. 6.2.1, produces a similar data valuation, therefore, we remove $\mathcal{D}_{train}$ batches according to ϕ_i^{LOO}, and second, we remove randomly selected 5% of the $\mathcal{D}_{train}$ at each step.

6.3.3 Data Minimizations Results

Remove Low Value Data *vs.* **Recall R_0.** For the `Campus dataset`, Fig. 6.1-6.2 present data-minimization results for several representative cells and for all the discussed methods (TMC-Shapley, LOO and Random), in terms of the recall R_0, as a function of the percentage of $\mathcal{D}_{train}$ removed. TMC-Shapley's performance either improves or remains the same when start removing low value data points compared to LOO and Random. There are two possible explanations. First, the batches with low valued $\mathcal{D}_{train}$ contain outliers and corrupted data; the data Shapley has correctly identified these points compared to LOO which does not show any benefit. Second, the data points with low ϕ_i do not have much predictive power to maximize the *defined performance* metric of interest for the *particular* learning task; essentially their removal lets the best suited data points to train the predictor. Very interestingly, after a certain threshold, TMC-Shapley's performance drops dramatically with just a removal of single batch, which means that this subset of points (highly "influential" points) hold significant predictive power. In contrast, by removing data randomly we keep bad quality data, however we might also keep some of these "influential" points and that explains that Random's performance neither improves nor decays very fast.



(a) Cell x034.

(b) Cell x312.

(c) Cell x306.

(d) Cell x914.

Figure 6.1: Campus dataset: Remove low valued data points (for Data-Shapley, LOO and Random) for various cells.

In order to grasp the root causes of the above results and understand better the underlying phenomena, we focus on a representative cell (cID x901) in Fig. 6.2. The label "A" in Fig. 6.2 refers to the beginning of the process where $\mathcal{D}_{train}$ is the full training dataset. The label "B" indicates the step where 65% of the data have been removed and the performance has reached its peak. Finally, the label "C" refers to the $\mathcal{D}_{train}$ after the sudden performance drop. Table 6.1 reports supplemental information for the cell x901 in Fig. 6.2 which includes: removed fraction and number of trianing data, the recall R_0, number of measurements per

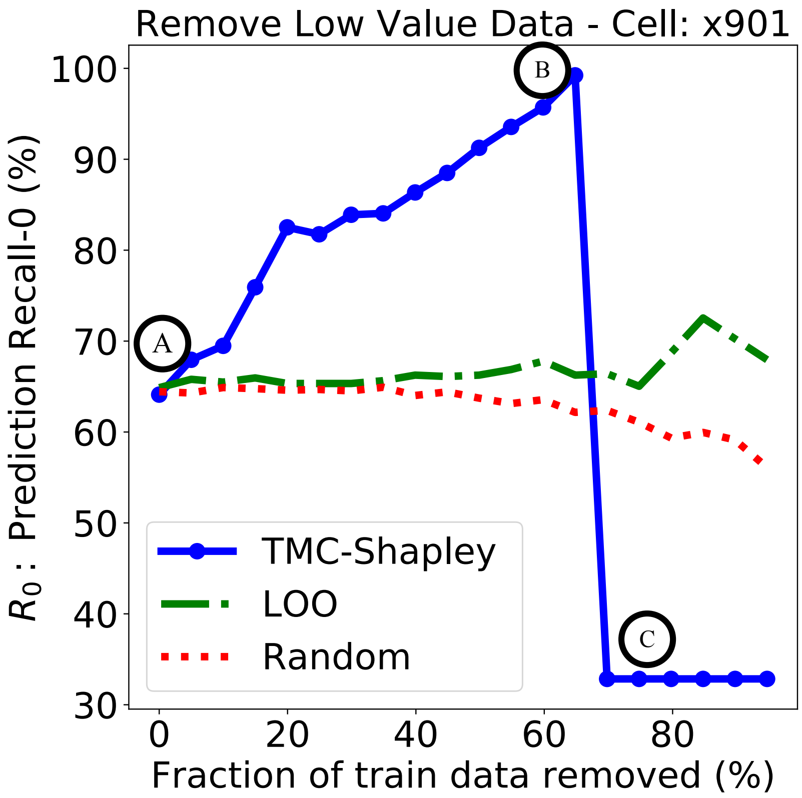


Figure 6.2: Campus Cell x901. We remove the lowest valued data points first in the case of TMC-Shapley and LOO as well as random. R_0 performance *vs.* fraction of data removed. Fig. 6.6 and FIg. **??** depicts the training measurements for the labels A, B and C on the above figure. Fig. 6.3 depicts the sampling density of the points for the same labels.

users as well as the number of 0s and 1s of both the held-out data and the predicted $\widehat{y}$, per each step of the removal process.

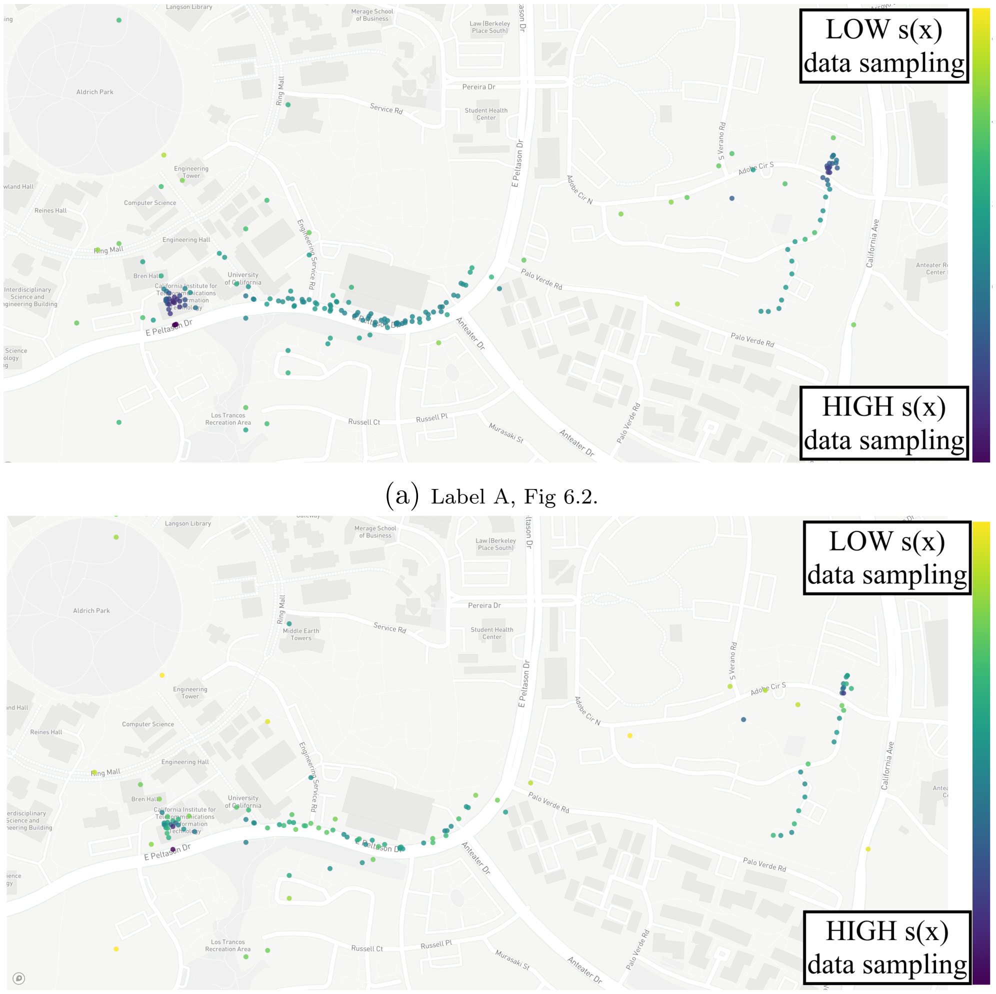
First, for the label B, where 65% of the data have been removed and R_0 has peak at 0.99, we notice that the predictor $\widehat{Q_c}(y)$ has predicted significant higher number of 0s than 1s (1631 0s *vs.* 294 1s). This does not surprise us, because, the predictor $\widehat{Q_c}(y)$ at label B is being trained with data points of higher quality for maximizing R_0. Essentially, in this scenario, data Shapley ϕ_i encodes the ability of the data to result in training predictors that would *minimize* the false negatives (*i.e.,* maximize recall) and tend to over-predict 0s than 1s. Apparently, for a different metric the low/high ϕ_i points could be different. When R_0 drops from 99% to 33% there is still data availability for both classes and users.

% Training Data Removed	N	R_0	userID-0	UserID-1	0s	1s	$\widehat{0}$	$\widehat{1}$
0.0	5777	0.64	5521	256	1938	3839	541	1384
0.05	5489	0.68	5246	243	1855	3634	601	1324
0.1	5201	0.69	4967	234	1752	3449	622	1303
0.15	4913	0.76	4697	216	1651	3262	733	1192
0.2	4625	0.83	4429	196	1550	3075	889	1036
0.25	4337	0.82	4159	178	1448	2889	885	1040
0.3	4049	0.84	3882	167	1337	2712	916	1009
0.35	3761	0.84	3611	150	1226	2535	918	1007
0.4	3473	0.86	3335	138	1146	2327	1007	918
0.45	3185	0.88	3059	126	1058	2127	1062	863
0.5	2897	0.91	2780	117	976	1921	1183	742
0.55	2608	0.94	2502	107	872	1737	1274	651
0.6	2321	0.96	2226	95	768	1553	1393	532
0.65	2032	0.99	1948	85	674	1359	1631	294
0.7	1745	0.33	1671	74	585	1160	195	1730

Table 6.1: x901 Cell detailed Data Minimization Results per removal step.

The sampling distribution of the data between label "A" vs label C offers also significant insights. Fig. 6.3a shows $w_u \propto \frac{1}{s(\mathbf{l})}$ for the $\mathcal{D}_{train}$ data at label A; the home and work locations where data have been oversampled are illustrated clearly; the average data density is $\mathbb{E}[\log s(\mathbf{l}) = -3.3]$. On the contrary, Fig. 6.3b depicts $w_u \propto \frac{1}{s(\mathbf{l})}$ for the remaining $\mathcal{D}_{train}$ at label C and it can be clearly seen that the data distribution is closer to uniform and the average data density has been dropped to $\mathbb{E}[\log s(\mathbf{l}) = -9.3]$. The held-out data were randomly sampled from the original distribution, therefore, there is now a miss-match between the original and target distribution (*i.e.,* the dataset shift problem we studied in Chapter 5) which can explain the drop in the performance. Last but not least, the data that are being removed from label B → label C (Fig. 6.6c) are primarily from the two oversampled regions, where we have a lot of held-out data to be tested (*i.e.,* if the most "influential points" are removed, significant predictive power can be lost).

Fig. 6.4 depicts the CDF values of ϕ_i values for the certain scenarios we have seen so far. Fig. 6.4a show the CDF of all data (*i.e.,* all $\mathcal{D}_{train}$ at label A). We can see that there is a

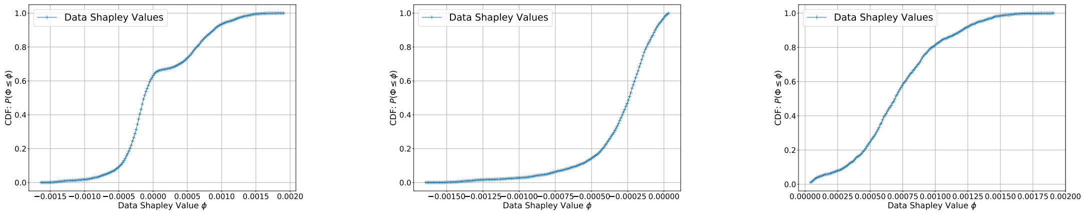


(a) Label A, Fig 6.2.

(b) Remaining Data's $s\mathbf{x}$ for performance at Label C, Fig 6.2.

Figure 6.3: `Campus dataset` cell x901. **Top**: Initial Sampling distribution $s(\mathbf{x})$ (Data for Label **A** in Fig. 6.2). $\mathbb{E}[\log s(\mathbf{x}) = -3.3]$ **Bottom**: Final Sampling distribution $s(\mathbf{x})$ (Data for Label **C** in Fig. 6.2). The procedure of removing data points eventually changed the sampling distribution of the data; at label A two regions were largely oversampled; at label C when the performance has finally been decreased the sampling distribution of the data look more uniform therefore it missmatches the original train distribution. $\mathbb{E}[\log s(\mathbf{x}) = -9.3]$.

portion of the data having negative ϕ_i value, however, the CDF sharply switches to positive values after a certain threshold. Very interestingly, the data points that have been removed at 65% removal (*i.e.,* label B, see Fig. 6.4b) have overwhelming negative values. Data Shapley has correctly identified that these points do not help the prediction for the *particular task* and the *performance score* (R_0) we consider. Fig 6.2 shows the CDF of ϕ_i at label B, *i.e.,* for the $\mathcal{D}_{train}$ the algorithm has achieved the best score and we observe that are all positive.



(a) CDF of ϕ_i for all data points, *i.e.,* Label A, Fig 6.2.

(b) Label A → Label B, Fig 6.2.

(c) after removing 65% of low valued data points.

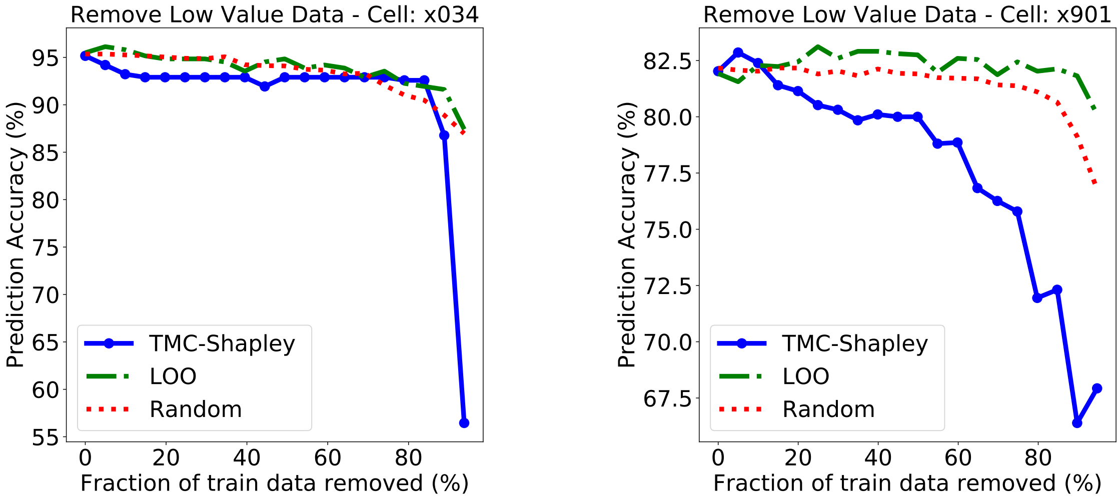
Figure 6.4: CDFs of Data Shapley $\phi_i$of the various scenarios in Fig. 6.2.

Please note that Fig. 6.4b includes a few positive values that were removed and there was a positive effect in R_0. This occurs because the TMC-algorithm itself is an *approximation* of data Shapley ϕ_i, which requires exponential number of computations (see eq. 6.1). In addition, we have slightly relaxed the convergence rate to save execution time given the bigger size of our datasets. Thus, it is expected that there are going to be some errors in ϕ_i values.

Removing Low Value Data *vs.* **Accuracy.** We also calculate data Shapley ϕ_i by using accuracy A as the performance metric V. Fig. 6.5 reports data for the same data removal process as we discussed so far. We observe that the TMC-Shapley's performance eventually outperforms LOO and Random when certain threshold of data removal has been reached, however after a certian point the performance of TMC-Shap drops, as happened with the recall. That is something we expect because the portion of the data that can be removed varies according to the particular dataset and most importantly for the particular error metric we evaluate. Different performance metrics produce different data Shapley valuations for the training data.

6.3.4 Weight Functions for Performance Score

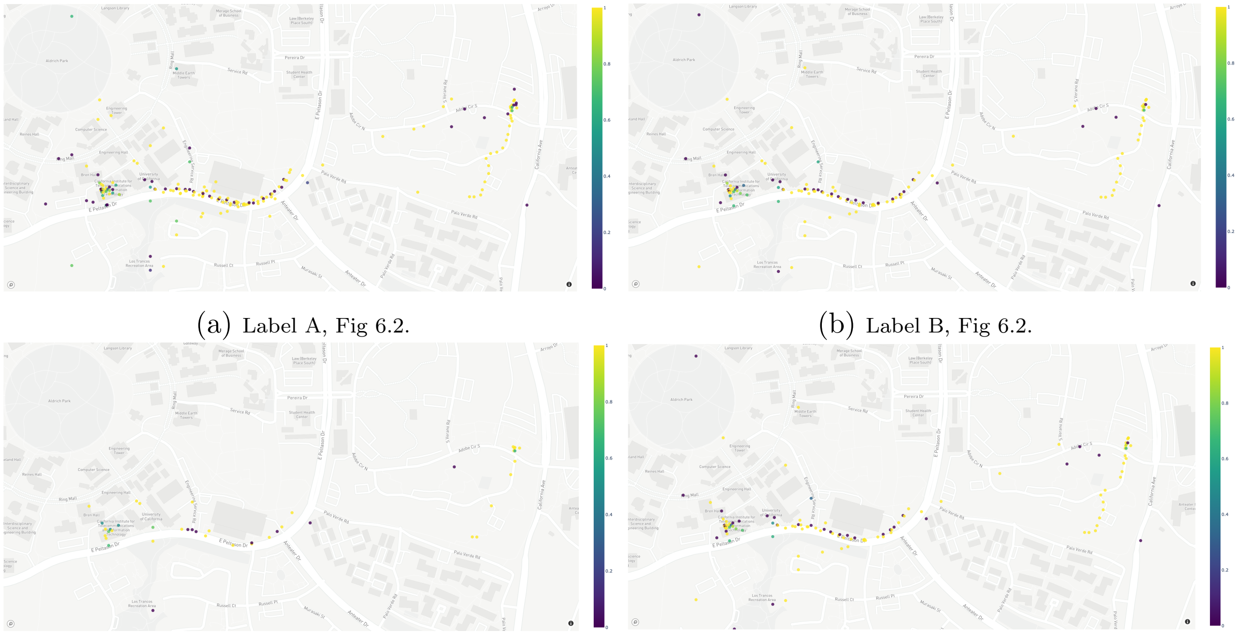
Data Shapley and importance sampling share many common characteristics but also have some distinct differences. We already argued that they can complement each other and create



(a) Cell x034.

(b) Cell x312.

Figure 6.5: **Campus dataset**: Remove low valued data points (for Data-Shapley, LOO and Random) for various cells and for the performance metric of accuracy A.



(a) Label A, Fig 6.2.

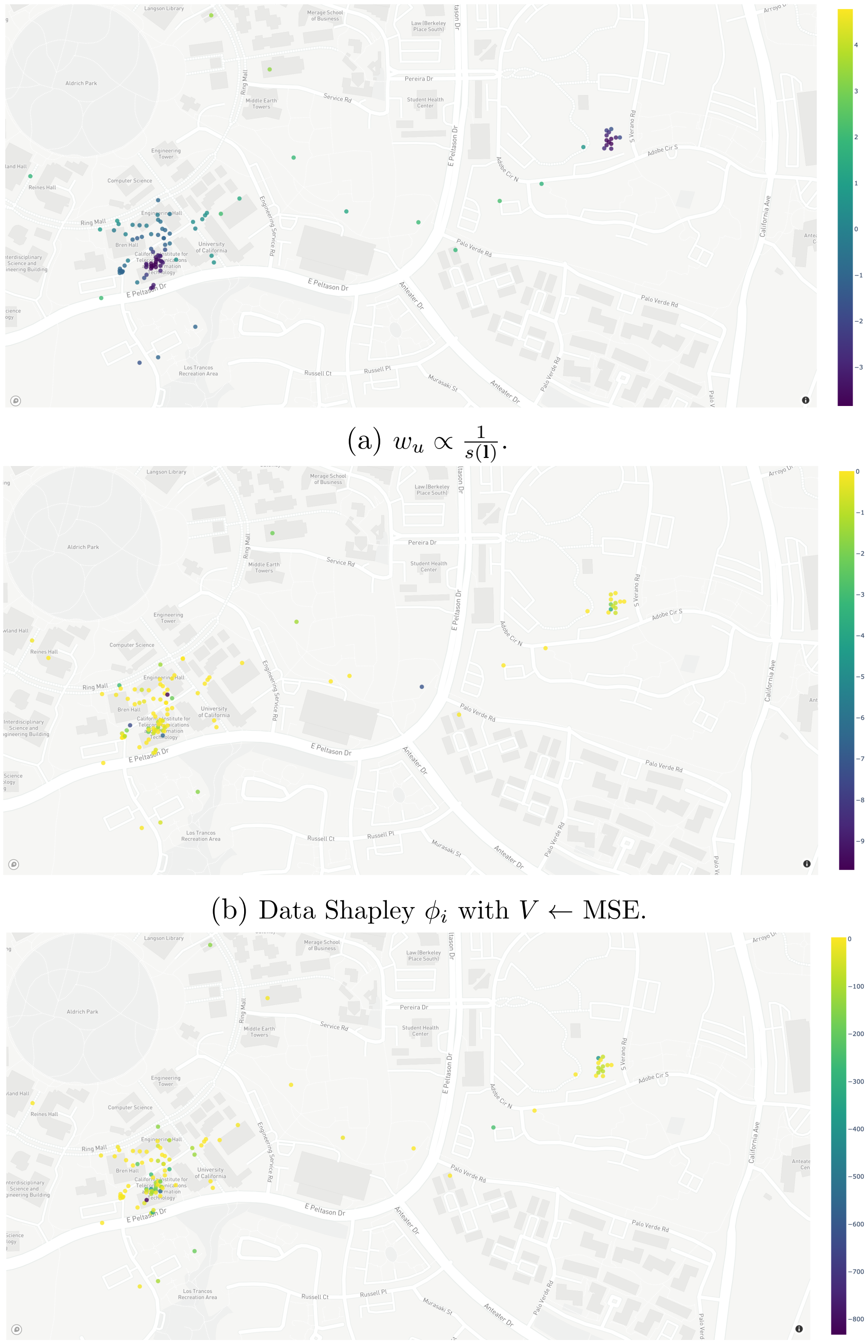
(b) Label B, Fig 6.2.

(c) Label B →Label C, Fig 6.2.

(d) Label C, Fig 6.2.

Figure 6.6: $Q_c(y)$ Values at different fractions-removed in Fig. 6.3. Please note that $Q_c(y)$= $\{0, 1\}$ but some spatial points might be experiencing both (that's the values between 0 and 1).

a very powerful framework for data valuation for mobile coverage maps prediction. Here, we showcase an example for how these different frameworks can be combined together by using the reweighted error metrics we developed in Chapter 5 as a a performance score for



(a) $w_u \propto \frac{1}{s(\mathbf{l})}$.

(b) Data Shapley ϕ_i with $V \leftarrow$ MSE.

(c) Data Shapley ϕ_i with $V \leftarrow \varepsilon_u$.

Figure 6.7: Example of how the reweighted error metric ε_u affects the data Shapley values. (a) w_u (*i.e.*, inversed data sampling). Please note the oversampled home and work locations. (b) Data Shapley values for the Mean Squared Error valuation. (c) Data Shapley for the ε_u performance score, *i.e.*, reweighted MSE. Please note how the oversampled areas (*i.e.*, with very low weights) at the home/work locations, have been assigned super small data shapley values. The performance score-evaluation function really matters.

data Shapley. Fig. 6.7 demonstrates a characteristic example. Fig. 6.7a shows the sampling distribution of the collected data and we can observe the oversampled data for the Calit2 building and the more sparse data at the surroundings of the building. Fig. 6.7b shows the data Shapley values for the typical performance metric of MSE. On the contrary, Fig. 6.7c shows the data Shapley values ϕ_i for the performance metric of ε_u (uniform spatila error) as defined in Chapter 5. As expected, the over-sampled areas have assigned a significant lower ϕ_i score because they do not contribute at the maximization of the performance score ε_u .

6.4 Summary

In this chapter we considered, for the first time, the problem of data Shapley valuation for mobile coverage maps. We defined jointly a specific prediction task and the performance-error metric of interest under the umbrella of data Shapley in order to quantify the data valuation of this particular predictive task. This approach is necessary since there is no universal data valuation score but rather it depends on the goal at a time. We calculated the data Shapley valuation for the mobile coverage indicator classification for evaluation metrics such as recall for coverage loss, R_0, which is of immense importance for cellular operators. We analyzed the distribution of data Shapley values in our datasets and we apply it for improving prediction and for data minimization. For instance, we were able to remove up to 65% of the low valued training data points and simultaneously improve the recall of coverage loss from 64% to 99%.

Last but not least, we showed how our novel reweighted performance scores based on importance sampling can be naturally combined with data Shapley, producing data valuations according to the importance ratio of the data points and we demonstrated an example for the uniform spatial error. Overall, assessing the data Shapley values of training data points enables improving prediction, (by removing data with negative Shapley values), data minimization (by removing data with lowest shapley values), and pricing of crowdsourced data. Therefore, we hope that our work can be used from both cellular operators and mobile analytics companies.

Chapter 7

Conclusions and Future Directions

Throughout this dissertation, we developed a principled machine learning framework to predict missing values for mobile coverage maps. We optimized mobile coverage maps prediction for objectives and error metrics of interest to cellular operators and we provided data Shapley valuation according to the specific prediction task.

In Chapter 4, we used the powerful tool of random forests (`RFs`), which we adapted in this context for the first time by evaluating different features readily available by Android APIs. We conclusively showed that the `RFs`-based predictors outperform state-of-the art data-driven predictors (geospatial interpolation) in all scenarios, when more features beyond just location are considered. We showed that the most important features were primarily cID, location, time and device type, which none of them can be naturally incorporated to geospatial interpolation. Most importantly, we demonstrated how we can significantly improve the tradeoff between prediction error and number of measurements needed compared to the state-of-the-art, *i.e.,* require 80% less data for the same error, or reduce the relative error by 17% for the same number of measurements. At the same time, we showed how device and wireless receiver's characteristics are very important and should be taken into account.

RFs regression minimizes the standard mean squared error (MSE), however, this does not always satisfy the goals of operators.

In Chapter 5, we address two limitations introduced by solely minimizing MSE. There are *certain* values of signal strength that might matter more than others (*e.g.,* low coverage areas) and sampled data (where MSE is calculated) do not correspond to the real target data distribution. Instead of evaluating prediction of signal strength itself w.r.t. conventional MSE, we introduced QoS functions (*e.g.,* call drop probability, coverage indicator) and importance ratio re-weighting (*e.g.,* for uniform, population, or arbitrary target distributions) that allows a cellular operator to express its operational objectives and optimize prediction accordingly.

We trained models directly on these QoS functions and showed an improvement up to 32% in terms of the relative error in the high CDP regime, which is of greatest concern to cellular operators, as well as improved recall from 76% to 92% for predictions of coverage loss (where false negatives are costly to operators). However, if signal strength (*e.g.,* RSRP) prediction is needed, our CDP QoS optimization framework provides an elegant way to improve the signal strength prediction *itself* (up to 3dB improvement), in its low values regime, where it matters more for the cellular operators. Our importance ratio re-weighting framework, apart from expressing the operational objective of interest, handles and mitigates the dataset shift problem [67]. The dataset shift is a prominent problem in the ML area and we hope our work can offer a framework to mitigate it specifically for mobile coverage maps. As we demonstrated in this thesis, the best practices of mobile analytics companies could introduce significant sampling biases; our technique of training models with reweighted loss decreases error by 20% for a uniform target distribution. We also showed how both adjustments operate together by implicitly changing the loss function optimized by the ML method, without modifying the underlying learning algorithm. QoS maps can be used directly by both mobile analytics and cellular operators for coverage maps and relevant applications, since a class of quality can provide information for certain tasks of interest (*e.g.,* visualizations, network decisions in SDN/SON, detect coverage holes *etc.*).

In the last part of our work, in Chapter 6, we considered, for the first time, the problem of data Shapley valuation for mobile coverage maps. Basically, we defined jointly a specific prediction task and the performance-error metric of interest under the umbrella of data Shapley in order to quantify the data valuation of this particular predictive task. We calculated the data Shapley valuation for the mobile coverage indicator classification for evaluation metrics of interest for cellular operators, such as recall for coverage loss. We analyzed the distribution of data Shapley values in our datasets and we apply it for improving prediction and for data minimization. For instance, we were able to remove up to 65% of the low valued training data points and simultaneously improve the recall of coverage loss from 64% to 99%. Last but not least, we showed how our novel reweighted performance scores based on importance sampling can be naturally combined with data Shapley, producing data valuations according to the importance ratio of the data points. Overall, assessing the data Shapley values of training data points enables improving prediction, (by removing data with negative Shapley values), data minimization (by removing data with lowest Shapley values), and pricing of crowdsourced data.

Throughout this thesis, we leveraged two types of real-world mobile (LTE) datasets to evaluate our methods: the first was collected at our university campus by an android App we developed and the second provided by a mobile crowdsourcing company for NYC and LA metropolitan areas, including approx. 11 million measurements. They are among the largest used and provided unique insights into city-wide coverage maps prediction. We hope that our work can useful to cellular operators and mobile analytics companies, to improve coverage maps prediction in a cost-efficient way.

7.1 Future Directions

There are many exciting research endeavors that can build on this thesis, particularly in the context of the upcoming 5G deployments.

Explicit Loss Functions. In this thesis, we introduced quality and weight functions and we showed how both adjustments operate together by implicitly changing the loss function optimized by the ML method. However, there are explicit loss functions that may be of interest to cellular operators and mobile analytics companies. For example, a Hubber asymmetric loss function for regression could put more emphasis in the low signal strength regime.

Hybrid Models, Transfer Learning and Applicability to 5G. Transfer learning [57] could also be explored. For example, consider a neighborhood where there are no collected (training) data and the data are collected from a totally different neighborhood (*i.e.,* we could not use the location of the measurements as features). In that case, we could build ML models that utilize features such as distance ($||\mathbf{l}_{\text{BS}} - \mathbf{l}_j||$), AoA (angle of arrival), $freq_{dl}$ *etc.*, which are location agnostic and omit the spatial coordinates feature. Thus, we would be able to generalize a prediction model to a new area by looking only at the similarities of different neighborhoods [22]; for example, a model trained on Seattle downtown (grid with skyscrapers) could be similar to SF downtown or NYC downtown. Moreover, hybrid models of data driven approaches and wireless propagation models can be explored.

Obtaining coverage maps in an accurate and cost-efficient way is a fundamental need in 5G and our method is directly applicable in that context. A major trend in 5G is to use a large numbers of small cells. Having accurate estimates of signal strength, coverage and other KPIs, will be necessary for knowing where to deploy more cells, and how to control 5G parameters. For example, we can use the coverage prediction for enabling LTE Direct communication or to turn on and off small cells [21] appropriately to save energy. Our framework can naturally handle prediction over small cells, *e.g.,* similarly to what we did with the NYC and LA datasets, where prediction was not per cell, but across an area covered by multiple cells, (*i.e.,* LTE TA) with cID used as a feature. Furthermore, small cells will introduce even higher sampling biases and our importance sampling framework handles an mitigates these training-target distribution mismatches. Last but not least, our quality, $Q(y)$, and weight functions, $W(\mathbf{x})$, with $\mathbf{x} \in \mathbb{X}$ (*e.g.,* geography, time, frequency band, device) could be expanded to express complex operator objectives in various 5G setups (*e.g.,* dense small cells, IoT, self driving cars).

Privacy-Preserving Coverage Maps. In Chapter 6, we presented tools for data minimization that can improve the privacy-utility tradeoff. One interesting direction is to further enhance the privacy aspects of mobile coverage maps with federated learning techniques (FL). Federated learning [11] is a technique for training a global ML model by sharing users' models updates instead of uploading raw data collecting on the devices, which is particularly important for crowdsourcing systems. A natural direction for future work is to"federate" the prediction methodology developed in this thesis to crowdsource the training of the model, without actually uploading the raw data from mobile devices to servers. Another idea for privacy-preserving signal maps would be to use the data Shapley valuation for data minimization, *i.e.,* to remove data for privacy reasons while preserving high predictive power. At the same time, data Shapley requires further research for faster approximation algorithms; although the TMC-Shapley algorithm offers a good framework, the execution time for datasets over 10 thousand points remains high (in the orders of dozens of days for a single core setup).

Bibliography

[1] A. Shubaa, A. Le, E. Alimpertis, M. Gjoka, A. Markopoulou. AntMonitor: System and Applications. *arXiv:1611.04268*, Nov. 2016.

[2] A. Achtzehn, J. Riihihjärvi, I. A. Barriía Castillo, M. Petrova, and P. Mähönen. Crowdrem: Harnessing the power of the mobile crowd for flexible wireless network monitoring. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 63–68, 2015.

[3] E. Alimpertis. Community RF Sensing. Diploma thesis, School of Electronic and Comp. Engineering, Tech. Univ. of Crete, Greece, 2012.

[4] E. Alimpertis. Smart Sensors of RF and Backscatter Signals with Localization. Master's thesis, School of Electronic and Computer Engineering, Technical Univ. of Crete, Chania, Greece, Aug. 2014.

[5] E. Alimpertis, N. Fasarakis-Hilliard, and A. Bletsas. Community rf sensing for source localization. *IEEE Wireless Comm. Letters*, 3(4):393–396, 2014.

[6] E. Alimpertis and A. Markopoulou. A system for crowdsourcing passive mobile network measurements. In *14th USENIX NSDI'17, Posters Sessions*, Mar. 2017.

[7] E. Alimpertis, A. Markopoulou, C. Butts, and K. Psounis. City-wide signal strength maps: Prediction with random forests. In *Proc. of the World Wide Web Conference*, WWW '19, pages 2536–2542. ACM, 2019.

[8] D. Applegate, A. Archer, D. S. Johnson, E. Nikolova, M. Thorup, and G. Yang. Wireless coverage prediction via parametric shortest paths. In *Proc. of the 18th ACM MobiHoc*, pages 221–230. ACM, 2018.

[9] G. Association. Definition of Quality of Service Parameters and their Computation, official document ir.42, Oct. 2016.

[10] AT&T. Enhanced control, orchestration, management & policy) architecture white paper. Tech. Report: `http://about.att.com/content/dam/snrdocs/ecomp.pdf`, Mar. 2016.

[11] E. Bakopoulou, B. Tillman, and A. Markopoulou. A federated learning approach for mobile packet classification. *arXiv preprint arXiv:1907.13113*, 2019.

[12] D. Balasubramanian. Qos in cellular networks. *Tech. Rep., Washington Univ. of Saint Louis*, 2006.

[13] T. Bilen, B. Canberk, and K. R. Chowdhury. Handover management in software-defined ultra-dense 5G networks. *IEEE Network*, 31(4):49–55, 2017.

[14] H. Braham, S. B. Jemaa, G. Fort, E. Moulines, and B. Sayrac. Fixed rank kriging for cellular coverage analysis. *IEEE Transactions on Vehicular Technology*, 66(5):4212–4222, 2017.

[15] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001.

[16] Y. J. Bultitude and T. Rautiainen. IST-4-027756 WINNER II D1. 1.2 V1. 2 WINNER II Channel Models. Technical report, 2007.

[17] R. J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD '13*, pages 160–172. Springer, 2013.

[18] A. Chakraborty, U. Gupta, and S. R. Das. Benchmarking Resource Usage for Spectrum Sensing on Commodity Mobile Devices. *HotWireless*, 2016.

[19] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das. Specsense: Crowdsensing for efficient querying of spectrum occupancy. In *Proc. of the IEEE INFOCOM '17*.

[20] C. Chen, A. Liaw, L. Breiman, et al. Using random forest to learn imbalanced data. *University of California, Berkeley*, 110(1-12):24, 2004.

[21] B. Cici, E. Alimpertis, A. Ihler, and A. Markopoulou. Cell-to-cell activity prediction for smart cities. In *Proc. of the IEEE INFOCOM WKSHPS '16*, pages 903–908. IEEE, Apr. 2016.

[22] B. Cici, M. Gjoka, A. Markopoulou, and C. T. Butts. On the decomposition of cell phone activity patterns and their connection with urban ecology. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc 15, page 317326, New York, NY, USA, 2015. Association for Computing Machinery.

[23] Cisco Inc. Visual Networking Index (VNI): Global Mobile Data Traffic Forecast Update, 2017-2022. `https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/190320-mobility-ckn.pdf`, accessed in May 2020, Mar. 2019.

[24] M. Clark and K. Psounis. Efficient resource scheduling for a secondary network in shared spectrum. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1257–1265, 2015.

[25] COST-231. Digital mobile radio towards future generation systems. Technical report, 1999.

[26] C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk. Gigascope: a stream database for network applications. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 647–651, 2003.

[27] H. Ekstrom. Qos control in the 3gpp evolved packet system. *IEEE Communications Magazine*, 47(2):76–83, 2009.

[28] A. Elnashar and M. A. El-Saidny. Looking at lte in practice: A performance analysis of the lte system based on field test results. *IEEE Vehicular Technology Magazine*, 8(3):81–92, 2013.

[29] R. Enami, D. Rajan, and J. Camp. RAIK: Regional analysis with geodata and crowdsourcing to infer key performance indicators. In *Proc. of the IEEE WCNC*, Apr. 2018.

[30] ETSI. LTE, Evolved universal terrestrial radio access (e-utra), physical layer measurements (3gpp ts 36.214 version 12.2.0 release 12), Apr. 2015.

[31] N. Fasarakis-Hilliard, P. N. Alevizos, and A. Bletsas. Variational inference cooperative network localization with narrowband radios. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2624–2628. IEEE, 2015.

[32] M. Fida and M. K. Marina. Impact of device diversity on crowdsourced mobile coverage maps. In *2018 14th International Conference on Network and Service Management (CNSM)*, pages 348–352, 2018.

[33] M. R. Fida, A. Lutu, M. K. Marina, and O. Alay. ZipWeave: Towards efficient and reliable measurement based mobile coverage maps. In *Proc. of the IEEE INFOCOM '17*, May 2017.

[34] A. Galindo-Serrano, B. Sayrac, S. B. Jemaa, J. Riihijärvi, and P. Mähönen. Harvesting mdt data: Radio environment maps for coverage analysis in cellular networks. In *Proc. 8th. on IEEE CROWNCOM*, pages 37–42, 2013.

[35] A. Ghorbani and J. Zou. Data Shapley: Equitable Valuation of Data for Machine Learning. 2019.

[36] A. Gomez-Andrades, R. Barco, P. Munoz, and I. Serrano. Data analytics for diagnosing the rf condition in self-organizing networks. *IEEE Transactions on Mobile Computing*, 16(6):1587–1600, 2017.

[37] B. F. D. Hähnel and D. Fox. Gaussian processes for signal strength-based location estimation. In *Proceeding of robotics: science and systems*, 2006.

[38] T. R. Hastie and J. Friedman. *Elements of statistical learning: data mining, inference, and prediction.* Springer, New York, 2003.

[39] S. He and K. G. Shin. Steering crowdsourced signal map construction via bayesian compressive sensing. In *Proc. of the IEEE INFOCOM '18*, pages 1016–1024, Apr. 2018.

[40] J. Huang, C. Chen, Y. Pei, Z. Wang, Z. Qian, F. Qian, B. Tiwana, Q. Xu, Z. Mao, M. Zhang, et al. Mobiperf: Mobile network measurement system. *Technical Report. University of Michigan and Microsoft Research*, 2011.

[41] A. Imran, A. Zoha, and A. Abu-Dayya. Challenges in 5G: How to empower SON with big data for enabling 5G. *IEEE network*, 28(6):27–33, 2014.

[42] A. P. Iyer, L. E. Li, and I. Stoica. Automating Diagnosis of Cellular Radio Access Network Problems. pages 79–87, 2017.

[43] Y. J. Jia, Q. A. Chen, Z. M. Mao, J. Hui, K. Sontinei, A. Yoon, S. Kwong, and K. Lau. Performance characterization and call reliability diagnosis support for voice over LTE. In *Proc. of the ACM MobiCom*, pages 452–463, 2015.

[44] J. Johansson, W. A. Hapsari, S. Kelley, and G. Bodog. Minimization of drive tests in 3gpp release 11. *IEEE Comm. Magazine*, 50(11), 2012.

[45] A. Katharopoulos and F. Fleuret. Not all samples are created equal: Deep learning with importance sampling. *arXiv:1803.00942*, 2018.

[46] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo. CRAWDAD dataset dartmouth/campus (v. 2009-09-09). Downloaded from https://crawdad.org/dartmouth/campus/20090909, Sept. 2009.

[47] M. Lichman and P. Smyth. Modeling human location data with mixtures of kernel densities. In *Proc. of the 20th ACM SIGKDD*, pages 35–44, 2014.

[48] Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors (technical report no. 1055). *University of Wisconsin*, 2002.

[49] R. Margolies, R. Becker, S. Byers, S. Deb, R. Jana, S. Urbanek, and C. Volinsky. Can you find me now? Evaluation of network-based localization in a 4G LTE network. In *Proc. of the IEEE INFOCOM '17*, pages 1–9, 2017.

[50] M. Molinari, M. R. Fida, M. K. Marina, and A. Pescape. Spatial interpolation based cellular coverage prediction with crowdsourced measurements. In *Proc. of the ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big Internet Data (C2BID)*, pages 33–38. ACM, Aug. 2015.

[51] K. P. Murphy. *Machine Learning: A Probabilistic Prespective.* The MIT Press, Cambridge, Massachusetts, 2012.

[52] A. Nika, Z. Zhang, X. Zhou, B. Y. Zhao, and H. Zheng. Towards commoditized real-time spectrum monitoring. In *Proceedings of the ACM workshop on Hot topics in wireless*, pages 25–30. ACM, 2014.

[53] A. Nikravesh, H. Yao, S. Xu, D. Choffnes, and Z. M. Mao. Mobilyzer: An Open Platform for Controllable Mobile Network Measurements. In *Proc. of the 13th ACM MobiSys*, pages 389–404. May 2015.

[54] O. Omotere, L. Qian, R. Jantti, M. Pan, and Z. Han. Big rf data assisted cognitive radio network coexistence in 3.5ghz band. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8, 2017.

[55] Open Signal Inc. Mobile Analytics and Insights, June 2011.

[56] A. Padmanabha Iyer, L. Erran Li, M. Chowdhury, and I. Stoica. Mitigating the Latency-Accuracy Trade-off in Mobile Data Analytics Systems. pages 513–528, 2018.

[57] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[58] F. Pedregosa et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct.):2825–2830, 2011.

[59] C. Phillips, M. Ton, D. Sicker, and D. Grunwald. Practical radio environment mapping with geostatistics. *Proc. of the IEEE DYSPAN '12*, pages 422–433, Oct. 2012.

[60] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[61] A. Ray, S. Deb, and P. Monogioudis. Localization of lte measurement records with missing information. In *Proc. of the IEEE INFOCOM '16*, Apr. 2016.

[62] Root Metrics by IHS Markit. Mobile phone network company, May 2020.

[63] Root Metrics Inc. Root Metrics Coverage Map Page, Oct. 2019. `http://webcoveragemap.rootmetrics.com/en-US`.

[64] S. O'Dea. Global unique mobile subscribers from 2010 to 2025, by region (in millions). `www.statista.com/statistics/740154/worldwide-unique-mobile-subscribers-by-region/`, accessed in May 2020, Feb. 2020.

[65] L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games*, 1953.

[66] B. Sklar. Rayleigh fading channels in mobile digital communication systems. i. characterization. *IEEE Comm. Magazine*, 35(7):90–100, 1997.

[67] J. Snoek and et al. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in NeurIPS*, pages 13991–14002. 2019.

[68] J. Sommers and P. Barford. Cell vs. wifi: on the performance of metro area mobile connections. In *Proc. of the 2012 ACM Internet Measurement Conference (IMC)*, pages 301–314, Boston, Massachusetts, USA, Nov. 2012. ACM.

[69] R. Srinivasan. *Importance sampling: Applications in communications and detection.* Springer Science and Business Media, 2013.

[70] Tutela Inc. Crowdsourced mobile data. `http://www.tutela.com`, June 2011.

[71] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In *IEEE INFOCOM '17*, pages 1–9, May 2017.

[72] Y. Wen, X. Tian, X. Wang, and S. Lu. Fundamental limits of rss fingerprinting based indoor localization. In *Proc. of the IEEE INFOCOM*, pages 2479–2487. IEEE, 2015.

[73] J. Yang, A. Varshavsky, H. Liu, Y. Chen, and M. Gruteser. Accuracy characterization of cell tower localization. In *Proc. of the ACM UbiComp '10*, pages 223–226, 2010.

[74] F. Yin and F. Gunnarsson. Distributed recursive gaussian processes for rss map applied to target tracking. *IEEE JSTSP*, 11(3):492–503, April 2017.

[75] X. Ying, S. Roy, and R. Poovendran. Incentivizing crowdsourcing for radio environment mapping with statistical interpolation. In *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 365–374, 2015.

[76] Z. Yun and M. F. Iskander. Ray tracing for radio propagation modeling: Principles and applications. *IEEE Access*, 3:1089–1100, 2015.

[77] C. Zhang and P. Patras. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In *Proc. of the 18th ACM MobiHoc*, pages 231–240. ACM, 2018.

[78] S. Zhang and et al. Cellular-enabled uav communication: A connectivity-constrained trajectory optimization perspective. *IEEE Trans. Commun.*, 67(3):2580–2604, 2018.

[79] T. Zhang and S. Banerjee. Inaccurate spectrum databases?: Public transit to its rescue! In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, page 6. ACM, 2013.

[80] T. Zhang, A. Patro, N. Leng, and S. Banerjee. A wireless spectrum analyzer in your pocket. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 69–74. ACM, 2015.