A Unified Prediction Framework for Signal Maps:

Not all Measurements are Created Equal

Emmanouil Alimpertis, *Member, IEEE*, Athina Markopoulou, *Fellow, IEEE*, Carter T. Butts, Evita Bakopoulou, and Konstantinos Psounis, *Fellow, IEEE*

Abstract—Signal maps are essential for the planning and operation of cellular networks. However, the measurements needed to create such maps are expensive, often biased, not always reflecting the performance metrics of interest, and posing privacy risks. In this paper, we develop a unified framework for predicting cellular performance maps from limited available measurements. Our framework builds on a state-of-the-art random-forest predictor, or any other base predictor. We propose and combine three mechanisms that deal with the fact that not all measurements are equally important for a particular prediction task. First, we design *quality-of-service functions (Q)*, including signal strength (RSRP) but also other metrics of interest to operators, such as number of bars, coverage (improving recall by 76%-92%) and call drop probability (reducing error by as much as 32%). By implicitly altering the loss function employed in learning, quality functions can also improve prediction for RSRP itself where it matters (*e.g.,* MSE reduction up to 27% in the low signal strength regime, where high accuracy is critical). Second, we introduce *weight functions* (W) to specify the relative importance of prediction at different locations and other parts of the feature space. We propose re-weighting based on importance sampling to obtain unbiased estimators when the sampling and target distributions are different. This yields improvements up to 20% for targets based on spatially uniform loss or losses based on user population density. Third, we apply the *Data Shapley* framework for the first time in this context: to assign values (ϕ) to individual measurement points, which capture the importance of their contribution to the prediction task. This can improve prediction (*e.g.,* from 64% to 94% in recall for coverage loss) by removing points with negative values and storing only the remaining data points (*i.e.,* as low as 30%), which also has the side-benefit of helping privacy. We evaluate our methods and demonstrate significant improvement in prediction performance, using several real-world datasets.

Index Terms—Cellular networks, LTE/5G/6G, signal strength, coverage, quality-of-service, signal maps, coverage maps, spatiotemporal maps, Data Shapley.

1 Introduction

Cellular operators rely on key performance indicators (*a.k.a.* KPIs) to understand the performance and coverage of their network, as well as that of their competitors, in their effort to provide the best user experience. KPIs usually include wireless signal strength measurements (*e.g.,* LTE reference signal received power, *a.k.a.* RSRP), other performance metrics (*e.g.,* throughput, delay) and other information (*e.g.,* frequency band, location, time, call drop probability *etc.*). Cellular performance maps (*a.k.a.* signal maps) consist of a large number of KPIs in several locations.

Traditionally, cellular operators collected such measurements by hiring dedicated vans (*a.k.a.* wardriving [1]) with special equipment, to drive through, measure and map the received signal strength (RSS) in a particular area of interest. However, in recent years they increasingly outsource the collection of signal maps to third parties [2]. Mobile analytics companies, such as OpenSignal [3] and Tutela [4], crowdsource measurements directly from end-user devices, via standalone mobile apps, or measurement SDKs integrated into other apps, such as games, utilities or streaming apps. Either way, signal strength maps are expensive for both operators and crowdsourcing companies to obtain, and may not be available for all locations, times, frequencies, and other parameters of interest. The upcoming dense deployment of small cells at metropolitan scales will only increase the need for accurate and comprehensive signal maps to enable 5G network management [5], [6].

For these reasons, there has been significant interest in signal map prediction techniques based on a limited number of spatiotemporal cellular measurements. These include propagation models [7], [8], data-driven approaches [9], [10], [11] and combinations thereof [12]. Increasingly sophisticated machine learning models are being developed to capture various spatial, temporal and other characteristics of signal strength [2], [13], [14] and throughput [15], [16]. In this paper, we build on a state-of-the-art RFs-predictor from [2] as our base "workhorse" ML model, and we develop a framework on top of it, to deal with the fact that not all measurements are equally important.

We observe that three different factors affect the importance of measurement data points when those are used for training ML predictors. First, *what* KPI we predict: operators are typically interested in performance metrics such as coverage, call drop probability, number of bars; these depend on but go beyond raw signal strength (RSRP). Second, *where* we make the prediction: the operators may be interested in predicting performance better in some locations (*e.g.,* those with weak coverage or at important sites), while they may have no control on how crowdsourced mobile measurements are distributed. Third, since measurements are expensive and may pose privacy risks, we may want to

Athina Markopoulou and Carter Butts are with the University of California, Irvine. Email: athina@uci.edu, buttsc@uci.edu. Emmanouil Alimpertis is with Apple. He was with the University of California Irvine, when this work was conducted. Email: ealimper@uci.edu. Evita Bakopoulou is with Google. She was with the University of California Irvine, when this work was conducted. Email: ebakopou@uci.edu. Konstantinos Psounis is with the University of Southern California. Email: kpsounis@usc.edu.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3221773

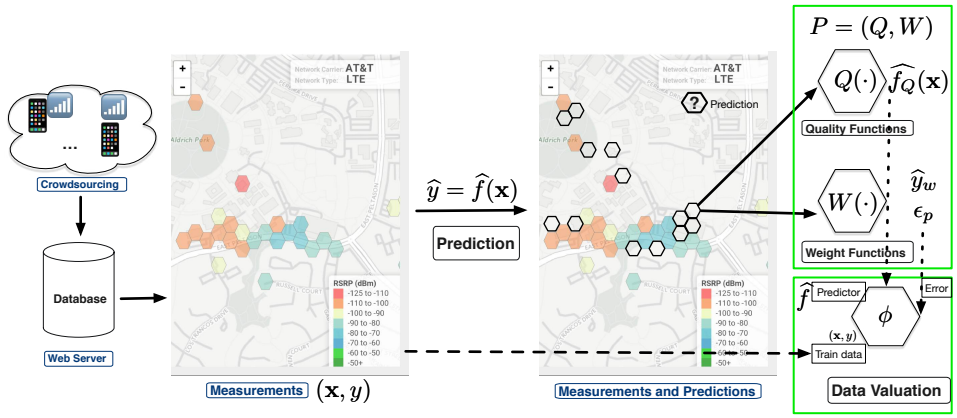


Fig. 1: **Overview of our framework:** the goal is to predict cellular values $\hat{y}$, based on features $\mathbf{x}$, while training on available (and unequally important) measurements. As base workhorse ML model, we use our state-of-the-art RF-predictor (Sec. 3.2.3), but any other predictor would do as well. Our framework builds on top of the base predictor and deals with the different importance of data points in three distinct ways. **(1)** We use quality-of-service functions Q to predict directly KPIs of interest; these depend on but are different from raw signal strength RSRP (Sec. 3.3). **(2)** We use weight functions W to target the mismatch between sampling and target distributions (Sec. 3.4). **(3)** We apply the Data Shapley framework to assess the importance (*i.e.,* predictive value) ϕ of the measurements w.r.t. the prediction task at hand (P, W) (Sec. 3.5).

identify those data points with the highest predictive *value* and discard outliers or redundant measurements.

We develop a unified framework for predicting cellular performance maps, which provides cellular operators and mobile analytics companies with knobs to express and deal with the unequal importance of available cellular measurements. We define two classes of functions Q and W, that jointly define the performance of the signal maps prediction problem $P = (Q, W)$. These functions tackle the mismatch between: **(1)** the operators' quality (QoS) metrics of interest and the raw signal strength (RSRP) and **(2)** the sampling and target distributions, respectively. Thus, operators can predict maps optimized beyond the standard MSE. In addition, **(3)** we compute the data Shapley values (ϕ) of measurement data points that capture their importance for training a predictor for the particular cellular map prediction problem $P = (Q, W)$ at hand. Our three contributions are summarized on Fig. 1 and are further elaborated upon next.

(1) Quality Functions Q. We consider *quality-of-service functions (Q)*, based on signal strength, which specify *what metric* operators and users care about, such as mobile coverage indicators (Q_c), call drop probability (Q_{CDP}), and number of bars (Q_B). Prior work exclusively minimizes the MSE for signal strength (RSRP) [9], [11], [12], [14], which does not directly optimize prediction for the aforementioned metrics. In contrast, we show that *learning directly on the Q domain* can significantly improve performance vs. state-of-the art, including: (i) recall gains from 76% to 92% and balanced accuracy gains from 87% to 94% for predictions of coverage loss (where false negatives are costly to operators); and (ii) reductions in relative error in predicted CDP by up to 32% (in the high CDP regime of greatest concern to cellular operators). Even for predicting *signal strength (RSRP) itself*, accuracy is often more critical in the low than in the high signal-strength regime. Specifying objectives via quality functions implicitly tunes the loss function, and allows operators to put more emphasis on values and use cases of signal maps that matter most. For example, we show improvement of prediction of RSRP itself in the low signal strength regime, of up to 27% (3dB in RMSE).

(2) Weight functions W. Sampling bias is inherent in crowdsourced data due to the non-uniform population density, as well as the commute and usage patterns. An operator may be interested in knowing KPIs at particular locations (as well as times, frequencies, and other features). Examples of target locations of interest for prediction include: locations where there is poor coverage, locations with dense user population, origins of calls to 911 dispatchers, client sites during working hours. However, these target locations may differ from the sampling locations where measurements are available, thus optimization to available data can lead to biased inference. This mismatch of the sampling distribution with the target distribution is also known as the dataset shift problem [17] in ML. To tackle this mismatch, we propose a re-weighting method, rooted in the framework of importance sampling, which leads to unbiased error. We introduce *weight functions (W)* that allow operators to express *where* (*e.g.,* in which particular locations, times, *etc.*) they are interested in predicting performance most accurately. We demonstrate improvement up to 20% for two intuitive target distributions: uniform loss across a spatial area and loss proportional to population density. Combining weight targeting and quality functions improves further, *e.g.,* up to 5% more for estimating CDP on targeted spatial losses.

(3) Data Shapley ϕ. We recognize and exploit the fact that not all measurements are equally important for predicting the metric of interest at the location of interest. We apply, for the first time in the context of signal maps, the *Data Shapley* framework, originally defined in economics and recently adapted to assign value to training data in ML [18]. Our Data Shapley framework takes as input the available cellular measurements, the ML prediction algorithm, and the error metric, re-weighted for the particular prediction problem $P = (Q, W)$; it then computes the Shapley values

(ϕ) of individual measurements used for training the ML model. The latter (i) enables us to remove measurements with negative or low Shapley values (*e.g.,* outliers), so as to train a better model and improve predictions; and (ii) can also minimize the amount of training data stored, which also has the side effect of enhancing privacy. For example, we show that we can remove up to 65-70% of data points, while simultaneously improving the recall of cellular coverage indicator from 64% to 99%.

Throughout the paper, we leverage two types of *large, real-world LTE datasets* to gain insights and evaluate prediction performance: (i) a dense Campus dataset, we collected on our own university campus; and (ii) several sparser city-wide (in NYC and LA) datasets, provided by a mobile data analytics company.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents our prediction framework, including the baseline predictor (Sec. 3.2.3) and the methodologies that deal with unequal importance of measurements, *i.e.,* the quality functions (Sec. 3.3), the weight functions (Sec. 3.4), and the data valuation (Sec. 3.5). Section 4 presents evaluation results. Section 5 concludes the paper. The Appendix (uploaded as "Supplemental Materials") provides details on: the datasets (A1), the Data Shapley formulation (A2), comparison of our base predictor to other state-of-the-art (A3), and additional results (A4).

2 Our Work in Perspective

Broadly speaking, signal strength can be predicted by propagation models or data-driven approaches, including geospatial interpolation, (*e.g.,* [9], [12]) and machine learning (*e.g.,* [2], [14]); or combinations thereof: [12].

Propagation models. State-of-the-art propagation and path loss (equation-based) models include WINNER I/II [8], Ray tracing [7] and others. However, this family of models requires a detailed map of the environment and fine grained tuning of model's parameters [2]. A simple yet widely used propagation model is LDPL [19] and its indoor variant [20].

Geospatial interpolation [9], [10], [12]. Methods such as Ordinary Kriging (OK) and OK Detrending [12], which are used by the ZipWeave [9] and SpecSense [10] frameworks, cannot naturally incorporate additional features beyond location (such as time, frequency, hardware *etc.*), as our ML models do (both in this paper and in our prior work [2]).

Machine learning: DNNs. Examples of RSRP/RSS prediction include [14] and [11]. Work in [14] uses DNNs along with detailed 3D maps from LiDAR and work in [11] uses Bayesian Compressive Sensing (BCS). ML models for the signal strength likelihood for user localization have also been developed by prior art [13], [21], [22]. Krijestorac *et al.* [23] propose state-of-the-art CNNs for estimating signal strength values and utilize a 3D map of the environment as features. In particular, they treat signal strength as a Gaussian random variable and predict its mean and variance. The evaluation is based on simulated data via ray-tracing software, which allows continuous infill of regions. However, when trained and evaluated on real-world datasets, this approach faces the limitation of sparse data. [24] uses autoencoders for predicting signal strength, but similarly their method is limited to simulated data.

	Features				Environment, Scale, Data			Evaluation
	Spatial	Time	Device, Network	LiDar, 3D Maps Agnostic	Environment Agnostic	City Wide	Real Data	Quality Beyond RSRP
Log-Distance Path-Loss ($LDPL$)	✓	X	X	✓	✓	X	✓	X
COST-231/ WINNER I-II/ Ray Tracing	✓	X	✓	X	X	X	✓	X
Geostatistics Specsense [10]	✓	X	X	✓	X	X	✓	X
BCS [11]	✓	✓	X	✓	X	X	✓	X
DNNs (RAIK [14])	✓	X	X	X	X	✓	✓	X
CNNs [23]	✓	X	X	X	✓	X	X	X
Auto-Encoders [24]	✓	X	X	X	✓	X	X	X
Our Framework	✓	✓	✓	✓	✓	✓	✓	✓

TABLE 1: Comparison of our framework to other approaches for signal map prediction.

Machine learning: Random Forests. In our prior work [2], we proposed random forests (RFs) for predicting signal strength (RSRP) based on a number of features, including but not limited to location and time (see Sec. 3.2.3). We demonstrated that RFs can outperforms prior art (including model-based and geospatial methods) because they can inherently capture spatial and temporal correlations, while also naturally extending to features beyond location (geospatial predictors can only handle location features). In Sections 3.2, 4.3 and Appendix A3, we expand on the description and evaluation of state-of-the-art geospatial interpolation methods (*e.g.,* [9], [10];) we also compare RFs to recent DNN and CNN-based predictors, and we demonstrate that RFs outperforms them when training on the (limited) available measurements.

Other Metrics - Dealing with unequal importance. Recent work in QoS has considered how RSRP measurements could be used as proxy to predict Quality of Experience (QoE) [25] or reinforcement-learning prediction techniques for video playback [26]. Apart from signal strength, prior work considered mobile traffic volume maps (*i.e.,* KPI throughput) [15], [16], [27], but solely focusing on the underlying ML model and MSE minimization. Importance sampling for deep neural network training is considered in [28]. The mismatch between sampling and target distributions is also related to the dataset shift problem [17] in machine learning.

Contributions. In this paper, we use our own implementation of RFs-predictor as a state-of-the-art "work horse" signal map predictor. For completeness, we present the method and its evaluation in Sec. 3.2.3 and Appendix A3. However, our focus here is orthogonal: we propose a framework that combines three methodologies (Q, W, ϕ) to deal with the unequal importance of available measurements for a particular prediction problem $P = (Q, W)$. Our framework builds on top of RFs, but could also utilize any other underlying ML algorithm minimizing a squared-error loss. To the best of our knowledge, this is the first paper to leverage QoS to improve prediction of RSRP and other error metrics in regimes where it matters; and to develop a (Q, W)-specific data Shapley valuation.

State-of-the-art approaches for signal map prediction are summarized in Table 1. Location privacy, including applications to signal strength, radiation maps, *etc.* [29],

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

TABLE 2: Terminology and Notation.

is an active research area and out-of-scope for this paper. Here we focus on optimizing prediction, in a centralized setting, given measurements of unequal importance. This being said, the Shapley framework also allows us to throw out some datapoints and store the minimum amount of data to train a good predictor, which saves storage and has the side-effect of data minimization (albeit after, not during, data collection).

3 Prediction Framework

3.1 Signal Maps Prediction

3.1.1 Definitions and Problem Space

An observed **signal map** is a collection of N measurements $(\mathbf{x_i}, y_i)$, $i = 1, 2...N$, where $y_i = \{y^P, y^I, y^C\}$ denotes the KPI of interest given the feature vector $\mathbf{x}_i$ (*e.g.,* location *etc.*) w.r.t. which the signal is to be mapped. In general, an operator's interest is not only 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 suggests approximation 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 on the observed data.

y: Key Performance Indicators (KPIs). There are many KPIs for LTE defined by 3GPP:

•*RSRP (y^P):* The reference signal received power is the average over multiple reference and control channels, reported in dBm. It is of great importance for LTE and utilized for cell selection, handover decisions *etc.*

•*RSRQ (y^I):* The reference signal received quality is a proxy to measure channel's interference.

•*CQI (y^C):* The channel quality indicator is a unit-less metric ($y^C = \{0, \cdots, 15\}$) of the overall performance. It is worth noting, that *all* prior work focused exclusively on predicting RSRP. Therefore, we use $y = y^P$, to refer to prediction of RSRP, unless otherwise noted.

$\mathbf{x}$: Measurement Features. For each measurement i in our datasets, we use several features available via Android APIs [2]. $\mathbf{x_i^{full}} = (l_i^x, l_i^y, d, h, cID, dev, out, ||\mathbf{l}_{BS} - \mathbf{l}_i||_2, freq_{dl})$. We consider the following features:

Training Options	y Domain	Q Domain
same $w = 1$ for all points	$P_B = (I, k)$	$P = (Q, k)$
	$\widehat{y} \to Q(\widehat{y})$	$Q(y) \to \widehat{Q}(y)$
training weights w (Table 4)	$P = (I, W)$	$P = (Q, W)$
	$\widehat{y_w} \to Q(\widehat{y_w})$	$Q(y) \to \widehat{Q}^w(y)$

TABLE 3: Overview of prediction problems $P = (Q, W)$. 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.

•*Location:* $\mathbf{l}_j = (l_i^x, l_i^y)$: GPS's latitude and longitude.

•*Time:* $\mathbf{t_i} = (d, h)$, where d is the weekday and h the hour the measurement was collected.

•*Cell ID and LTE TA:* KPIs are defined per serving LTE cell, which is uniquely identified by the CGI (cell global identifier, cell ID or cID) 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. LTE also defines Tracking Areas, LTE TA, by the concatenation of MCC, MNC and TAC, to describe a group of neighboring cells, under common LTE management.

•*Device hardware* type (dev): This refers to the device model (*e.g.,* Galaxy s21 or iPhone 13) and *not* to device identifiers. In [2], we considered all features and showed the most important ones to be location, time, cell cID and device hardware information dev. In this paper, we consider $\mathbf{x}$ to be the full set of features, unless otherwise noted.

Signal Map Prediction. Our goal is to predict an *unknown* signal map value $\hat{y}$ at a given location and other features ($\mathbf{x} \in \mathbb{X}$), based on available spatiotemporal measurements with labeled data $(\mathbf{x}_i, y_i)$, $i = 1, ...N$, either in the same cID or in the same LTE TA. The real world underlying phenomenon is a complex process $y = f(\mathbf{x})$ that depends on $\mathbf{x}$, and characteristics of the wireless environment.

It is important to consider the *loss* to be minimized by the choice of predictor: certain loss functions improve performance for certain objectives, while degrading it in others. We consider two factors relating to the choice of loss. First, an operators' interest is not always in KPI y itself, but in some *quality of service function*, $Q(y)$; see Section 3.3 for concrete examples. While conventional training schemes focus on predicting y (*e.g.,* w.r.t. mean squared error, or MSE), we consider signal map prediction that minimizes error in the predicted value of $Q(y)$ itself. We demonstrate that 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 weight accuracy for some values of $\mathbf{x}$ more heavily than others. While conventional training schemes implicitly assume 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 closely correspond to the distribution of sampled observations; see Table 3 and Section 3.4 for details.

Prediction Problem Space: $P = (Q, W)$. With the above motivation, we may formalize the prediction problem as follows. Let $\mathbb{Y}$ be the space of potential KPI values. $Q \in \mathbb{Q} : \mathbb{Y} \mapsto \mathbb{R}$ is a quality function, as described above. Similarly, $W \in \mathbb{W} : \mathbb{X} \mapsto \mathbb{R}_0^+$ is the above described weight

function. We define the *prediction problem space* as $\mathbb{P} = \mathbb{Q} \times \mathbb{W}$, whose elements $P = (Q, W) \in \mathbb{P}$ are *prediction problems*.

This representation provides a simple unifying formalism for a range of different problems. For instance, note the *base problem* $P_B = (I, k)$ where I is the identity function and k is a constant function, which amounts to the conventional learning problem assumed in the prior literature. Here, we develop not only predictors which minimize loss under P_B, but also or other arbitrary $P \in \mathbb{B}$. In practice, this amounts to finding 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 $W(\mathbf{x})$. Table 3 provides a taxonomy of all the prediction problems our framework can address. Table 2 summarizes the terminology and notation.

Transformation between problems. Any method for solving the base problem $P_B = (I, k)$ can be transformed to solve an alternative prediction problem, $P = (Q, k)$, by training on $Q(y)$ instead of y; we pursue this in Sec. 3.3. Likewise, we can transform a procedure for solving P_B to a procedure for solving $P = (I, W)$ by applying importance sampling, as described in Section 3.4. In addition, given a problem $P = (Q, W)$, we may transform any procedure for solving P_B to a procedure for solving P by (i) training on $Q(y)$ via the methods of section 3.3 and (ii) applying the importance reweighting of section 3.4 using W.

3.2 Base Predictor $P_B = (I, k)$

3.2.1 Model-Based Prediction LDPL

As a representative baseline from the family of model-based predictors, we consider the Log Distance Path Loss (LDPL) propagation model [30], which is simple yet widely adopted in the literature (*e.g.*, [19], [10]). LDPL 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 *a.k.a.* cell tower), as a log-normal random variable (*i.e.*, normal in dBm) [19]:

$$y_j^P(t) = P_0(t) - 10 n_j \log_{10}(||\mathbf{l}_{\text{BS}} - \mathbf{l}_j||_2 / d_0) + w_j(t). \quad (1)$$

We consider two cases regarding path loss exponent (PLE) n_j. (1) Homogeneous LDPL: Much of the literature assumes that the PLE n_j is the same across all locations. We can estimate it from Eq. (1) from the training data points. (2) Heterogeneous LDPL: Recent work (*e.g.*, [10], [19]) considers different PLE across locations. We considered several ways to partition the area into regions with different PLEs, and we present knn regression, where we estimate $\widehat{n_j}$ from its k nearest neighbors, weighted according to their Euclidean distance, which we defer to as "LDPL-knn".

3.2.2 Geospatial Interpolation

State-of-the-art approaches in data-driven RSS prediction [10], [31] have primarily relied on geospatial interpolation, which however is inherently limited to only spatial features (l^x, l^y). The best representative of this family of predictors is ordinary kriging (OK) [31] and its variants [10], *e.g.*, OKD, which are used as baselines for comparison in this paper.

Ordinary Kriging (OK): It predicts RSS at the testing location $\mathbf{l}_j = (l_j^x, l_j^y)$ as a weighted average of the K nearest measurements in the training set: $y_j = \sum_{i=1}^{K} w_i y_i$. The weights w_i are computed by solving a system of linear equations that correlate the test with the training data via the semivariogram function $\gamma(h)$ [10].

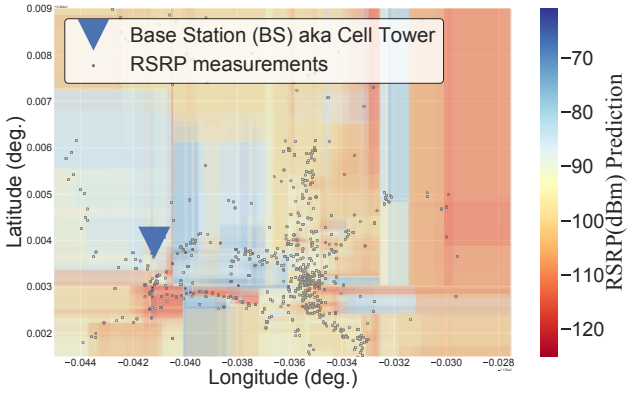


Fig. 2: Example of decision boundaries chosen by $\texttt{RFs}_{x,y}$ for Campus cell x306. We can see that RFs can naturally identify spatially correlated measurements, *i.e.*, regions with similar wireless propagation properties (*e.g.*, note how the model identifies the antenna's directionality/backlobe).

Ordinary Kriging Detrending (OKD) [10], [12]: OK assumes spatial stationarity, which does not hold for RSRP. OKD incorporates a version of LDPL in the prediction in order to address this issue [10].

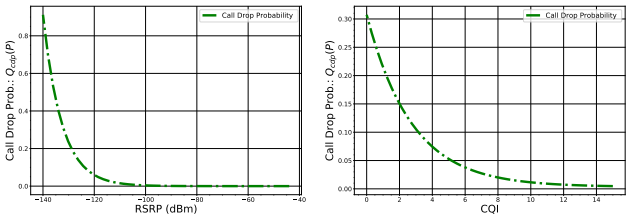
Both for the model-based as well as the geospatial interpolation, we refer to Appendix A3 for further details.

3.2.3 Proposed Predictor: Random Forests (RFs)

For the purposes of this paper, we use a state-of-the art signal map predictor, weintroduced in [2]: Random Forest (RFs) regression and classification. This predictor is an ensemble of multiple decision trees [32] and provides good trade-off between bias and variance by using bagging [32].

Why RFs for data-driven prediction: RFs naturally incorporate multiple features *vs.* just location in geostatistics and automatically produce correlated areas in the feature space with similar wireless propagation properties. RFs are scalable, need minimal hyper-parameter tuning, they do not overfit and they require minimum amount of data. For example, decision boundaries produced by $\texttt{RFs}_{x,y}$ (for LTE RSRP data) are depicted in Fig. 2. 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 UCI campus (*e.g.*, notice the backlobe of the antenna). Essentially, these axis-parallel splits assume that measurements close in space and time most likely should be in the same tree node, which is a reasonable assumption for signal strength statistics. Automatically identifying these disjoint regions with spatiotemporal correlated RSRP comes for free to RFs, and is particularly important due to wireless propagation properties. In contrast, prior art (*e.g.*, OKP, [9], [10]) requires additional preprocessing for addressing this spatial heterogeneity.

In [2], we introduced this predictor for the first time for signal maps, and we showed that it outperforms state-of-the-art predictors including model-based, spatiotemporal; see Section 4.3 in this paper. RFs also outperform DNNs when a limited number of measurements are available,



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

Fig. 3: Call Drop Probability (CDP) $\widehat{Q}(y)$ as a function of KPIs.

which is usually the case in the context of mobile analytics companies. DNN predictors, require a vast number of measurements per cell / small geographic area [14]. Moreover, RFs offer interpretability, as discussed in Fig. 2(a). In this paper, we use RFs as the underlying ML model for our framework. However, we emphasize that the ideas of our framework build on top and can be combined with any other learning strategy that can be applied on square-integrable real functions.

RFs predictor: Under the *base problem*, P_B, signal map value $y \in \mathbb{Y}$ to be estimated can be modeled as follows, given a set of feature vectors $\mathbf{x} \in \mathbb{X}$. $y|\mathbf{x} \sim \mathcal{N}(RFs_\mu(\mathbf{x}), \sigma^2_{\mathbf{x}})$, 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 $\sigma^2_{\mathbf{x}} = RFs_\sigma(\mathbf{x}) + \sigma^2_{RFs}$, where σ^2_{RFs} is the error from the construction of the RFs itself. The final prediction $\widehat{y} = \widehat{f}_y(\mathbf{x}) = RFs_\mu(\mathbf{x})$ is the maximum likelihood estimate.

3.3 Quality of Service Functions ($Q(y)$)

As described above, *QoS function*, Q, is a function of KPI y that reflects an outcome that depends on y. Examples of QoS of interest to cellular operators include the following.

Call Drop Probability (CDP). One of the most important cellular network quality metrics is the call drop probability. We model CDP with the exponential function, $Q_{cdp}(y) = ae^{-by} + c$, with parameters a, b, c estimated using empirical data from the literature [33], [34]. An example of CDP vs. KPIs (RSRP and CQI) is shown on Fig. 3. 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, call drop probability estimation $\widehat{Q}_{cdp}(y)$ can be treated as a ML regression problem.

LTE Signal Bars. Absolute RSRP values y are translated to the widely used signal bars $Q_B(y)$ on mobiles' screens. Mobile analytics companies usually produce 5-colors map to visualize signal bars [3]. See Appendix A1.2 for typical values of $Q_B(y)$ for iOS and Android devices.

Mobile Coverage. Detecting areas with weak/no signal (*i.e.,* bad coverage) is a major problem for cellular operators, and is essentially a binary classification. We define the mobile coverage indicator as a function of LTE RSRP [35]:

$$Q_c(y) = \begin{cases} 0 & \text{if } y^P <= -115 \text{ dBm, } i.e., \text{ 0 or 1 bar} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

The rationale is that the call drop probability increases exponentially and the service deteriorates significantly below -115dBm [33]. We want to detect areas of bad coverage because undetected $Q_c(y) = 0$ could impact the operator's reputation, revenue and overall performance.

Minimizing the error that matters, not MSE. 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 transforming the prediction problem from P_B to some alternative P; intuitively, we are implicitly modifying 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. 3, $y^P <= -100$dBm). Our experimental results in Section 4 show how the prediction of bad coverage ($Q_c(y) = 0$) can be improved for these regions that matter the most.

Prediction of $\widehat{Q}$ using Random Forests. We use RFs to predict $\widehat{Q}$, similarly to predicting $\widehat{y}$ in the previous section. Given prediction problem $P = (Q, k)$, $Q(y)|\mathbf{x} \sim \mathcal{N}(RFs'_\mu(\mathbf{x}), \sigma^2_{\mathbf{x}})$, 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 $Q(\hat{y}(\mathbf{x}))$) allows us to transform the base problem to any element of P involving constant weight function; this last restriction is lifted below.

3.4 Importance Sampling (Weights W)

We have argued above that not all values of y are equally important from a QoS perspective. Similarly, not all inputs $\mathbf{x}$ are necessarily equally important. For instance, an operator may be particularly interested in accurate predictions in some times or locations, *e.g.,* 911 locations, health facilities, areas with high revenue or competitive advantage during business hours, areas with dense user population *etc.* We generally refer to points in the input space generically as "locations"; however the dimensions involved may include space, time, frequency, device, *etc.*

Prior work [10], [14] 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. By turns, 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

TABLE 4: Importance sampling for operators(examples)

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 $d(\mathbf{x})$	$w_d \propto \frac{d(\mathbf{x})}{s(\mathbf{x})}$
Operator's custom target distr. $p(\mathbf{x})$	$w_c \propto \frac{p(\mathbf{x})}{s(\mathbf{x})}$

signal strength observations in $\mathbb{X}$ and the *target distribution* that captures the operator's desired loss function can be viewed as mismatches of the desired prediction problem $P(Q, W)$ versus the base problem P_B; to remove prediction bias, we show how direct estimation of $P = (Q, W)$ can be performed using techniques from importance sampling.

3.4.1 Importance-Reweighted Prediction Error

We are interested in assessing performance via error metric corresponding to an operator-defined objective, which is some measure of expected prediction error (i) integrated over the feature space $\mathbb{X}$, (ii) with some weight function that expresses how much the operator cares about different points in that space. Consider the modified prediction problem $P = (I, W)$ (where, for now, we leave $Q = I$). The expected prediction error over the target data distribution of interest $p(\mathbf{x})$ can for this problem be written as:

$$\varepsilon_p = \varepsilon\left(\mathbf{x}, W, \widehat{y}, y\right) = \int_{\mathbb{X}} W(\mathbf{x}) \mathbb{E}\left[L\left(\widehat{f}(\mathbf{x}) - f(\mathbf{x})\right)\right] d\mathbf{x} \quad (3)$$

where, $W(\mathbf{x}) \rightarrow \mathbb{R}^+$ is the weighting function for importance sampling, L is the loss function, $f(\mathbf{x}) = y \rightarrow \mathbb{R}^y$ and $\int_{\mathbb{X}} W(\mathbf{x})\, d\mathbf{x} < \infty$. 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. We can sample from $\mathbf{x}$ 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). In order to deal with the mismatch of the sampling and the target distributions, we use importance sampling techniques.

$$\widehat{\varepsilon_p} = \frac{1}{N} \sum_{i=1}^{N} \frac{W(\mathbf{x}_i)}{s(\mathbf{x}_i)} \left(\widehat{f}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2, \mathbf{x}_i \sim s(\mathbf{x}_i) \quad (4)$$

where N is the number of sampled data points, $s(\mathbf{x}_i)$ is the sampling distribution, $p(\mathbf{x}_i)$ is the target distribution and the adjustment factor $W(\mathbf{x}_i)/s(\mathbf{x}_i)$ 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 $s(\mathbf{x})$. This procedure allows us to transform any method for solving P_B into a method for solving $P = (I, W)$. The base problem weighting function is inappropriate for many practical tasks. Some intuitive examples of alternative choices of W are summarized in Table 4 and described next.

(1) ε_u ***uniform over*** $\mathbb{X}$**.** This is equivalent to the expected loss evaluated at a random location in $\mathbb{X}$, reflecting that 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 weigh 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_d ***proportional to population density.*** An intuitive target for operators is loss averaged over the user population, denoted by $d(\mathbf{x})$ at point $\mathbf{x}$ of the input space. We then want $W(\mathbf{x}) \sim d(\mathbf{x})$, thus importance ratio $w_d \sim \frac{d(\mathbf{x})}{s(\mathbf{x})}$. 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. Crowdsourced data collection is inherently biased due to human mobility and usage patterns.

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 which case s must generally be inferred. In the experimental results Section 4 we estimate $s(\mathbf{x})$ via adaptive bandwidth kernel density estimation (KDE) [36] on the 2D spatial space and the importance ratio is $w_u \propto \frac{u(\mathbf{1})}{s(\mathbf{1})}$. Our experimental results show that the main source of bias is location of devices.

Training Weighted Random Forests. The `RFs` algorithm splits each node utilizing a random set of features. The criterion of each split is to maximize the Information Gain (for classification), or to minimize the MSE (for regression). For N training samples, weighted `RFs` [37] adjust MSE for each split 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 $w_i = 1$.

3.5 Shapley Values of Cellular Measurements

The **Shapley Value** [38] is a celebrated framework in cooperative game theory, used to assign value to the contributions of individual players. Recently, it has inspired *data Shapley [18]*, which quantifies the contribution of training data points in supervised ML. More precisely, data Shapley provides a measure of the value ϕ_i of each training data point (*a.k.a.* datum) $(\mathbf{x}_i, y_i)$, for a supervised ML setting which consist of: (i) a training data set $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}_1^N$, (ii) a learning algorithm $\mathcal{A}$ that produces a predictor $\widehat{y} = \widehat{f}_y(\mathbf{x})$ or $\widehat{f}_Q(\mathbf{x})$ and (iii) a performance metric $V(\mathcal{D}, \mathcal{A})$. The prediction of the ML algorithm, and thus the value of the training data depend on all three. More precisely, 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 (null property, symmetry, summation and linearity); see Appendix A1, [18] for details.

Intuition and Computation. Intuitively, a data point interacts and influences the training procedure, in conjunction with the other training points. Thus, conditions which formulate the interactions among the data points and an holistic data valuation should be considered. A simple method is **leave-one-out**, which calculates the datum value by leaving it out and calculating the performance score, *i.e.,* $\phi_i^{\text{LOO}} = V(D) - V(D - \{i\})$. However, this formulation does not consider all subsets the point may belong to, thus does

not satisfy the equitable conditions [18]. According to [18], the data Shapley value must have the following form:

$$\phi_i = C \sum_{\mathcal{D}-\{i\}} \frac{V(\mathcal{D}\cup\{i\}) - V(\mathcal{D})}{\binom{n-1}{|\mathcal{D}|}} \quad (5)$$

In other words, the data Shapley value is the average of the leave-one-out value (*a.k.a.* marginal contributions) of all possible training subsets of data in D_{train}. Data Shapley, in its closed form in Eq. (7), would require an exponential number of calculations. An approximation - truncated Monte Carlo algorithm (TMC-Shapley) is provided by [18]. We adapt and extend the library for our custom error metrics, in order to estimate the data Shapley value ϕ_i of each training data point $(\mathbf{x}_i, y_i)$. More specifically, we augment it with the recall R_0 performance metric for classification, RFs regression and our reweighted-spatial uniform error ε_u for the performance metric V, as defined in Sec. 3.4.

Application to Cellular Measurements and Prediction. In [18], the Data Shapley value was tacitly defined for classification of medical data. In this paper, we apply, for the first time, Data Shapley valuation to mobile performance data, and not only for the base problem P_B but also for the general problem $P = (Q, W)$. The input to data Shapley in our context is: (i) the available cellular measurement data used for training (ii) the ML prediction algorithm (RFs) and (ii) the performance metric V, which in our case we define as the re-weighted error based on the operators' objectives, presented earlier in the paper. The output is a value assigned to each individual measurement training data point, used for the training for the particular prediction task (Q, W).

Data Shapley vs. Weight Functions: We have already described how both the choice of a loss function and of the evaluation metric really matter (Sec. 3.3 and 3.4.1). Data Shapley and weight functions share common characteristics but also differ, and can complement each other creating a powerful framework. Each framework independently can inform us whether the data are scarce and valuable for our objective, hence can inform how to acquire future data to improve the predictor. However, weight functions on their own do not not quantify the contribution of training data points, *e.g.,* if a training data point is an outlier. On the other hand, data Shapley inherently requires a performance score to evaluate the test data. There is no universal data valuation and for different learning tasks (*i.e.,* objectives) some data points might be more valuable than other. Hence our (Q, W) framework define the reweighted error, which provides the performance score for the data Shapley value.

Preview of results. After computing the Shapley values of our cellular measurements, we can then use them to remove those with negative or low Shapley values, in order to both (i) train a better model and improve prediction and (ii) minimize storage of the dataset, which has the side-benefit of enhancing privacy. The results in Section 4.7 show that we can remove up to 65-70% of data, while improving recall for coverage from 64% to 99%.

4 Performance Evaluation

4.1 Data Sets

We evaluate our framework using two types of real-world LTE datasets obtained in prior work [2], the characteristics of which are summarized on Table 5. Both datasets include cellular measurements, and we use the same subset of information from all datasets, *e.g.,* RSRP, RSRQ or CQI values and the corresponding features defined in Section 3. User identifiers were neither collected nor stored for this study.

Data Format. We use the same subset of information from all datasets, *e.g.,* RSRP, RSRQ or CQI values and the corresponding features defined in Sec. 3. An example, with some of the KPIs fields (obfuscated) is shown below:

```
{"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": [-73.9x,40.7x]}}
```

Listing 1: GeoJSON example with LTE KPIs and location, in MongoDB (obfuscated for presentation).

Properties of Datasets.

- *Data Density:* Measurements per unit area (N/m^2).
- *Cells Density:* Number of unique cells per unit area, $|C|/km^2$. The higher it is, the more cID helps as a feature.
- *Dispersion:* In order to capture how concentrated or dispersed are the measurements in an area, we use the Spatial Distance Deviation (SDD) metric [39], defined as the standard deviation of the distance of measurement points from the center.

Campus Dataset. We collected the first dataset on our university campus, via a user-space Android app we developed ourselves and used to collect data from volunteers [40]. This Campus dataset is relatively small: $180,000$ data points, collected by seven Android devices that belong to graduate students, using 2 cellular providers, and moving between student housing, offices and other locations on campus (approx. $3km^2$). However, this is a dense dataset, with multiple measurements over time on nearby locations. Due to space limitations, we refer to [2], [40] for details. [1]

NYC and LA datasets. These were collected by a major mobile crowdsourcing and data analytics company and shared with us.[2] They contain 10.9M measurements, covering approx. $300km^2$ and $1600km^2$ in the metropolitan areas of NYC and LA, respectively, for a period of 3 months with tens of thousands of unique cells (*i.e.,* unique cID). An example of the NYC neighborhood in East NYC is

1. This study was deemed "non human-subjects research" by IRB in our institution, since the dataset did not include user or device identifiers. However, the dataset did include pseudo-ids, that are useful in other studies that need multiple measurements from the same user. In the context of this paper, no PIIs os pseudo-ids were processed and only the fields summarized in the GeoJSON example were used.

2. Each measurement datapoint in the dataset contained 4 categories of data: (1) device data (2) connection data (3) quality data and (4) application data. From all the available fields, we only used those listed in the example GeoJSON above. The datasets were "anonymized" in the sense that there were no device or user identifiers provided to us. However, it is well- known that a combination of other fields (*e.g.,* device model, OS, applications installed, etc., can -in principle- be used as quasi-identifier to eventually de-anonymyze a dataset.

TABLE 5: Overview of datasets used throughout the paper. Campus is collected by us on our university campus. NYC and LA were provided by a Mobile Analytics Company

Dataset	Period	Areas	Type of Measurements	Characteristics
Campus	02/10/17 - 06/18/17	Univ. Campus Area $\approxeq 3km^2$	LTE KPIs: RSRP, [RSRQ]. Context: GPS Location, timestamp, dev, cid. Features: $\mathbf{x} = \left(l_j^x, l_j^y, d, h, dev, out, ||\vec{l}_{\text{BS}} - \vec{l}_j||_2\right)$	No. Cells $= 25$ No. Meas $\approxeq$ 180K Density ($\frac{N}{m^2}$) Per Cell: 0.01 - 0.66 (Table 6) Overall Density: 0.06
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:$\mathbf{x} = \left(l_j^x, l_j^y, d, h, cid, dev, out, ||\vec{l}_{\text{BS}} - \vec{l}_j||_2, freq_{dl}\right)$	No. Meas NYC $\approxeq$ 4.2M No. Cells NYC $\approxeq 88k$ Density NYC-all $\approxeq 0.014\frac{N}{m^2}$
		LA metropolitan Area $\approxeq 1600km^2$		No. Meas LA $\approxeq$ 6.7M No. Cells LA $\approxeq$ 111K Density LA-all $\approxeq 0.0042\frac{N}{m^2}$

depicted in Fig. 14. Examples of representative LTE TAs from NYC and LA datasets used in our evaluation are summarized later in Table 10. These are large datasets in terms of any metric, such as number of measurements, geographical scale, number of cells *etc.* As such, they provide novel insight into the problem at a scale that is relevant to operators and mobile analytics companies.

Anonymization issues. In this study, we did not use user or device identifiers from either dataset, we used only the fields of the GeoJSON in Listing 1. (Accordingly, we also applied for and obtained an IRB exemption.) However, in principle, it is still possible that other fields in each measurement can be used in conjunction as a "quasi-identifier" to map different measurements to the user they belong. Furthermore, quasi-identifiers can be linked to external datasets to de-anonymize users. Therefore, we have conservatively chosen to *not* release neither our own UCI Dataset (we are not liberty of releasing the company dataset anyway) nor trained models that might memorize sensitive features. This being said, we have established a process for researchers to request and obtain access to our UCI Campus dataset.

4.2 Evaluation Setup

RFs **predictor.** We train Random Forest (RFs) to predict the KPI $\widehat{y}$; then we compute $Q(\widehat{y}))$ or $\widehat{Q}(y)$ directly. We used the Python scikit-learn/SKLEARN packages [41]. We use state-of-the-art RFs [2] as the underlying predictor, but it could be replaced by other ML models.

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 [41] 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 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.

An important design choice is the granularity we choose to build our RFs models. As we demonstrated in [2], 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 for large number of measurements per cID. 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 paper, we utilize models per cID for the Campus dataset and per LTE TA models for NYC and LA datasets as [2]. Essentially, our framework can also learn the signal environment from the overlapping or neighboring cells, using cID as a feature.

To improve the reweighted prediction error ε_p according to operators' objectives (Section 3.4), we train weighted random forests RFs_w, with $w_i = \{w_{u_i}, w_{d_i}\}$ proportionally to the target distribution (see Table 4). In essence, the ML training weights are set equal to the importance ratio of each sample [28]. We compare RFs_w with the default RFs, where all samples are weighted equally.

Data Shapley Setup. For $\widehat{\phi}_i$ estimation with the TMC-Shap algorithm, work in [18] suggests a convergence (stopping) criterion of $\frac{1}{n}\sum_{i=1}^{n}\frac{|\phi_i^t - \phi_i^{t-100}|}{|\phi_i^t|} < 0.05$ with an observation that the algorithm usually convergences with up to $3N_{train}$ iterations. However, our datasets are significantly larger; [18] use approx. up to 3000 points and on the contrary the cell x901 demonstrated later contains approx. 15000 measurements. Thus, we relax the convergence criterion to save execution time and we set a 30% convergence if we exceed $2N_{train}$ iterations.

Splitting Training vs. 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 $Q_c(y), Q_{cdp}$(y)). The reported results are averaged over $S = 10$ random splits.

Our choices differ for data Shapley where we split the data as following: 60% of the data for $\mathcal{D}_{train} = \{\mathbf{X}_{train}, \mathbf{y}_{train}\}$, 20% for $\mathcal{D}_{test} = \{\mathbf{X}_{test}, \mathbf{y}_{test}\}$ and 20% for the held-out set $\mathcal{D}_{\text{held-out}}$. Data Shapley values ϕ_i are being calculated per training point $(\mathbf{x}_i, y_i)$ w.r.t. 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 since $\mathcal{D}_{test}$ was used for the data Shapley ϕ_i itself.

Evaluation Metrics - Coverage Classification. We evaluate the performance of $Q_c(y)$ in terms of binary classification metrics, *i.e.,* recall, precision, F1 score and balanced accuracy. *Recall* 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* is defined as $Pr = \frac{T_p}{T_p + F_p}$. *F1 Score* is the weighted average of precision and recall and *Balanced Accuracy* the average of recall for each class.

Evaluation Metrics for Regression. *(I) Root MSE (RMSE).* 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 $y^P$and RSRQ $y^I$and unitless for CQI y^C. *(II) Absolute Relative Improvement (ARI):* This captures the improvement of each

TABLE 6: Campus dataset: Comparing predictors the Base Problem, $P = (I, k)$, for all cells ((cID) of the Campus dataset. One can see that our RFs predictor, in the last column(s), achieves lower MSE than all alternative predictors: model-based (LDPL, LDPL-knn), geospatial(OK, OKD), and DNN. This holds for all cells and densities (N/m^2) and dispersion (SDD) observed.

Cell	Cell Characteristics				Prediction Error $RMSE$ (dB)								
cID	N	$\frac{N}{m^2}$	SDD	$\mathbb{E}[y]$	σ^2	LDPL hom	LDPL kNN	OK	OKD	DNN	$RFs_{x,y}$	$RFs_{x,y,t}$	RFs_{all}
x312	10140	0.015	941	-121	12	17.5	1.63	1.70	1.37	2.05	1.58	0.93	**0.92**
x914	3215	0.007	791	-94	96	13.3	3.47	3.59	2.28	6.48	3.43	1.71	**1.67**
x034	1564	0.010	441	-101	338	19.5	7.82	7.44	5.12	11.59	7.56	3.82	**3.84**
x901	16051	0.162	355	-108	82	8.9	4.60	4.72	3.04	5.69	4.54	1.73	**1.66**
x204	55566	0.666	325	-96	24	6.9	3.84	3.85	2.99	4.44	3.83	2.30	**2.27**
x922	3996	0.107	218	-103	30	5.6	3.1	3.16	2.01	4.51	3.10	1.92	**1.82**
x902	34193	0.187	481	-112	8	21.0	2.60	2.47	1.64	2.8	2.50	1.37	**1.37**
x470	7699	0.034	533	-107	17	24.8	3.64	2.73	1.87	3.33	2.78	1.26	**1.26**
x915	4733	0.042	376	-111	204	14.3	7.54	7.39	4.25	9.94	7.31	3.29	**3.15**
x808	12153	0.035	666	-105	8	4.40	2.41	2.42	1.60	2.84	2.34	1.75	**1.59**
x460	4077	0.040	361	-88	33	11.2	2.35	2.28	1.56	3.60	2.31	1.84	**1.84**
x306	4076	0.011	701	-99	133	18.3	4.85	4.30	2.80	7.07	3.94	3.1	**3.06**
x355	30084	0.116	573	-94	43	9.3	2.42	2.31	1.85	3.28	2.26	1.79	**1.79**

predictor over the variance in the data: $ARI = 1 - (1/|\mathbb{C}|)\sum_{i\in\mathbb{C}}(MSE_i/Var_i)$, where $|\mathbb{C}|$ is the number of the different cells in the dataset, and Var_i is cell i's variance.

(III) Mean Decrease Impurity (MDI), a.k.a. Gini Importance: It 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, averaged over all trees in the ensemble [41].

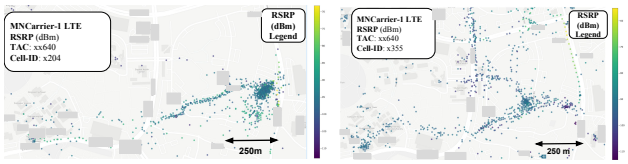
(IV) Reweighted Error ε_p for a target distribution $p(\mathbf{x})$. According to Eq. (4), $\varepsilon_p = \frac{1}{N}\sum_{i=1}^{N} w_i\left(\widehat{y}_i - y_i\right)^2$, with $w_i = \{w_{u_i}, w_{d_i}\} \propto \{\frac{1}{s(\mathbf{l}_i)}, \frac{d(\mathbf{l}_i)}{s(\mathbf{l}_i)}\}$, as defined in Table 4, 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. We use only location density $s(\mathbf{l})$ to calculate the (i) uniform error ε_u or (ii) w_d, over the space $\mathbb{X}$, but our methodology is applicable to an arbitrary space $\mathbb{X}$.

4.3 Results for Base Problem $P = (I, k)$

Prior work has exclusively focused on RSRP prediction minimizing MSE. Evaluating RFs for this base problem demonstrates that our framework outperforms existing solutions, learns the signal environment and produces a good signal map from limited measurements. Furthermore, we show that Random Forests [2] are a good choice for the underlying "workhorse"/baseline ML predictor for a range of spatiotemporal densities on top of which, we can develop our extended $P = (Q, W)$ framework. We compare RFs against several baselines: model-driven (LDPL-knn and LDPL-hom), geospatial interpolation (OK and OKD), and a multilayer DNN (trained with l^x, l^y features).

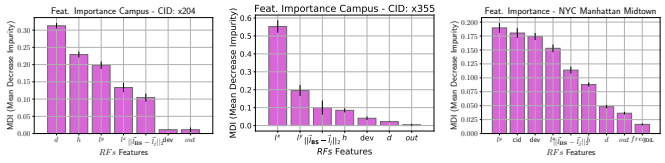
4.3.1 Random Forests vs. Other Predictors

a. Campus dataset*:* Table 6 compares all predictors for the cells in the Campus dataset, for the default 70-30% split. For each cell, it reports the measurement characteristics (number of measurements N, density N/m^2, dispersion (SDD), average signal strength value $E[y]$), and the prediction error ($RMSE$) for various predictors. We observe that our RFs-based predictors ($RFs_{x,y,t}$, RFs_{all}) outperform model-based (LDPL) and other data-driven (OK, OKD), or DNN predictors, for each cell and across all different measurements densities and dispersion. Fig. 9 also compares all methods, calculating $RMSE$ over the entire Campus dataset, instead of per cell.



(a) Example cell x204: high density (0.66), low dispersion (325). (b) Example cell x355: small density (0.12) higher dispersion (573).

Fig. 4: LTE RSRP Map Example from the Campus dataset.



(a) Cell x204 (b) Cell x355 (c) Manhattan Midtown

Fig. 5: Campus dataset (a): Feature Importance for cell x204 (dense) and (b) Feature importance for cell x355 (sparse); both cells' data are depicted in Fig. 4. NYC dataset: (c) MDI score for one LTE TA of MNC-1, in midtown Manhattan.

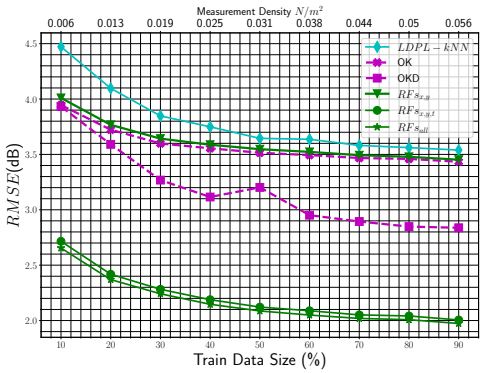
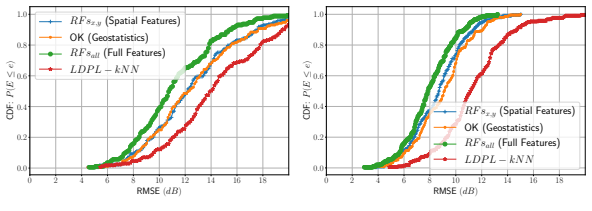


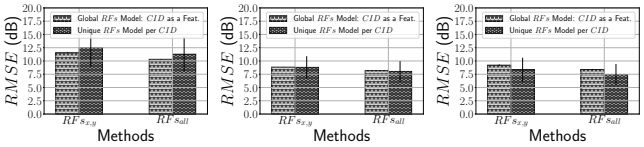
Fig. 6: Campus dataset: $RMSE$ vs. Training Size. vs. Measurement density N/m^2. Our methodology (RFs with more than spatial features, *i.e.,* $RFs_{x,y,t}$, RFs_{all}) significantly improves the RMSE-cost tradeoff: it can reduce $RMSE$ by 17% for the same number of measurements compared to state-of-the-art 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.

Fig. 6 shows the $RMSE$ as a function of the training size (as % of all measurements in the dataset) as well as absolute density (N/m^2). We note that the performance of OK and $RFs_{x,y}$ is almost identical, as it can be seen for $RMSE$ over all measurements (Fig. 6 and Fig. 9) and $RMSE$ per cell (Table 6). 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 $RFs_{x,y}$ uses multiple decision trees and data splits. In addition, considering additional features can significantly reduce the error. For the Campus dataset, when time features $\mathbf{t} = (d, h)$ are added, $RFs_{x,y,t}$ significantly outperforms OKD: it decreases $RMSE$ from 0.7 up to 1.2 dB. Alternatively, in terms of training size, $RFs_{x,y,t}$ 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



(a) NYC Manhattan Midtown (b) LA Southern Suburb

Fig. 7: NYC and LA datasets: CDFs for $RMSE$ per cID for two different LTE TAs, for the same operator. RFs$_{all}$ offer 2dB gain over the baselines (90th percentile).



(a) MNC-1, NYC Manhattan Midtown (urban) (b) MNC-1, East Brooklyn (urban/suburban) (c) MNC-2, Southern LA (suburban)

Fig. 8: RMSE in NYC and LA datasets. This figure compares: (1) urban vs suburban LTE TAs; (2) cID as feature vs. training a different RFs model per cID; (3) operatorsMNC-1 vs. MNC-2.

state-of-the-art geospatial predictors with 80% less measurements. The absolute relative improvement of RFs$_{x,y,t}$ compared to OKD is 17%, shown in Fig. 9(b). This RMSE-data size tradeoff can be used to imply the minimum sample size to meet an acceptable error as well as the objective of the operators (the reweigthed error ϵ_p in Section 4.5) and the data quality (see Section 4.7.1). Last but not least, RFs can naturally handle different data densities, as shown in Fig. 6, by automatically selecting the most important features (see also discussion and results in 4.3.2).

b. NYC and LA datasets*:* Fig. 8 shows results for different LTE-TAs for RFs per cID or with cID used as a feature. We can see that the higher the cells density (*e.g.,* urban) the more useful cID is as a feature; if there are many data available per cell, a model per cell is preferable. Fig. 7 shows the error for two different LTE TAs, namely for NYC Manhattan Midtown (urban) 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 performas very close to RFs$_{x,y}$, because they both exploit spatial features. However, RFs$_{all}$ with the rich set of features improves by approx. 2dB over the baselines for the 90th percentile, in both LTE TAs. Interestingly, the feature dev is important, which is expected since this data has heterogeneous devices reporting RSRP.

There are multiple reasons why RFs$_{all}$ outperform geospatial interpolation predictors. The mean and variance of RSRP depend on time and location, other features (dev, cID) and the complex propagation environment, which can be inherently captured by RFs as we explained in Fig. 2 and Sec. 3.2.3. OK also relies on some assumptions (same mean over space, semivariogram depending only on the distance between two locations), which do not hold for RSRP. Even hybrid geospatial techniques (OKD) cannot naturally incorporate additional features (*e.g.,* time, device type, *etc.*).

Similarly, the RFs-based predictors outperform the DNN-based predictor. First, DNNs cannot handle efficiently the disjoint regions and discontinuous RSRP values as RFs inherently do (*e.g.,* in Fig. 2). Second, DNNs need a

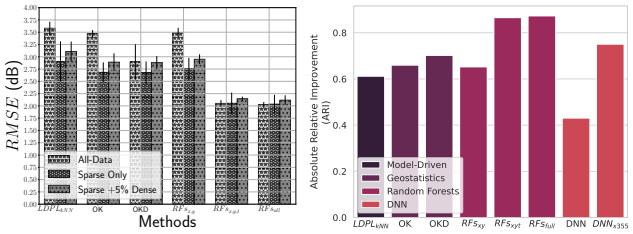


Fig. 9: Comparison of all predictors over the entire Campus dataset. Left (a) $RMSE(dB)$ under various scenarios, Right (b) ARI over all data points. Our Approaches (RFs$_{x,y,t}$, RFs$_{all}$) outperform prior art in all scenarios. Higher ARI is better.

huge amount of data in order to perform well with discontinuous and non smooth functions. Furthermore, more complex DNNs architectures would be needed along with very expensive to obtain features such as Li-DAR data or 3D maps *etc.* [14]. Finally, cannot support both categorical (*e.g.,* cID) and continuous features (*e.g.,* lat).

4.3.2 *Effect of the spatial and temporal density*

We have already shown that RFs achieves lower MSE when all data or a randomly sampled fraction of the data are used. In the following, we show that our methods achieve lower MSE also under different spatial and temporal densities.

4.3.2.1 Examples of dense measurements from Campus dataset: We train one RFs model per cID for the set of features $\mathbf{x} = (l_j^x, l_j^y, d, h, ||\vec{l}_{\text{BS}} - \vec{l}_j||_2, out, dev)$. Results w.r.t. MDI are shown on Fig. 5. We observe that, in cells with high data density and low dispersion, the most important features w.r.t. MDI are the time features (d, h). An example of such a cell is x204, depicted in Fig. 4(a), which has $SDD = 325$, density=0.66 points/m^2 and its MDI is shown in Fig. 5(a).

4.3.2.2 Examples of sparse and dispersed measurements: First, in the Campus dataset, an example such cell is x355, with $SDD = 573, density = 0.116N/m^2$ in Fig. 4(b); in this case, the location features (l_j^x, l_j^y) have higher MDI (Fig. 5(b)). Second, the NYC and LA datasets are also sparse and contain thousands of cells. As a representative example, we report the feature importance, in Fig. 5(c), for the LTE TA of a major mobile network carrier located in NYC Midtown Manhattan and depicted in Fig. 20 in the Appendix (see supplemental materials). The most important features turn out to be the spatial features (l_j^x, l_j^y) as well as the cell cID and dev. This is because the data are sparse and the whole LTE TA is served by geographically adjacent or overlapping cells.

4.3.2.3 Location density and overfitting: Our RFs predictors outperform the geospatial predictors for cells of all densities in Table 6. We also noticed that a significant fraction of the data comes from a few locations, *e.g.,* from participants' home and work, which begs the question whether this leads to overfitting. We investigated this question and found that our RFs predictors neither get a performance boost nor overfit.

To that end, we applied HDBSCAN [42], a state-of-the-art clustering algorithm, to identify very dense clusters of measurements. We refer to data in those clusters as "dense" and to the remaining ones as "sparse-only" data. Fig. 9(a) reports the $RMSE$ of different methods when training and

	Recall		Precision		F-1		Accuracy	**Balanced**
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 7: Campus Coverage $Q_c(y)$ results: (i) Recall for Class-0 (No-Coverage) 76% $\rightarrow$ 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).

testing is based on (i) all-data, (ii) sparse-only data and (iii) sparse data with a 5% random samples from the dense data. Our RFs$_{x,y,t}$ and RFs$_{all}$ have similar performance in all scenarios and consistently outperform baselines in all scenarios. Please note that OK and LDPL-knn's errors slightly decrease for "sparse-only"; OK cannot handle repeated locations and LDPL-knn may overfit. These findings are similar to the results per cell, in Table 6.

4.4 Results for QoS functions $P = (Q, k)$

4.4.1 *Coverage Domain*, $P = (Q_c, k)$

This setup is a typical binary classification problem, where class 0 corresponds to bad coverage and class 1 corresponds to good coverage. As a baseline, we train the RFs regression models, we predict $\widehat{y}$ and compute the proxy $Q(\widehat{y})$. We compare that with our proposed approach, which is to train RFs classifiers, with the same features, on quality-transformed observations $(\mathbf{x}_i, Q(y_i))$ and predicting $\widehat{Q}(y)$. For coverage indicator, we employ $y = y^P$ since it is defined on RSRP. RFs use the default training ($\forall i, w_i = 1$).

In this setup, bad coverage (class-0) misclassified as good coverage areas (class-1) impact reputation, revenue, and overall performance.Therefore, from the operators' perspective, it is desirable to maximize the Recall for class-0 R_0 because higher Recall means fewer false negatives, *i.e.,* our algorithm did not classify a bad coverage ($Q(y) = 0$) as a good coverage area ($\hat{Q}(y) = 1$).

(a) Campus dataset: Fig. 10 illustrates the improvement in the Campus dataset 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. 10(c) 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 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. 11, 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 our predictor. The overall classification results, in terms of the binary classification metrics, are shown in Table 7. We see an improvement of 16% for Recall R_0, per Fig. 10, as well an improvement in balanced accuracy from 87% to 94%. These improvements do not come at the expense of F1 and Accuracy, which improve by approx. 1%.

(b) NYC and LA datasets: Table 8 lists the classification results for some examples from NYC and LA datasets. We see a similar increase up to 12% in terms of R_0 for our predictor $\widehat{Q_c}(y)$ compared to the baseline proxy.

	Recall		**Precision**		**F-1**		**Accuracy**	**Balanced**
Class Label	0	1	0	1	0	1		**Accuracy**
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 8: 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 observer in other LTE TAs .

4.4.2 *Call Drop Probability Domain*, $P = (Q_{cdp}, k)$

CDP Q_{cdp}(y) estimation is a continuous value prediction problem on the [0,1] interval. As with the coverage domain, we train RFs models 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 report the relative reduction in $RMSE$.

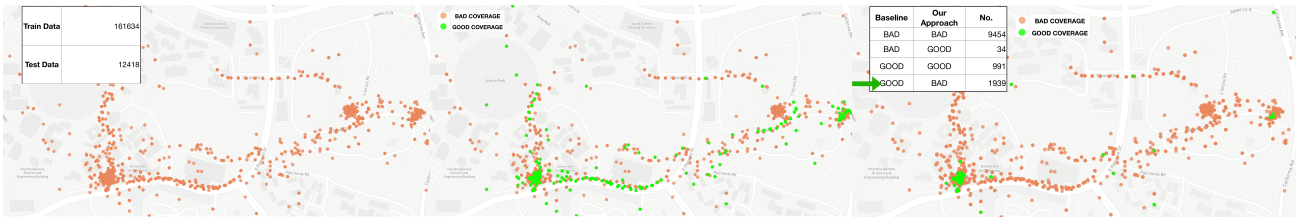
(a) Campus dataset: In Fig. 12, we report results for estimating CDP, when using the proxy baseline vs. predicting CDP directly. Fig. 12(a) shows the relative reduction in $RMSE$ error of CDP estimation *vs.* RSRP, which confirms our design choice. Our estimation $\widehat{Q}_{cdp}(y)$ reduces the relative estimation error up to 27% in the lower reception regime (0-1 vars, $y^P \leq -115$dBm), where the error function being minimized is highly sensitive to predictive performance.

4.4.2.1 *Transform $Q_{cdp}(y)$ to the RSRP $y^P = y$ domain* (i.e., $P = (Q, k) \rightarrow P_B = (I, k)$)*::* It is important to highlight that our QoS domain methodology can also improves *RSRP prediction itself* for values with high CDP that matter the most. An example is shown in Fig. 12(b): we compare the prediction error of $\widehat{y}$ values (RSRP) themselves, *vs.* inverting $\widehat{Q}_{cdp}(y)$ to return back to the original y^P space. We group the error by signal bars and we observe that the change in learning objective shifts the effort to reducing error where it is most critical (in 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 predicted CDP). We can hence view our procedure as allowing us to train on an application-specific loss function, without modifying our underlying learning algorithm.

(b) NYC and LA datasets: We also present results for CDP prediction on the NYC and LA datasets, and using RSRQ y^I and CQI y^C data in addition to RSRP y^P. Fig. 13(a) and Fig. 13(b) show $RMSE$ of CDP estimation with RSRQ y^I and CQI y^C respectively. 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. We see that our procedure successfully focuses improvement where it is needed for CDP prediction, rather than wasting statistical power on the high signal strength regime. Due to lack of space, results for $Q_{cdp}(y)$ and $Q_c(y)$in LTE TAs for NYC and LA are in App. A4.1.

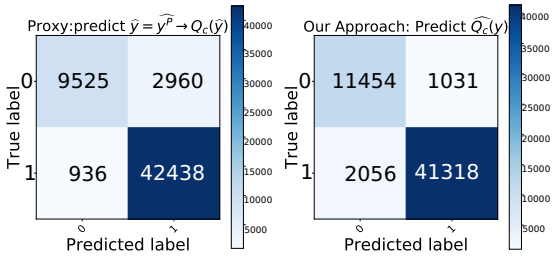
4.4.3 *Discussion: why minimizing MSE can be naive.*

In signal strength prediction, an error of few (say 5) dB will not reflect much change in QoS when the user's received



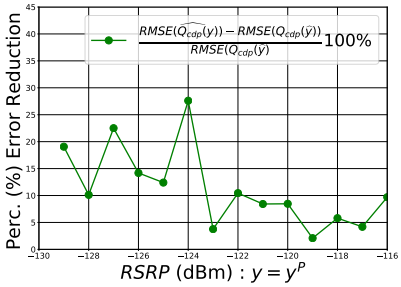
(a) Actual Spots with Bad Coverage. (b) Baseline Prediction $Q(\widehat{y})$. (c) Our Approach $\widehat{Q_c}(y)$

Fig. 10: LTE Coverage Map for our own Campus dataset. Display only Test Data. (a) Bad Coverage from Test Data (b) Baseline-Proxy- Prediction $Q(\widehat{y})$ (c) Our Model Prediction. It can be seen that (c) has more red points than (b), implying better classification. For this example, we discover 1939 data points which the baseline would not detect (16% of the total 12418 bad coverage points). Best viewed in color.

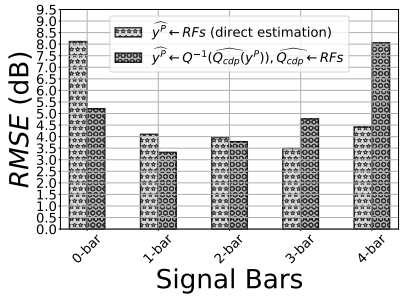


(a) Baseline-Proxy $Q_c(\widehat{y})$ (b) Our method $\widehat{Q}_{cdp}(y)$.

Fig. 11: For Campus dataset: Confusion Matrix for coverage, $Q_c(y)$ ($P=(Q_c,k)$). The points incorrectly classified as "good coverage" by the baseline $Q_c(\widehat{y})$ predictor are shifted to the "bad coverage" class under our predictor $\widehat{Q}(y)$.



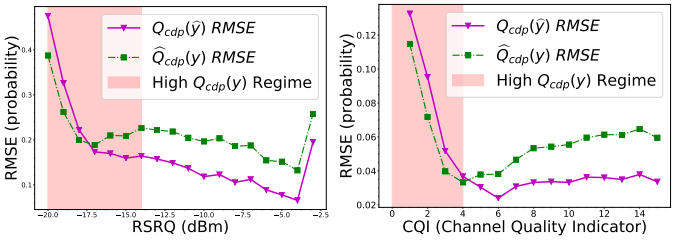
(a) $RMSE$ Relative Reduction.



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

Fig. 12: Improving the RSRP prediction itself via $P=(Q,k)\rightarrow P_B=(I,k)$ transform, for the Campus dataset. (a) $Q_{cdp}(y)$ $RMSE$ *vs.* RSRP. Our methodology reduces the error up to 27% for lower RSRP values (0-1 bars). (b) $Q^{-1}(\widehat{Q}_{cdp})$ $RMSE$ *vs.* Predicting directly RSRP values: the improvement has shifted towards the lower RSRP (fewer bars) accordingly to the new QoS function $Q_{cdp}(y)$ we trained for.

signal strength is high (*e.g.*, -50 to -60 dBm, see Fig. 3). The user experiences excellent QoS in that regime, and hence even moderately large errors in predicted RSRP would not greatly impact predictions of QoS. In contrast, 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. 3). For QoS



(a) NYC:Manhattan Uptown (b) LA: Suburb x470

Fig. 13: Call Drop Probability $Q_{cdp}(y)$ estimation for NYC and LA datasets. (a) RSRQ ($y=y^I$), Models Per cID. (b) CQI ($y=y^C$).

prediction, it can hence be worth trading greater RSRP error in the high-strength regime 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 QoS prediction, up to 32% for the values that matter more to cellular operators.

4.5 Results for Importance Sampling $P=(I,W)$

Next, we evaluate our framework in terms of the reweighted error ε_p. We predict RSRP and compare RFs vs. RFs$_w$ (*i.e.*, w_i are set to importance ratio as in Section 4.2).

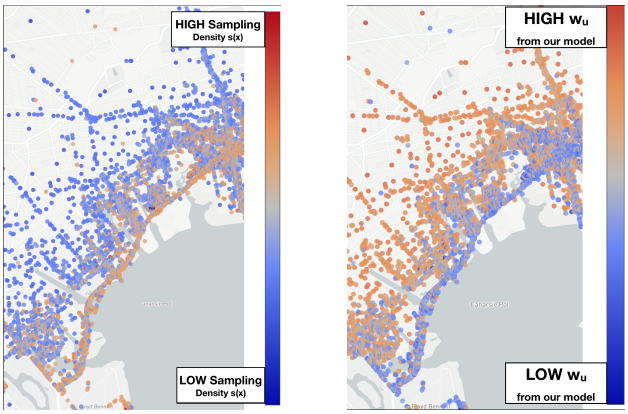
4.5.1 ε_u over Uniform Spatial Distribution, $P=(I,w_u)$.

(a) Campus dataset: 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 3.4. Table 9 reports the error ε_u for both the default RFs predictor and the RFs$_w$. We observe an improvement of up to 20% for ε_u for cells that are oversampled in just few locations; the average relative improvement is approx. 5%, which demonstrates the benefit of readjusting the training loss.

(b) NYC and LA datasets: Fig. 14(a) depicts the sampling distribution $s(\mathbf{l})$ in spatial dimensions (estimated by adaptive bandwidth KDE as we described in 3.4), 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 can speculate 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; it is a common practice to minimize the impact on

Cell Characteristics		Default $RFs \rightarrow \widehat{y}$	$RFs_{w_u} \rightarrow \widehat{y}_w$	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 9: Campus dataset, RSRP y prediction: ε_u Error (*i.e.,* reweighted according to the uniform distribution $\equiv P = (I, w_u)$): (i) Train on Default RFs *vs.* (ii) train on RFs$_w$ $w_i = w_u \propto \frac{1}{s(\mathbf{1})}$. Models per cID. For each cID and training case, we pick the best performing adaptive bandwidth KDE for estimating $s(\mathbf{1})$. Our methodology shows improvement up to approx. 20%. For cells * with extremely high sampling density in few locations -see [2]-, we utilize fixed bandwidth estimation both in space and time.



(a) Actual Measurements $s(\mathbf{1})$.

(b) Importance sampling.

Fig. 14: NYC dataset: RSRP prediction and uniform error ε_u. (a) Actual sampling ($s(\mathbf{1})$) estimated by an adaptive bandwidth KDE. (b) Reweighting with w_u from importance sampling. It can be 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.

users' devices [40]. Fig. 14(b) illustrates the importance ratio weights w_u and how our model readjusts for the sampling-target distribution mismatch. Similar patterns are observed throughout many different areas in NYC and LA datasets: *e.g.*, see the 405 highway in the Long Beach area x210.

Table 10 reports the error ε_u for different LTE TAs in NYC and LA datasets. The average performance improvement by training RFs$_{w_u}$ is approx. 3%, with up to 5% in some units. We also examined the area x532 where the benefit of our method was small and as expected the spatial distribution was indeed approx. uniform (omitted for space limitations). At the other extreme, regions with highly biased data collection (*i.e.,* x540 East NYC and x210 Long Beach in LA) show the highest error reduction (3.6% and 5.3% respectively). Overall, we find higher gains to reweighting on Campus dataset, as it is collected from a small number of users and hence more unevenly sampled. We expect that this feature will be common for small-scale

LTE-TA Characteristics				$RFs : \widehat{y}$	$RFs_{w_u} : \widehat{y}_w$	Improvement	
TAI	N	$\frac{N}{m^2}$	**Info**	$\sqrt{\varepsilon_u}$	$\sqrt{\varepsilon_u}$	**Diff.**	**Diff. (%)**
x210	197521	0.0005	Long Beach Lakewood	4.06	3.84	0.21	**5.3**
x552	97942	0.0011	Eastern Brooklyn	5.38	5.12	0.26	**4.9**
x540	136105	0.0009	E. Long Island	5.01	4.83	0.18	**3.6**
x535	121159	0.003	W. Queens	5.36	5.17	0.19	**3.6**
x641	10663	0.0001	Covina Hacienda Heights	1.8	1.74	0.06	**3.5**
x561	62448	0.0126	Manhattan Mid.	5.64	5.46	0.18	**3.2**
x470	198252	0.0009	LA Downtown Hollywood	4.56	4.43	0.13	**2.8**
x211	77049	0.0003	Suburban S. LA	4.06	3.96	0.1	**2.4**
x442	14538	0.0001	Manhattan Uptown Queens - Bronx	3.23	3.19	0.05	**1.5**
x537	37247	0.01	Manhattan Midtown East	7.62	7.53	0.09	**1.1**
x321	5111	0.0003	Eastern Brooklyn	3.87	3.83	0.04	**1.1**
x532	136508	0.004	Brooklyn	5.46	5.43	0.03	**0.5**
ALL	**1094543**	0.0042-0.014	**NYC & LA**	**4.88**	**4.72**	**0.16**	**3.16**

TABLE 10: NYC and LA datasets, RSRP y prediction: ε_u Error (*i.e.,* reweighted according to the uniform distribution $\equiv P = (I, w_u)$): (i) Train on Default RFs *vs.* (ii) train on RFs$_w$ $w_i = w_u \propto \frac{1}{s(\mathbf{1})}$. Models per LTE TA. We use adaptive bandwidth KDE for estimating $s(\mathbf{1})$ [36]. Our methodology shows improvement up to 5.3%.

data sets as well as setups with biased sampling because of mobile analytics companies practices, making reweighting especially important to correct for sampling bias.

4.5.2 ε_P : reweighting for Population Density $P = (I, w_d)$.

Weighing errors by local population density, instead of uniformly, results 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 $d(\mathbf{l}_i)$. Reweighted ε_d for RSRP data by using the weighted train RFs$_{w_d}$ *vs.* the the default RFs show an improvement of up to 5.7%.

Table 11 reports the error weighted proportional to the actual user population density over the same spatial area.

	Training Options	y domain $\rightarrow Q(\widehat{y})$		$Q(y)$ domain	
KPI: CQI All LTE-TA regions	$w_i = 1, \forall i$	$Q_{cdp}(\widehat{y})$	0.0107	$\widehat{Q}_{cdp}(y)$	0.0088
	$w_i = w_P \propto \frac{d(\mathbf{l}_i)}{s(\mathbf{l}_i)}$	$Q_{cdp}(\widehat{y}_w)$	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
	$w_i = w_d \propto \frac{d(\mathbf{l}_i)}{s(\mathbf{l}_i)}$	$Q_{cdp}(\widehat{y}_w)$	0.0047	$\widehat{Q}^w_{cdp}(y)$	0.0034
	Relative Difference		-5%		4%

TABLE 11: NYC and LA datasets. The error ε_d is re-weighted according to the population distribution) is computed on the Q domain, *i.e.,* $P = (Q_{cdp}, w_d)$. When predicting $\widehat{y}$ with weights and then converting to $Q(y)$, information is lost from the transformation. When training with the importance sampling weights, then predicting $\widehat{Q}(y)$, can further reduce the error up to 5%.

Table 12 includes the reweighted ε_d for RSRP data by using the default RFs vs. the weighted train RFs$_{w_d}$; we see performance improvement up to 5.7

4.6 Reweighted Error for QoS functions: $P = (Q, W)$

So far, we have separately evaluated the improvement from (1) predicting QoS directly and (2) re-weighting by importance ratio. Here, we combine the two and calculate the reweighted error ε_p (how we handle the input space) for a QoS function (how we handle the output space) of interest. Due to lack of space, we only show results for Q_{cdp}(y). In Table 3, we show four cases to be compared. First, $Q_{cdp}(\widehat{y})$

LTE-TA Characteristics				$RFs \rightarrow \widehat{y}$	$RFs_{w_P} \rightarrow \widehat{y}_w$	Improvement	
TAI	N	**Info**	$\frac{N}{m^2}$	Default $\sqrt{\varepsilon_P}$	$\sqrt{\varepsilon_P}$	**Diff.**	**Diff. (%)**
x561	63303	Manhattan Midtown	0.0126	7.23	6.82	0.41	**5.7**
x321	7014	Eastern Brooklyn	0.0003	4.94	4.8	0.14	**2.8**
x535	122071	W. Queens	0.003	6.03	5.87	0.15	**2.5**
x552	98240	Eastern Brooklyn	0.0011	5.35	5.29	0.06	**1.2**
x532	137962	Brooklyn	0.004	6.24	6.22	0.02	**0.3**
x537	37964	Manhattan Midtown E.	0.01	8.82	8.81	0.01	**0.1**
x540	138495	E. Long Island	0.0009	5.09	5.09	0.00	**0.0**
x442	16372	Manhattan Uptown Queens - Bronx	0.0001	3.97	4.21	-0.24	**-6.1**
ALL	**621421**	**NYC**	0.0003-0.0126	**5.98**	**5.90**	**0.08**	**1.35**

TABLE 12: `NYC and LA datasets` RSRP y prediction: ε_d Error (*i.e.,* reweighted according to the population distribution): (i) Train on Default `RFs` *vs.* (ii) train on $\texttt{RFs}_w$ $w_i = w_d \propto \frac{d(\mathbf{l})}{s(\mathbf{l})}$. Models per LTE TA. We use adaptive bandwidth KDE for estimating $s(\mathbf{l})$ [36]. Our methodology shows improvement up to 5.7%.

	Training Options	y domain $\rightarrow Q(\widehat{y})$		$Q(y)$ 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%

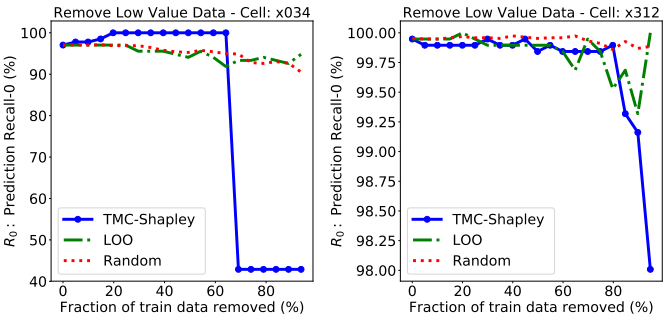
TABLE 13: $P = (Q_{cdp}, w_u)$, `NYC and LA datasets`, error ε_u (*i.e.,* reweighted according to the uniform distribution), results on the Q domain. Predicting $\widehat{y}$ with weights and then converting to $Q(y)$ does not help because information is lost from the transformation. Predicting $\widehat{Q}(y)$ after training with the importance sampling weights further reduces error up to 5%.

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 to 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 (*i.e.*, $P = (Q_{cdp}, W)$).

Table 13 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 have very little or even negative effect for mapping back to CDP. Similar results are observed for error proportional to user population density, but omitted due to lack of space. This demonstrates the importance of choosing the loss function, jointly controlled by w and Q, to optimize performance for a specific prediction problem.

4.7 Data Shapley Results

In this section, we compute the Data Shapley values (using the techniques presented in Section 3.5 and Appendix A2) of datapoints in our datasets. For each prediction problem $P = (Q, W)$ and dataset of interest, we compute the performance score V, and eventually the shapley value ϕ_i value of every datum i in that training dataset. We then order all datums in decreasing shapley values, and remove datapoints with negative or low values. First, we show that by removing measurements with negative Shapley values, we can actually *improve the prediction* performance up to 30%. Second, we show that we can further remove a large percentage of data points (up to 65%, depending on the dataset and problem P) with low Shapley values, while maintaining high prediction performance. The latter allows for "data minimization", *not before* but *after* the data collection phase, in the following sense: we can store the minimum amount of data needed to train a good prediction model. This reduces the cost of storing such training datasets, informs pricing when these datasets are bought, and is also a good practice from a privacy perspective.



(a) Cell x034. (b) Cell x312.

Fig. 15: `Campus dataset`: Removing low valued data points (for Data-Shapley, LOO and Random) affects Recall R_0.

Data Minimization Setup: As we described in Sec. 3.5, 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 indicator classification $\widehat{Q}_c(y)$, *i.e.,* whether there is coverage in a location or not (as per Sec. 4.4). We remove batches of 5% of $\mathcal{D}_{train}$ starting from the least valuable (*i.e.,* lowest ϕ_i). At each step, we re-train the $\texttt{RFs}_{all}$ model with the remaining $\mathcal{D}_{train}$ and calculate the performance of the prediction on the $\mathcal{D}_{\text{held-out}}$ data.

Baselines for Comparison: To compare against data Shapley valuation, we consider two baselines. First, Leave-one-out (LOO) [18], calculates the datum's value by leaving it out and calculating the score V, *i.e.*, $\phi_i^{\text{LOO}} = V(D) - V(D - \{i\})$. Second, we remove randomly batches of $\mathcal{D}_{train}$ at each step.

4.7.1 *Effect of removing low value data on Recall.*

For the `Campus dataset`, Figures 15(a), 15(b) and 16(a) present results for three representative cells (c034, x312, x901), in terms of the recall R_0 (see Sec. 4.4 for its utility for operators), as a function of the percentage of data removed from $\mathcal{D}_{train}$, by all discussed methods (TMC-Shapley, LOO and Random). TMC-Shapley's performance either improves or remains the same when we remove low value data points compared to LOO and Random. There are two plausible 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 datums 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. Interestingly, after a certain threshold, TMC-Shapley's performance drops with the removal of just a single batch, which holds significant predictive power. In contrast, by removing data randomly we keep both bad and quality data, which explains why Random's performance neither improves nor decays fast.

% Train 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 14: x901 Cell: Data Minimization Results per removal step.

Let us discuss in more depth Fig.16(a), which presents data minimization results for cell x901 of the `Campus dataset`. The removal of low valued data according to TMC-Shapley's valuation improves recall R_0 compared to Random and LOO. Moreover, in Fig. 16(a) we annotate with the label "A" the beginning of the process (*i.e.,* still using the entire $\mathcal{D}_{train}$); label "B" indicates the step where 65% of the data have been removed and R_0 has achieved its maximum value. Fig. 16 depicts the ϕ_i CDF values for all $\mathcal{D}_{train}$ (*i.e.,* label "A").

Table 14 reports additional detail for data minimization in cell x901, in Fig. 16(a), including: the removed fraction and number of training data, the recall R_0, number of measurements per 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. We make the following observations. 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. Second, when R_0 drops from 99% to 33% there is still data availability for both classes/users.

Dataset Shift and Data Shapley, for cell x901. The dataset shift problem [17] (*i.e.,* the mismatch of the training and the target distribution) for the labels "A" *vs.* "C" offers also significant insights of how the final performance is affected after a certain threshold of removing training data. Fig. 17(a) shows $w_u \propto \frac{1}{s(\mathbf{1})}$ 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 $\mathbb{E}[\log s(\mathbf{1}) = -3.3]$. On the contrary, Fig. 17(b) depicts $w_u \propto \frac{1}{s(\mathbf{1})}$ 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 decreased to $\mathbb{E}[\log s(\mathbf{1}) = -9.3]$. The held-out data were randomly sampled from the original distribution, therefore, there is now a mismatch between the original and target distribution (in other words, the dataset shift problem we studied in Sec 3.4) which can explain the drop in the performance.

4.7.2 Effect of Data Minimization on Metrics beyond R_0

We also considered cell x034 and the coverage classification task, but for a different performance metric (V): accuracy (A). Fig. 18(a) reports results from the same data removal process as previously (*i.e.,* remove an increasing percentage of lowest valued datapoints). 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 certain threshold, the performance of TMC-Shap drops, as happened with the recall for the same cell (Fig. 18(b)). That is expected because the portion ofdata that can be removed depends on the dataset and the performance score, even for the same predictor.

5 Conclusion and Discussion

We presented a general framework for predicting cellular performance from available measurements, which gives knobs to operators to express what they care most, *i.e., what* performance metrics, in *what* regimes (via quality functions Q), and in *what locations* (via weights W). To that end, (1) we trained directly on the Q instead of the RSRP domain; (2) we used the importance ratio re-weighting to address the mismatch between target and sampling distributions; and (3) we applied the data Shapley framework to assess the value ϕ of available measurements for the particular prediction task $P = (Q, W)$, which in turn enables data cleaning and minimization. We evaluated these ideas on large, real-world LTE datasets and demonstrated their benefits.

Applications to 5G Deployment. First, our framework can naturally handle prediction over mmWave 5G with small cells, similarly to what we did in the Midtown Manhattan `NYC dataset`. There, many overlapping small cells (100s in the same LTE TA) are used to cover the dense urban environment. We performed prediction per LTE TA(not per cell) and cID was used as a feature, to learn the radio environment. Second, cellular operators have aggressively pushed the sub-6GHz deployments due to the mmWave limitations [43] (*e.g.,* range of approx. 100s meters, only line-of-sight). Sub-6Ghz deployments share the same physical layer and network characteristics with the LTE networks. Third, sampling biases will be amplified with small cells deployment and our $W(\mathbf{x})$ re-weighting schema could help mitigate it, by expressing complex 5G operator objectives.

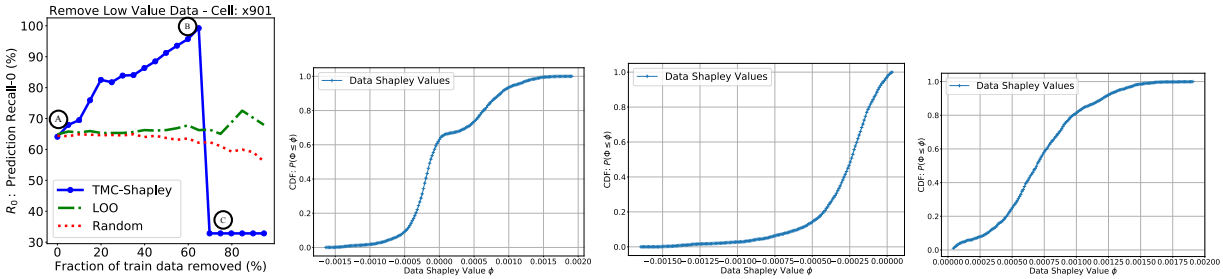
Acknowledgments

This work was supported by NSF Awards 1956393, 1956393, 1939237, 1900654, 1901488, 1649372, and 1526736. We are grateful to Tutela for sharing their city-wide datasets. We thank former group members who participated in the UCI study; Justin Ley and Mengwei Yang for discussions on DNNs and data shapley values, respectively.

References

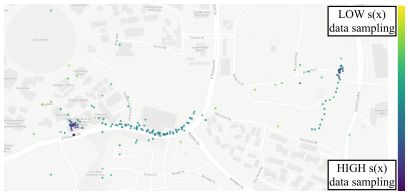
[1] J. Yang, A. Varshavsky, H. Liu, Y. Chen, and M. Gruteser, "Accuracy characterization of cell tower localization," in *Proc. of the ACM UbiComp '10*, 2010, pp. 223–226.

[2] E. Alimpertis, A. Markopoulou, C. Butts, and K. Psounis, "City-wide signal strength maps: Prediction with random forests," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2536–2542. [Online]. Available: https://doi.org/10.1145/3308558.3313726

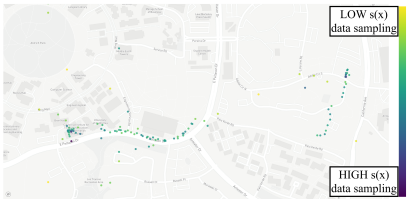


(a) Removing % Data. (b) Label A: all data in cell. (c) data between Labels A and B. (d) Data at Label B.

Fig. 16: **Data minimization for** Campus **Cell x901. (a)** Removing % datapoints using different techniques: data Shapley, Leave-One-Out, Random. As we start removing data with low data Shapley (from A to B), recall improves. If we remove more data beyond a certain point (B), recall sharply decreases (from B to C). **(b)(c)(d)** CDFs of Data Shapley values ϕ_i fr the points among labels A,B,C in (a).

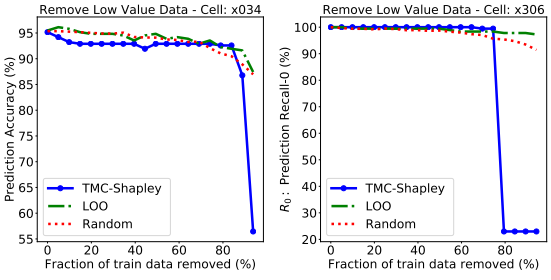


(a) Label A, Fig 16(a).



(b) Remaining Data's $s\mathbf{x}$ for performance at Label C, Fig 16(a).

Fig. 17: Campus dataset cell x901. **Top**: Initial Sampling distribution $s(\mathbf{x})$ (Data for Label **A** in Fig. 16(a)). $\mathbb{E}[\log s(\mathbf{x}) = -3.3]$ **Bottom**: Final Sampling distribution $s(\mathbf{x})$ (Data for Label **C** in Fig. 16(a)). The procedure of removing data points it eventually changed the sampling distribution of the data; at label A two regions were largely over-sampled; at label C when the performance has finally been decreased the sampling distribution of the data look more uniform therefore it mismatches the original train distribution. $\mathbb{E}[\log s(\mathbf{x}) = -9.3]$



(a) Accuracy A. (b) Recall R_0.

Fig. 18: Campus dataset: Remove low valued data points (for Data-Shapley, LOO and Random) for cell x034; **Left (a)** Results for Accuracy, **Right (b)** Results for Recall R_0 for coverage loss.

[3] Open Signal Inc., "Network Experience Insights,," https://www.opensignal.com, last accessed on Nov 29 2022.

[4] Tutela Inc., "Crowdsourced mobile data," http://www.tutela.com, Jun. 2011.

[5] J. Garcia-Reinoso *et al.*, "The 5g eve multi-site experimental architecture and experimentation workflow," in *IEEE 2nd 5G World Forum*, Sep. 2019, pp. 335–340.

[6] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: How to empower SON with big data for enabling 5G," *IEEE network*, vol. 28, no. 6, pp. 27–33, 2014.

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

[8] Y. J. Bultitude and T. Rautiainen, "WINNER II Channel Models," Technical Report , IST-4-027756 WINNER II, D1. 1.2 V1. 2,, Tech. Rep., Sep. 2007.

[9] 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.

[10] 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*.

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

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

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

[14] 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.

[15] 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*, May 2017, pp. 1–9.

[16] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proc. of the 18th ACM MobiHoc*, ser. Mobihoc '18. ACM, 2018, pp. 231–240.

[17] J. Snoek and et al., "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in NeurIPS*, 2019, pp. 13 991–14 002.

[18] A. Ghorbani and J. Zou, "Data Shapley: Equitable Valuation of Data for Machine Learning," 2019. [Online]. Available: http://arxiv.org/abs/1904.02868

[19] E. Alimpertis, N. Fasarakis-Hilliard, and A. Bletsas, "Community rf sensing for source localization," *IEEE Wireless Comm. Letters*, vol. 3, no. 4, pp. 393–396, 2014.

[20] 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*. ACM, 2018, pp. 221–230.

[21] 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*, 2017, pp. 1–9.

[22] B. F. D. Hähnel and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proc. of Robotics: Science and Systems*. MIT Press, Aug. 2006.

[23] E. Krijestorac, S. Hanna, and D. Cabric, "Spatial signal strength prediction using 3d maps and deep learning," *arXiv preprint arXiv:2011.03597*, 2020.

[24] Y. Teganya and D. Romero, "Data-driven spectrum cartography via deep completion autoencoders," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

[25] V. Adarsh, M. Nekrasov, U. Paul, A. Ermakov, A. Gupta, M. Vigil-Hayes, E. Zegura, and E. Belding, "Too late for playback: Esti-

mation of video stream quality in rural and urban contexts," in *Proc. of the International Conference on Passive and Active Network Measurement*. Springer, 2021, pp. 141–157.

[26] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 197–210. [Online]. Available: https://doi.org/10.1145/3098822.3098843

[27] A. e. a. Narayanan, "Lumos5g: Mapping and predicting commercial mmwave 5g throughput," in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 176–193. [Online]. Available: https://doi.org/10.1145/3419394.3423629

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

[29] S. Boukoros, M. Humbert, S. Katzenbeisser, and C. Troncoso, "On (the lack of) location privacy in crowdsourcing applications," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 1859–1876.

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

[31] 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)*. ACM, Aug. 2015, pp. 33–38.

[32] L. Breiman, "Random forests"," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[33] 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*, 2015, pp. 452–463.

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

[35] 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*, 2013, pp. 37–42.

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

[37] C. Chen, A. Liaw, L. Breiman *et al.*, "Using random forest to learn imbalanced data," Technical Report TR-666, vol. 110, no. 1-12, p. 24, University of California, Berkeley, https://statistics.berkeley.edu/tech-reports/666, Jul. 2004.

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

[39] Z. Li, A. Nika, X. Zhang, Y. Zhu, Y. Yao, B. Y. Zhao, and H. Zheng, "Identifying value in crowdsourced wireless signal measurements," in *Proc. of the ACM WWW '17*. ACM, May 2017, pp. 607–616.

[40] E. Alimpertis and A. Markopoulou, "A system for crowdsourcing passive mobile network measurements," in *Poster in 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Mar. 2017.

[41] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct., pp. 2825–2830, 2011.

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

[43] J. Brodkin, "The Limits OF 5G: Millimeter-wave 5G will never scale beyond dense urban areas, T-Mobile says T-Mobile CTO says 5G's high-frequency spectrum won't cover rural America." https://arstechnica.com/information-technology/2019/04/millimeter-wave-5g-will-never-scale-beyond-dense-urban-areas-t-mobile-says/, accessed in Mar 2021, Apr. 2019.

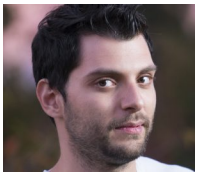
[44] E. Alimpertis, "Mobile Coverage Maps Prediction," PhD Thesis, Donald Bren School of Information and Computer Sciences, University of California Irvine, Irvine, CA, USA, 2020.

[45] COST 231 Report, "Digital mobile radio towards future generation systems," http://www.lx.it.pt/cost231/, Tech. Rep., 1999.

[46] O'Malley et al., "Keras tuner," https://github.com/keras-team/keras-tuner, 2019.

[47] L. L. et al., "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018.

[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010, pp. 249–256.



Emmanouil Alimpertis (Member, IEEE) received his M.Sc. and Diploma degrees in Electronic and Computer Engineering at the Technical University of Crete, Greece, in 2012 and 2014 respectively. In 2020, he received his Ph.D. degree in the Networked System Program at the University of California, Irvine. He is with Apple Inc. at Cupertino, CA. His research interests include network measurements, machine learning for mobile systems, and location estimation.

Athina Markopoulou (S'98-M'02-SM'13-F'21) is a Professor in EECS Department at UC Irvine. She received the Diploma degree in Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 1996, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 1998 and 2003, respectively. She had short-term positions at SprintLabs, Arista Networks, and IT University of Copenhagen. She has received the NSF CAREER award in 2008, the HSSoE Faculty Midcareer Research Award in 2014, and a UCI Chancellor's Fellowship 2019-22. She has served as an Associate Editor for IEEE/ACM Trans. on Networking and for ACM Computer Communications Review, as the General Co-Chair for ACM CoNEXT 2016, as TPC Co-Chair for ACM SIGMETRICS 2020 and NetCod 2012. She is a Distinguished Member of the ACM. Her interests include networking, privacy, network coding.

Carter T. Butts (Member, IEEE) is a Professor in the departments of Sociology, Statistics, and Electrical Engineering and Computer Science (EECS) and the Institute for Mathematical Behavioral Sciences at the University of California, Irvine. His research involves the application of mathematical and computational techniques to theoretical and methodological problems within the areas of social network analysis, mathematical sociology, quantitative methodology, and human judgment and decision making. His work has appeared in a range of journals, including Science, Sociological Methodology, the Journal of Mathematical Sociology, Social Networks, and Computational and Mathematical Organization Theory.

Evita Bakopoulou received her B.Sc. and M.Sc. degrees in Computer Science from Athens University of Economics and Business, Greece, in 2014 and 2016, respectively. She received a Ph.D. in Networked Systems from UC Irvine in 2021, and joined Privacy Engineering at Google in 2022. She was a Summer Intern with Bell Labs (2017), Oath/Verizon Digital Media Services (2018) and Google (2020). Her research interests are primarily in the area of machine learning and privacy.

Konstantinos Psounis (Fellow, IEEE) received the BS degree from the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece, in 1997, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University, Stanford, CA, USA, in 1999 and 2002, respectively. He is currently a Professor of Electrical and Computer Engineering and of Computer Science with the University of Southern California. He works on modeling, performance analysis, algorithm design, and system implementation for efficient and privacy-preserving networked, distributed systems. He is a Distinguished Member of the ACM.