Unsatisfied Today, Satisfied Tomorrow: a simulation framework for performance evaluation of crowdsourcing-based network monitoring

Andrea Pimpinella, Marianna Repossi, Alessandro E. C. Redondi
Dipartimento di Elettronica, Informazione e Bioingegneria,
Politecnico di Milano, Italy
e-mail: {andrea.pimpinella, marianna.repossi,
alessandroenrico.redondi}@polimi.it

Abstract

Network operators need to continuously upgrade their infrastructures in order to keep their customer satisfaction levels high. Crowdsourcing-based approaches are generally adopted, where customers are directly asked to answer surveys about their experience. Since the number of collaborative users is generally low, network operators rely on Machine Learning models to predict the satisfaction levels/QoE of the users rather than directly measuring it through surveys. Finally, combining the true/predicted users satisfaction labels with information on each user mobility (e.g, which network sites each user has visited and for how long), an operator may reveal critical areas in the network and drive/prioritize investments properly. In this work, we propose an empirical framework tailored to assess the quality of the detection of under-performing cells starting from subjective user experience grades. The framework allows to simulate diverse networking scenarios, where a network characterized by a small set of under-performing cells is visited by heterogeneous users moving through it according to realistic mobility models. The framework simulates both the processes of satisfaction surveys delivery and users satisfaction prediction, considering different delivery strategies and evaluating prediction algorithms characterized by different prediction performance. We use the simulation framework to test empirically the performance of under-performing sites detection in general scenarios characterized by different users density and mobility models to obtain insights which are generalizable and that provide interesting guidelines for network operators.

Preprint submitted to Computer Communications *October 10, 2021*

Keywords: QoE, Crowdsourcing, Network Monitoring

1. Introduction

According to recent Cisco estimates [1], by 2021 mobile cellular networks will connect more than 11 billion mobile devices and will be responsible for more than one fifth of the total IP traffic generated worldwide. Moreover, the global average broadband speed will more than double from 2018 to 2023, from 45.9 Mbps to 110.4 Mbps. This will result in increased utilisation of high-bandwidth demanding applications, such as on-demand 4K video streaming, cloud storage, etc. To face this unprecedented growth in both volume of mobile traffic and data rate needs of customers, network operators continuously invest in all network domains, including but not limited to spectrum, radio access network (RAN) infrastructure, transmission and core networks. The final goal of such investments is to generate profit by (i) attracting as many customers as possible and (ii) minimizing the number of churners, i.e., users who stop their current subscriptions and move to a different operator.

Concerning the latter point, a well established process mobile operators perform to avoid churns is to monitor their customers satisfaction levels through directed surveys: as an example, the Net Promoter Score (NPS) survey asks users to indicate the likelihood of recommending the network operator to a friend or colleague on a scale from 0 to 10. In addition to such a generic survey, operators often ask customers to reply very specific questions related to the user satisfaction or Quality of Experience (QoE) relative to certain network services (network coverage, voice and video quality, etc.), which can better highlight possible problems in the network, such as under-perfoming or malfunctioning network cells/sites. In fact, recognizing faults in a given network cell/site looking solely at objective QoS measurements when there is no clear degradation of performance is a complex challenge, considering that such a strategy forces an operator to work in a unsupervised fashion where there is no ground truth about which network element is truly under-performing. As an example, regarding video streaming applications, low throughput does not always interrupt viewers' watching experiences [2], meaning that QoS based metrics could fail to capture the reasons for users dissatisfaction. Indeed, it is often the case that a mobile operator has no clear evidence of a fault at a network site (i.e., the operator is not able to recognize it directly from objective measurements) but yet the visitors of the site are not satisfied about one or more network services. Therefore, tracking users satisfaction to get objective insights about the performance of network cells/sites represents a useful tool for improving the effectiveness of network monitoring processes. Unfortunately, such a direct way to track users satisfaction is costly and cumbersome for operators, mainly due to the generic poor cooperative attitude of customers. Moreover, the problem of the reliability of users' replies to such surveys is subject to intense investigations [3, 4, 5, 6]: regardless of the subject of the surveys, studies confirm that it is not a trivial task to gather reliable responses from crowds, especially when no reward systems are conceived.

To cope with these issues, several studies in the recent literature addressed the problem of predicting the satisfaction level of customers, rather than directly measuring it through surveys [7, 8, 9, 10, 11]. Following the renovated interest in big data, machine learning and artificial intelligence, the goal of such works is to identify the set of unsatisfied customers starting from a large variety of objective features, both operative (e.g., average throughput and signal quality) and business-related (e.g., gender, age or tariff plan). Such features, and the corresponding ground-truth satisfaction levels, are generally used to train machine-learning models and eventually used to estimate the satisfaction levels/QoE of a much larger population. Finally, combining the true/predicted users satisfaction levels with information on each user mobility (e.g, which network sites each user has visited and for how much time), an operator may reveal critical areas in the network and drive/prioritize investments properly.

However, the detection of under-performing cells starting from true/predicted subjective grades has its own issues. First, users are heterogeneous and their perception of network quality is highly subjective. Second, when a negative satisfaction expressed by a user refers to a long period of time (e.g., one month), it is difficult to identify which of the network sites visited during that period is the most responsible. Third, in case the user satisfaction level is estimated through a machine learning algorithm, a prediction error is likely to be expected. Therefore, in this complex scenario, an operator may argue about the validity/quality of the detected under-performing cells. To solve these issues, we propose an empirical framework tailored to assess the quality of the detection of under-performing cells starting from subjective users grades. In details, the contributions of this paper are:

1. We build a framework that allows to simulate a network composed of a (small) set of under-performing/malfunctioning cells, with heterogeneous users moving freely in it according to realistic mobility models. Depending on each user mobility and subjective profile, the framework allows to obtain each user's (true) satisfaction level.

2. The framework also simulates the process of satisfaction surveys delivery performed by the operator, which is able to sample only a subset of the true users satisfaction levels through surveys. We consider two different delivery strategies: a completely random one and one which maximizes the number of covered network sites.

3. Moreover, the proposed framework allows to simulate the process of users satisfaction prediction using a machine learning algorithm whose performance can be changed at will. This allows to quantify the impact of prediction errors on the detection process, and to understand what are the minimum performance a prediction model for user satisfaction should possess to be applied in the overall methodology.

4. Finally, we test empirically the simulation framework with different users density and mobility models to obtain insights which are generalizable.

The remainder of this article is organized as follows: Section 2 describes a general crowdsourcing network monitoring process than can be adopted by an operator to perform detection of under-performing sites, leveraging both objective data and true/predicted user satisfaction levels; Section 3 describes the simulation framework that can be used to assess the quality of the detection process. Diverse scenarios characterized by different users densities, mobility models and surveys delivery strategies are simulated in Section 4, to empirically test the performance of the detection of under-performing cellular sites. Section 5 reviews the relevant literature on QoE prediction and QoE-based issues detection in cellular networks. Finally, Section 6 summarises remarks and conclusions.

2. Under-Performing Sites Detection Process

Detecting possible issues in an operator network infrastructure using information about the perceived user experience is a process known under the name of crowdsourcing network monitoring, a field which has received increasing attention in the last few years [12, 13, 14]. According to this approach, the mobile operator administers to its customers population $\mathcal{U}$, $|\mathcal{U}| = N$, a set of user experience/satisfaction surveys (either directly or through the help of proper apps installed on the users equipments), whose answers may help to reveal critical/under-performing network sites, hence steering investments in the right directions (e.g., increasing the bandwidth or the output power available at specific base stations).

Rather than detecting all sites responsible for users dissatisfaction, a more convenient output for a mobile operator consists in a *site ranking*, i.e., a sorted list of network sites in which the ones responsible for the highest number of unsatisfied users appear at the top positions. In such a way, an operator may allocate the available budget for investing into the first k sites in the list in a prioritized fashion.

When the responses gathered from the users are few (and this is often the case [11]), operators may rely on data science techniques to predict the satisfaction of additional users, artificially enlarging the set of available responses. This is generally obtained by exploiting pre-trained machine learning models that correlate objective network measurements collected from the users (e.g., throughput, channel quality, amount of time spent with limited service) with the users perceived satisfaction. Since the objective network measurements are generally available for a much larger amount of users compared to the (subjective) satisfaction responses, this strategy allows to greatly enlarge the knowledge base usable for detecting or ranking under-performing sites. The general process is illustrated in Figure 1: let $\mathcal{U} = \{\mathcal{U}_{\text{a}} \cup \mathcal{U}_{\text{na}}\}$ be the total set of network users, composed of customers whose survey response is available ($\mathcal{U}_{\text{a}}$) or not ($\mathcal{U}_{\text{na}}$). Similarly, let $\mathbf{X}_{\text{a}}$ and $\mathbf{X}_{\text{na}}$ be the set of objective network measurements for the two sets of users. A machine learning model $f(\cdot)$, trained and possibly updated with the knowledge coming from users whose answers $\mathbf{s}_{\text{gt}}$ are available, can be used to predict the satisfactions $\hat{\mathbf{s}}$ of non-answering users. We underline that the model $f(\cdot)$ can be trained independently of the detection process and updated any time new surveys responses are gathered by the operator. Finally, the objective network measurements ($\mathbf{X}_{\text{a}}$, $\mathbf{X}_{\text{na}}$) and the true and predicted users satisfaction ($\mathbf{s}_{\text{gt}}$ and $\hat{\mathbf{s}}$) are leveraged to produce a ranked list of sites in the network, which we refer to as $\hat{\mathcal{J}}_{\text{u}}$.

Several important questions related to such an approach can be raised by a mobile network operator:

Q.1 *Ranking strategy*: Assuming the availability of the (true) satisfaction of the entire set of users, how can under-performing sites be ranked/detected?

Q.2 *User heterogeneity:* Different users react to network issues in different ways. How does such heterogeneity impact on the detection of under-performing sites?

Q.3 *Prediction errors:* When a ML model $f(\cdot)$ is used to predict the satisfaction of the non-answering users, a prediction error is generally expected. How

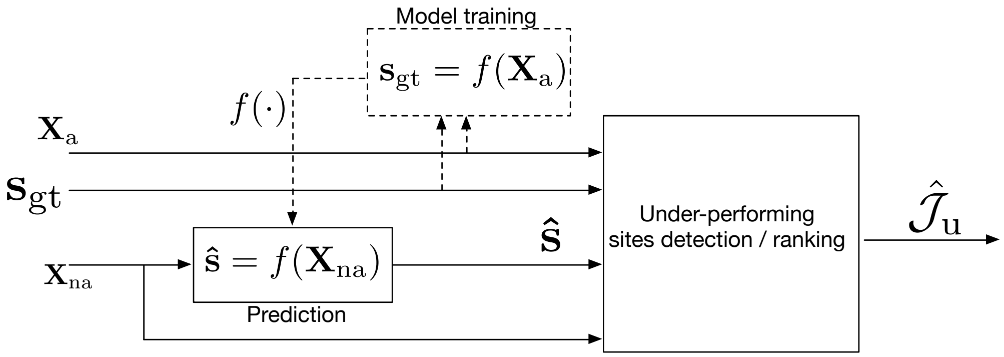

Model training
$\mathbf{s}_{\mathrm{gt}} = f(\mathbf{X}_{\mathrm{a}})$
$f(\cdot)$
$\mathbf{X}_{\mathrm{a}}$
$\mathbf{s}_{\mathrm{gt}}$
$\mathbf{X}_{\mathrm{na}}$
$\hat{\mathbf{s}} = f(\mathbf{X}_{\mathrm{na}})$
Prediction
$\hat{\mathbf{s}}$
Under-performing
sites detection / ranking
$\hat{\mathcal{J}}_{\mathrm{u}}$

Figure 1: General process for crowdsourcing-based sites ranking. The satisfaction grades from the users, true or predicted, are combined with objective information (such as user visit times) to detect critical network sites and rank them according to their impact on users satisfaction. Dashed lines refer to the fact that the model $f(\cdot)$ is independent from the detection process and may be updated asynchronously whenever new survey responses are gathered by the operator.

does such an error impact on the ranking/detection of under-performing sites?

Q.4 *Users density:* What is the relationship among the cardinality of the sets of answering and non-answering users, the number of sites in the network and the performance of the ranking/detection operation?

Q.5 *Survey delivery:* If only a subset of users is expected to answer the satisfaction surveys, is there a way to select such a subset in order to increase the performance of the detection process?

In the following, we describe a simulation framework that an operator can leverage in order to find answers to such questions.

3. Simulation Framework

In order to answer to questions Q.1-Q.5, we propose a simulation framework composed of several building blocks, illustrated in Figure 2. The following Sections provide details on each component of the framework.

3.1. Topology Generator (TG)

The TG is responsible of generating mobile network instances composed of a set of network sites $\mathcal{J}, |\mathcal{J}| = M$, deployed in a realistic scenario (e.g., urban or

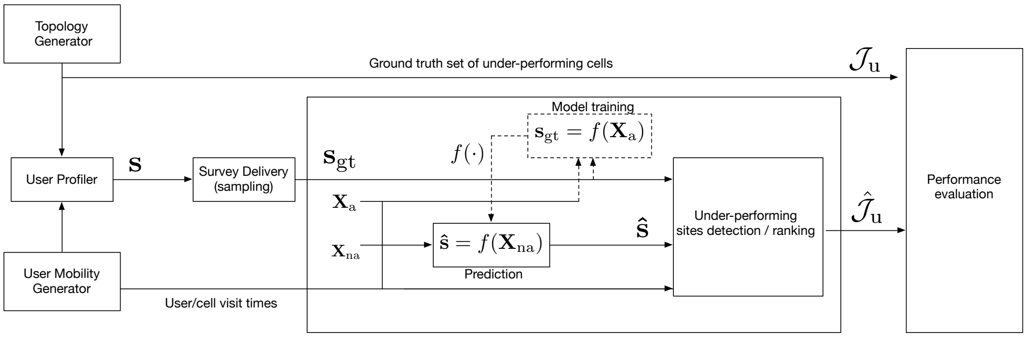

Topology Generator
Ground truth set of under-performing cells
$\mathcal{J}_{\mathrm{u}}$
Performance evaluation
Model training
$\mathbf{s}_{\mathrm{gt}} = f(\mathbf{X}_{\mathrm{a}})$
$f(\cdot)$
User Profiler
$\mathbf{S}$
Survey Delivery (sampling)
$\mathbf{s}_{\mathrm{gt}}$
$\mathbf{X}_{\mathrm{a}}$
$\mathbf{X}_{\mathrm{na}}$
$\hat{\mathbf{s}} = f(\mathbf{X}_{\mathrm{na}})$
Prediction
$\hat{\mathbf{S}}$
Under-performing sites detection / ranking
$\hat{\mathcal{J}}_{\mathrm{u}}$
User Mobility Generator
User/cell visit times

Figure 2: Architecture of a simulation framework to test the anomaly detection system.

rural). The TG also defines which sites $\mathcal{J}_{\mathrm{u}} \subset \mathcal{J}, |\mathcal{J}_{\mathrm{u}}| = \Omega < M$, are malfunctioning or under-performing in a given network topology. In particular, considering that a network site is composed by different but co-located network elements which mostly work in a shared fashion to deliver diverse network services to the end users (e.g., the base-band unit processes information coming from different antennas working a different RF bands), in our work the TG does not associate the failure to one or the other specific network element but rather it marks the overall network site as under-performing. The selection of the under-performing sites is performed according to a random process, specified in input. In this work we consider a uniform distribution (i.e., all network sites have the same probability of being under-performing), assuming that the degradation of the performance of a under-performing site is due to a random failure. Note that in general an operator may use any other input distribution, e.g. to simulate the case in which it has a prior regarding what most likely has caused the degradation of performance in its own network. As an example, if an operator suspects that congestion is the main cause of the degradation of its customers experience, then it can tune the TG such as sites characterized by a higher level of congestion (e.g., visited by large number of users) are selected as under-performing with a higher probability. We recall that both the total number of sites M and the number of malfunctioning sites Ω are input parameters of the simulation framework. To conclude, we observe that the TG can be easily generalized to support the case where network sites are composed of different sectors (multiple antennas) or frequency layers that can independently be subject to failure: this requires the operator to specify the mapping between physical location and most likely serving cell, an operation which is generally performed through the use of coverage maps.

3.2. User Mobility Manager (UMM)

The UMM models the mobility of the population of users $\mathcal{U}$ through the cellular network simulated by the TG. In particular, the UMM leverages a human mobility model which defines for the i-th user i) which network sites are visited and ii) for how long. Several models are available in the literature to simulate the statistical properties of human mobility [15, 16, 17, 18]. In this work we consider the model proposed in [18], which is based on the following observations: i) humans have a periodic tendency to return to previously visited places, ii) humans spend most of their time in a few number of locations and iii) the distributions of the time spent by a user in a location $P(\Delta t)$ and the distance covered between two sightings $P(\Delta r)$ are fat-tailed, i.e. $P(\Delta r) \sim |\Delta r|^{-1-\alpha}$ and $P(\Delta t) \sim |\Delta t|^{-1-\beta}$. In details, the mobility model implemented in the UMM works according to different steps, as illustrated in Figure 3:

- *Initialization:* let S_i be an integer variable which counts the number of distinct locations visited by the i-th user, initially set to 1. At startup, each user is associated to one site in the network topology, chosen at random. Then, each user waits for a random period of time Δt and eventually decides whether to explore a new location (Exploration step) or to return to an already visited site, including the current one (Preferential Return step).

- *Exploration:* with probability $P_{new} = \rho S_i^{-\gamma}$, the user jumps in a random direction θ, uniformly distributed in the range $[0, 2\pi)$ and with a random jump length Δr. The closest site to the landing location will be visited by the user. As the user moves to this new position, the number of previously visited locations increases from S_i to $S_i + 1$.

- *Preferential Return:* with probability $1 - P_{new}$, the user returns to a previously visited location with a probability proportional to the number of visits the user previously had to that location.

These steps are repeated independently for each user: at each iteration the UMM updates the vector $\mathbf{t}_i \in \mathbb{R}^M_{\geq 0}$ whose entries $t_{i,j}$ correspond to the visit times of the i-th user in the j-th network site. The process ends when the total visit time for each user is equal to the simulation time horizon T, i.e., when $\sum_j t_{i,j} = T, \ \forall i$. The parameters controlling the user's tendency of exploring a new place ρ and γ, as well as the fat-tail distribution parameters for the jump sizes α and the waiting times β can be modified according to the specific case under consideration. We detail the choice of such hyper-parameters in Section 4.

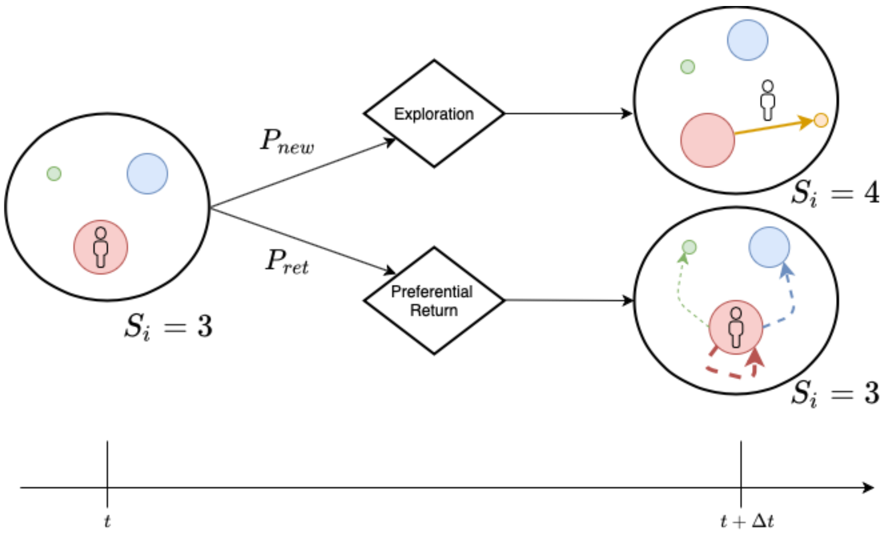


Figure 3: Considering a generic user i, S equals the cardinality of the set of visited places, circles stems for the sites already visited by the user while their size represents the probability that the user visits the corresponding network site.

3.3. User Profiler (UP)

As illustrated in Figure 2, the UP leverages the network topology created by the TG and the mobility information output by the UMM to simulate the users (subjective) reactions s to the corresponding experiences in the network. As generally done in the field of QoE research [19, 20, 11], in this work we assume the users reactions to be binary, i.e., s $\in [0, 1]^N$. In details, the i-th user reaction s_i is defined as:

$$s_i = \begin{cases} 1, & \text{if the } i\text{-th user is dissatisfied with her/his network service} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

It is well known from the literature that the duration of a network disservice has great impact on the experience perceived by a user. As an example, in the case of video streaming, QoE is primarily influenced by the frequency and duration of stalling events [21, 2]. Similarly, for web browsing, the number and duration of IRAT handovers is shown to have a strong negative impact on users experience [22]. In both cases users are observed to tolerate a certain amount of disservice before expressing a negative opinion: for video streaming, one stalling event per clip is acceptable as long as its duration is below 3 seconds, while for web browsing a single IRAT handover is generally tolerated.

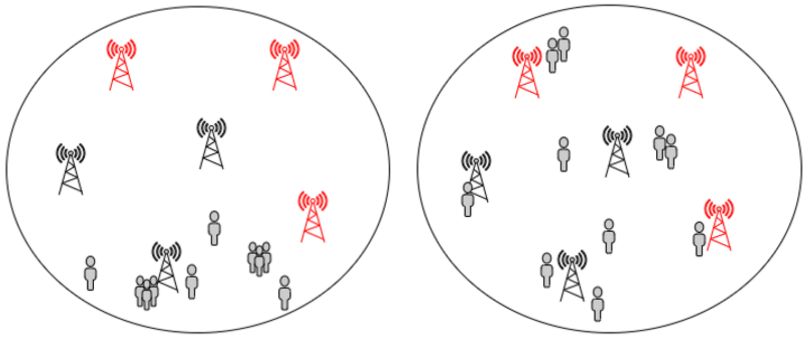
Following these observations, it is reasonable to link the user satisfaction s_i to the time spent in under-performing or malfunctioning network sites. Operatively, the UP leverages the set of under-performing sites $\mathcal{J}_u$ and the users visit times $t_{i,j}$ to generate users satisfaction according to the following:

$$s_i = \begin{cases} 1, & \text{if } \sum_{j \in \mathcal{J}_c} t_{i,j} \geq u_i T \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where T is the simulation time horizon and u_i is a percentage value corresponding to the *user tolerance*. In other words, we assume that each user has a specific patience level with respect to negative network experiences. Intuitively, the higher the tolerance of a user the more she will tolerate low service quality during her network activity. To model the heterogeneity of the users, we assume that the user tolerance u_i is a Gaussian-distributed random variable with mean μ and standard deviation σ, i.e., $u_i \sim \mathcal{N}(\mu, \sigma^2)$. For the sake of clarity, we specify that while the average profile of the users is defined jointly by the mean and the standard deviation of the Gaussian distribution, the heterogeneity of the population is embedded in the ratio between the standard deviation and the mean of the distribution, i.e. in the so-called coefficient of variation σ/μ of the distribution. Later in Section 4.2 we will discuss about the choice of their values. Finally, we observe that the reported user satisfaction depends also on factors completely unrelated with the network service itself, such as the ones relative to users personal attitudes and expectations [10]. The UP models such noisy behaviours by generating a percentage ψ of the satisfaction labels s_i at random, regardless of the sites visited by users and their tolerance. Again, ψ represents a hyper-parameter of the model that can be set by the operator to simulate different population types.

3.4. Survey Delivery (SD)

Any crowdsourcing-based network monitoring system is limited by the associated *network coverage*, i.e., the percentage of sites visited by users answering the surveys, as it is not possible to detect under-performing sites for which no information from users is available. As aforementioned, the number of users which answer to surveys is generally small compared to the total number of customers, and the set of corresponding Ground Truth (GT) responses s_{gt} is much smaller than s. The SD component of the proposed framework simulates the process of administrating satisfaction surveys to the customers, and can therefore be thought of as a sampling process of the true users reactions. In this paper we consider two scenarios for the surveys delivery strategy:



(a) RD: low network coverage scenario

(b) OD: high network coverage scenario

Figure 4: Strategies for the delivery of satisfaction surveys: a network coverage perspective. User icons refer to the users who replied to the received surveys, while blackish and redish base stations refer to regular and anomalous network sites respectively.

- Random Delivery (RD): in the most general case, we assume that the set of users answering the surveys is randomly sampled from the total population. This strategy is represented in Figure 4(a), where user icons represent those users who replied to the received survey. In this (unlucky) example the operator has low network coverage, as the received responses do not cover under-performing sites in the network.

- Optimized Delivery (OD): with this policy, depicted in Figure 4(b), the operator delivers surveys in a way to maximize the network coverage. This is done by leveraging the users visit times $t_{i,j}$ to set up an optimization problem (formalized in Section Appendix A) that selects the smallest set of users whose responses would allow to maximize the number of visited sites. Clearly, in this case we assume that the operator puts in place an incentive strategy for the voting procedure, such that a user selected from the optimization problem is rewarded for her answer (e.g. with premium data plan access for limited time or other incentives), as they are proven to foster users participation to this type of campaigns [23, 24, 25, 26].

Regardless of the chosen delivery strategy, the SD block samples the set of users reactions s and returns a set of GT users feedbacks s_{gt}, which is input to the under-performing sites ranking and detection algorithm. **Note that the output of this block is assumed to be uniformly distributed with respect to users personal details (such as age, gender, education level, job, etc.), as it should**

always be targeted by mobile operators in practical scenarios to ensure that the set of GT users reactions is a representative sample of the satisfaction of the whole customer base.

3.5. Under-Performing Sites Ranking and Detection Algorithm

At this point, the detection system mentioned at the end of Section 2 can be used to detect/rank under-performing sites in the network. The ranking algorithm will leverage: i) the user-specific cells visit times information generated by the UMM; ii) the set of GT survey responses generated by the SD block and iii) a pre-trained ML model $f(\cdot)$ to predict the satisfaction feedback of all those users who did not answer a survey. **Note that the ML model is assumed to be already trained: while detailing which learning features belong to $\mathrm{X_a}$ and $\mathrm{X_{na}}$ is outside the scope of this paper, we underline that in this work the joint distributions of model features and classification target of answering and non-answering users are assumed to be of the same type (as it usually happens when users belong to the same environment [27, 28], e.g., a urban scenario).** For the sake of clarity, we remark that the ranking algorithm is blinded about the true location of the under-performing sites, i.e., it does not know which network site belongs to $\mathcal{J}_u$.

For each network site in the network topology, the algorithm computes a score r_j according to the following procedure:

1. First, the set $\mathcal{V}_j$ of all the dissatisfied visitors of site j is selected. Note that $\mathcal{V}_j$ contains all those users such that $t_{i,j} > 0$ and the associated ground truth or predicted satisfaction is 1. Considering that a user visits multiple sites and its dissatisfaction may be due only to one of them, we tighten the time constraint as it follows:

$$t_{i,j} \geq \xi \sum_j t_{i,j} \tag{3}$$

   where ξ is a percentage that acts as an activation threshold for considering site j as responsible to the experience of the i-th user. Further details about the choice of the value of ξ will be given in Section 4.2.

2. Then, considering only users relative visit times above the threshold ξ, the site score is computed as it follows:

$$r_j = \sum_{i \in \mathcal{V}_j} \frac{t_{i,j}}{\sum_j t_{i,j}} \cdot \frac{t_{i,j}}{\sum_i t_{i,j}} \tag{4}$$

The proposed scoring rule takes into account: i) the fraction of time that a dissatisfied visitor i has spent in a site j with respect to the overall time spent by the visitor in the network (i.e., $\frac{t_{i,j}}{\sum_j t_{i,j}}$) and ii) the fraction of time that the visitor i has spent in cell j with respect to the overall service time of cell j (i.e., $\frac{t_{i,j}}{\sum_i t_{i,j}}$). In other words, if the score of a network site is high then it means that i) many dissatisfied visitors have visited the site for most of the time they have spent in the network and ii) the site has served dissatisfied visitors for most of its service time. Note that in this work $\sum_j t_{i,j} = T$ for each user $i \in \mathcal{U}$, although the scoring rule can be applied as it is even if the total visit time differs for different users. Finally, network sites are ranked in descending order according to r_j and the operator may use such an information to prioritise upgrading investments in the network. In fact, Equation 4 scores network sites without assuming anything about the cause of the degradation of their performance (if any), which could then be recognized by the operator after a more focused operative intervention. In particular, here we assume that the operator has a budget for investigating/upgrading k network sites. Feeding the value of k into the detection system allows to output a set $\hat{\mathcal{J}}_{\text{u}}$, $|\hat{\mathcal{J}}_{\text{u}}| = k$, containing the first k sites of the ranked list.

In the following Section we run the simulation framework in different scenarios and we compare the ranked set $\hat{\mathcal{J}}_{\text{u}}$ with the true set of under-performing cells $\mathcal{J}_{\text{u}}$ to assess the detection performance.

4. System evaluation

We use the proposed simulation framework to perform several experiments, with the goal of answering questions (Q.1-Q.5). This section is organized as follows: first, we provide details on the experimental setup in Section 4.1. Then, Section 4.2 focuses on the relationship between users satisfaction profile and the detection performance, providing an answer to Questions Q.1 and Q.2. Finally, Section 4.3 comments on the impact that both the surveys delivery strategy and the satisfaction prediction errors have on the overall ranking task, thus answering Questions Q.3, Q.4 and Q.5.

4.1. Experiments Overview

We feed the Topology Generator with information gathered from a real cellular network, currently operative in a middle-sized European city. The network is composed of 136 network sites deployed in an area of approximately 180 Km^2, whose locations are illustrated in Figure 5. We consider three different densities of

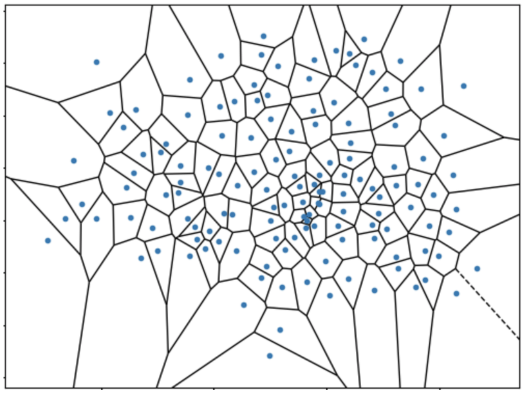


Figure 5: Voronoi representation of the considered network.

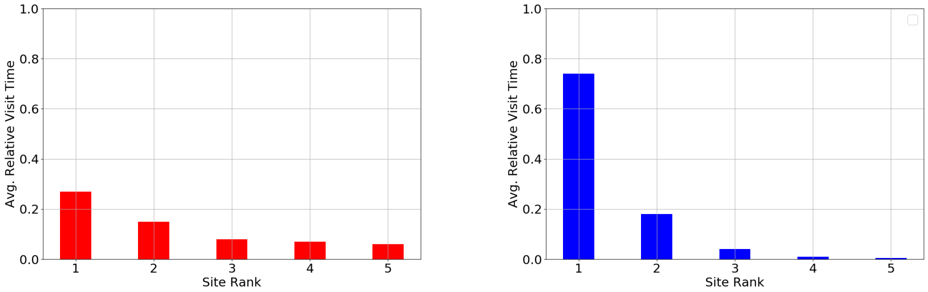
users per network site, corresponding to population sizes equal to $100\,\text{k}$, $10\,\text{k}$ and $1\,\text{k}$ users. Moreover, regardless of the population size, we consider two different mobility scenarios according to the value of the hyper-parameter γ that is input to the UMM:

- *Scenario 1 (S1)*: this case reproduces the setup described in [18], where a dataset containing one-year period trajectories of three million anonymized mobile-phone users is used to statistically estimate the values of the hyper-parameters. In this case, γ is set equal to 0.21;

- *Scenario 2 (S2)*: we reproduce the users mobility patterns observed in a dataset of 1500 anonymised customers in the same cellular network used to feed the TG for a period of 1 month. In this case, γ is set equal to 3. Therefore, this scenario is characterized by a lower tendency of the users to visit new sites compared to the first scenario.

For what regards the others parameters input in the UMM (i.e., α, β, and ρ), they are set to values estimated in [18] for both scenarios, that is $\alpha = 0.55, \beta = 0.8$ and $\rho = 0.6$. Moreover, the simulation time horizon T is set equal to 30 days for both mobility scenarios.

We plot in Figures 6(a) and 6(b) the average proportion of visit time resulting from the UMM for the case of $100\,\text{k}$ users for the two scenarios, ranked in decreasing order. The first bar refers to the average proportion of time spent by users in the most visited site, the second bar refers to the second most visited site and so on. As one can see, in both scenarios the distributions have a negative exponential trend, with the five most visited sites representing on average more



(a) Scenario S1.

(b) Scenario S2.

Figure 6: Average distribution of users visit time versus the site's rank of importance, for scenarios S1 (red bins) and S2 (blue bins).

than 60% and 95% of the overall users visit time in the network for S1 and S2 scenarios, respectively.

For what concerns the generation of the under-performing sites and the users profiling, we set the value of Ω (i.e., the number of under-performing sites in the network) to $\lfloor 0.1M \rfloor$, while μ and σ (i.e., mean and variance of the random variable u_i that controls users tolerance) are set so that the percentage of dissatisfied users ranges between 15% and 30% of the whole population. This is because cellular users feedbacks are typically unbalanced [7, 11, 10], i.e., the class of satisfied users is usually much larger then the class of dissatisfied ones. Finally, we leave to the next Section the discussion about the choice of the value of ξ, where we will also comment its relationship with the profile of the visiting population (i.e., with the hyper-parameters μ and σ).

4.2. Detection Performance and Users Heterogeneity

As a first experiment, we use the simulation framework to find answers to questions Q.1 and Q.2. We leave aside the problem of predicting users satisfaction, deactivating the sampling process in the Survey Delivery block and assuming an ideal scenario in which the operator has knowledge of the true satisfaction s for all the users. At the same time, we are interested in understanding how users heterogeneity impacts on the process of detecting under-performing sites. Therefore, we analyse the effect of parameters ξ, μ and σ on the detection performance. In particular, ξ is varied between 0.05 and 1 while μ takes values in [0.05, 0.15, 0.25, 0.35]. Note that for each value of the average user tolerance μ, the corresponding value of σ is adjusted in order to let the fraction of dissatisfied users be

Table 1: Values of μ and σ considered in the next experiment.

$\boldsymbol{\mu}$ (%)	$\boldsymbol{\sigma}$ (%)	$\boldsymbol{\sigma/\mu}$ (%)
5	1.5	30
15	3	20
25	3	12
25	12.5	50
35	3	8.5

within 15% and 30%. Table 4.2 summarises the values of μ and σ chosen in the experiments, as well as the value of the ratio σ/μ which embeds the heterogeneity of the customers population. We recall that the framework outputs a set $\hat{\mathcal{J}}_u$ containing the first k sites of the ranked list of under-performing sites, where k is an input parameters which depends on the operator financial budget. The metrics used for evaluating the detection performance are the *Precision* and *Recall at* k ($P@k$, $R@k$), defined as:

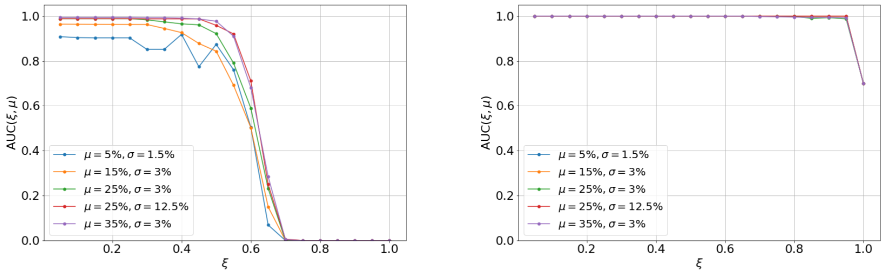
$$P@k = \frac{|\hat{\mathcal{J}}_u(k) \cap \mathcal{J}_u|}{|\hat{\mathcal{J}}_u(k)|} \tag{5}$$

$$R@k = \frac{|\hat{\mathcal{J}}_u(k) \cap \mathcal{J}_u|}{|\mathcal{J}_u|} \tag{6}$$

where the numerators correspond to the number of correctly detected sites, while the denominators equal k and Ω respectively.

As one can see, $P@k$ is defined as the proportion of the top-k ranked network sites that are actually under-performing. On the other hand, $R@k$ corresponds to the proportion of correctly detected under-performing sites. We perform several experiments with different values of ξ and considering for μ and σ the values reported in Table 4.2. Since an operator is unaware of the true number of under-performing sites, we evaluate the performance for different values of k (i.e., $k = 1 \dots M$), evaluating each time the metrics $P@k$ and $R@k$. Finally, for a fixed triple (ξ, μ, σ) we first compute the Precision-Recall ROC curve at different values of k, and then we summarize the performance of the system with the *Area Under the Curve* value, AUC(ξ, μ, σ). We highlight that the AUC summarizes the detection performance for all possible values of k.

We run the tests 10 times, each time generating a new random set of under-performing network sites. Figures 7(a) and 7(b) plot the average AUC values of the detection process when applied to S1 and S2 scenarios, respectively. Referring



(a) Scenario 1

(b) Scenario 2

Figure 7: AUC vs ξ for different average users tolerance to bad network events μ. The population size equals $100k$ users, who move according to mobility scenarios S1 (left) and S2 (right).

to S1 (Figure 7(a)), for $\xi \leq 0.50$ we observe that when the satisfaction feedbacks are retrieved from a population of excessively *touchy* users (i.e., when $\mu < 20\%$) the system AUC lowers on average by 15% and 10% with respect to the other population profiles for $\mu = 5\%$ and $\mu = 10\%$ respectively. Also, we observe that the detection system performs similarly for μ greater than 20%. When ξ is above 50%, the performance of the system rapidly drop as very few dissatisfied users meet the time constraint specified in Equation 3, due to the specific characteristics of the mobility type of the population. Differently, in scenario S2 (Figure 7(b)) the values of the AUC are similar regardless of the value of μ, σ and ξ and always greater or equal to 70%. As for the former scenario, the reason for which AUC values do not drop when users move according to scenario S2 are traced in the characteristics of the corresponding mobility pattern. In particular, also when $\xi = 1$ the system is still able to detect some under-performing sites as there is a non-empty set of dissatisfied users who have visited a single (under-performing) network site. From these observations we conclude that:

1. The detection performance depends i) on the way users move throughout the network and ii) on their subjective profiles. In general, considering that the true value of μ is unknown and uncontrollable by the operator, the ranking algorithm yields good detection performance in both scenarios. In particular, AUC values are greater than 75% for $\xi \leq 0.50$ when users move according to S1 while they are above 98% for $\xi \leq 0.95$ in case of scenario S2. This answers to question Q.1 introduced in Section 2.

2. Regardless of the average tolerance μ of the population and the users heterogeneity (i.e. of the ratio σ/μ), we observe that the detection performance

are stable with respect to the value of ξ, highlighting a certain inherent robustness of the system. This is promising, as an operator doesn't need to worry about i) estimating μ and σ or ii) tuning ξ with excessive care. As a rule of thumb, setting ξ as the mean of the average times spent in the most visited and second most visited site is a good working point. This provides the answer to question Q.2.

4.3. Satisfaction Prediction Trade-Off

To tackle question Q.3, we analyze the performance of the system in a more realistic case where the operator has only partial information about users satisfaction feedbacks. In particular, we investigate whether it is more reliable for an operator to perform detection considering only GT users feedbacks (i.e., only s_{gt}) or it is convenient to include also the predicted satisfaction labels (i.e., also $\hat{\mathrm{s}}$). Without loss of generality, the analysis is performed fixing the values of μ, σ and ξ to 25%, 3% and 0.2, respectively. The performance metric used for this analysis is the Recall at Ω, $R@\Omega$. Note that $R@\Omega$ has a maximum value of 1, if all under-performing cells are detected. The size of the GT users sets are fixed to 1% of the overall population size, i.e., we will assume a users response rate to satisfaction surveys of 1%, as observed in [11]. Consequently, the three populations of 1k, 10k and 100k users will be respectively characterized by an average density of 0.073, 0.73 and 7.35 GT users per network site, which we refer to as Low, Medium and High density. Concerning the delivery strategies, the OD optimization problem is solved by setting $n = 3$. For a fair comparison, the OD budget B is equal to the number of GT surveys used in the RD case.

4.3.1. Users QoE Prediction for Anomaly Detection

Any machine learning algorithm an operator can use to predict the user satisfaction $\hat{\mathrm{s}}$ will be characterized by a certain prediction error. Considering the nature of the satisfaction prediction problem, we assume the availability of a binary classifier and express its performance with the False Positive Rate (FPR) and the True Positive Rate (TPR) metrics. In details, the FPR corresponds to the rate of false alarms (satisfied users predicted as dissatisfied), while the TPR corresponds to the recall of the classifier (percentage of dissatisfied users detected). We observe that ML classifiers are characterized by several FPR and TPR working points, which can be traded-off by tuning a decision threshold. To perform a comprehensive analysis, we run the simulation framework assuming the availability of several ML classifiers in order to cover all possible FPR and TPR working points. In particular, we let both the FPR and the TPR vary between 0 and 1 with step-size equal to 0.05, thus analyzing 400 different performance points.

As an example, Figure 8 shows the obtained $R@\Omega$ for the case of 100k users moving according to Scenario 1, where 1k user satisfaction grades are sampled according to the RD strategy and the remaining are predicted with a ML classifier. Curves with different colours refer to classifiers with different (and fixed) FPR values, while the TPR is shown on the abscissa. Fixing a value of FPR (i.e., referring to one of such curves), allows to observe the recall of the sites detection system (i.e., $R@\Omega$) versus the TPR of the users satisfaction classifier. The colored stars refer to the performance of the classifier proposed in [11].

Considering a population of $100k$ users (Figures 8 and 9) we observe that:

- for a fixed TPR, the detection accuracy increases with decreasing FPR values;
- for a fixed FPR, the detection accuracy improves with increasing TPR values;
- for a fixed value Δ, decreasing the FPR by Δ is more beneficial than increasing the TPR by the same value.

These observations suggest that i) predicting that a satisfied user is dissatisfied (i.e., having a false positive) is more detrimental for the detection process than missing a dissatisfied user (i.e., missing a true positive) and ii) when deciding the FPR/TPR tradeoff of its classifier, an operator should prefer working points at low FPR rather than at high TPR. Moreover, this holds regardless of the population size, the mobility type and the surveys delivery strategy, as illustrated in Figures 12 and 13. This provides an answer to question Q.3.

4.3.2. To Predict or not to Predict?

It is worth analysing the best performance $R_{\mathrm{C}}@\Omega$ achievable by a realistic users satisfaction classifier C, such as the one we proposed in [11]. We plot its performance points (FPR$_{\mathrm{C}}$, TPR$_{\mathrm{C}}$) as coloured stars in Figures 8, 9, 12 and 13. For the sake of clarity, we summarize in Tables 2 and 3 the best values of $R_{\mathrm{C}}@\Omega$ achievable in all tested scenarios, and we compare it with $R_{\mathrm{gt}}@\Omega$, the best performance obtained leveraging only the available GT users satisfaction. We observe that $R_{\mathrm{gt}}@\Omega$ corresponds to the performance of a classifier that predicts each non-GT user as satisfied (i.e., FPR=0 and TPR=0), since satisfied users do not contribute to the ranking score. Therefore, $R_{\mathrm{gt}}@\Omega$ corresponds to the top-left brown point of a given performance cloud.

Table 2: (**S1**) Working points of a real binary classifier that yield best anomaly detection accuracy, where $R_{\mathrm{C}}@\Omega$ and $R_{\mathrm{gt}}@\Omega$ refer to the recall obtained when $k = \Omega$ and either the classifier C or only ground truth users information are leveraged by the system, respectively.

GT Users/Site	**Delivery**	(FPR$_{\mathrm{C}}$, TPR$_{\mathrm{C}}$)(%)	$R_{\mathrm{C}}@\Omega$ (%)	$R_{\mathrm{gt}}@\Omega$ (%)
Low	RD	(9,10)	20	7
	OD	(9,10)	**21**	8
Medium	RD	(15,26)	26	35
	OD	(15,26)	27	**45**
High	RD	(5,9)	42	72
	OD	(5,9)	42	**75**

Table 3: (**S2**) Working points of a real binary classifier that yield best anomaly detection accuracy, where $R_{\mathrm{C}}@\Omega$ and $R_{\mathrm{gt}}@\Omega$ refer to the recall obtained when $k = \Omega$ and either the classifier C or only ground truth users information are leveraged by the system, respectively.

GT Users/Site	**Delivery**	(FPR$_{\mathrm{C}}$, TPR$_{\mathrm{C}}$) (%)	$R_{\mathrm{C}}@\Omega$ (%)	$R_{\mathrm{gt}}@\Omega$ (%)
Low	RD	(20,33)	20	7
	OD	(30,45)	**22**	11
Medium	RD	(20,33)	57	41
	OD	(20,33)	**58**	49
High	RD	(20,33)	94	84
	OD	(30,45)	**95**	91

As one can see from Table 2, which refers to Scenario S1, we observe that for high density of GT users $R_{\mathrm{gt}}@\Omega$ is 30% and 33% higher than $R_{\mathrm{C}}@\Omega$, for RD and OD strategy respectively. For medium density of GT users per site the recall gap reduces to 9% (RD) and 18% (OD) while the situation is inverse when we consider low density of GT users, where we observe that it is better for the operator to leverage the classifier $f_C(\cdot)$ for detecting under-performing sites in the network. In fact, in such case $R_{\mathrm{C}}@\Omega$ is more than 13% better than $R_{\mathrm{gt}}@\Omega$ for both delivery strategies. For what regards Table 3, where we summarize the detection performance when users move according to mobility scenario S2, we observe that regardless of the tested densities of GT users per network site it is more convenient for the operator to include predicted users satisfaction feedbacks in the process of detecting under-performing network sites. In fact, in the latter case $R_{\mathrm{C}}@\Omega$ is always higher than $R_{\mathrm{gt}}@\Omega$, with a recall gap equal to 13%, 16% and 10% when RD strategy is adopted and equal to 11%, 9% and 4% when OD strategy is adopted for respectively Low, Medium and High density instances.

Such results are also illustrated in Figures 10(a) and 11(a).
Comparing the detection performance obtained in the two scenarios, we observe that the impact of satisfaction prediction on the detection process depends both on the density of GT users and on the characteristics of users mobility in the network. On the one hand, in Figure 10(a) it is clear that when using the binary classifier proposed in [11] there exists a critical GT users density (represented with a colored star) above which satisfaction prediction becomes detrimental in terms of detection performance. In fact, as observed in Figure 6(a), when users move according to mobility scenario S1 they visit on average multiple network sites for relatively similar visit times, this making in general harder identifying the network sites that caused a visitor's dissatisfaction. In such a scenario, it is not convenient for a network operator to enlarge the set of GT feedbacks if the density of GT users is large enough, i.e. if it is above the critical threshold, as it would increase the complexity of the problem due to the introduction into the system of satisfaction prediction errors. On the other hand, in Figure 11(a) we observe that satisfaction prediction benefits the detection process regardless of the density of GT users per network site. In fact, as shown in Figure 6(b), in this mobility scenario users visit the favourite site for most of their time in the network, thus increasing the probability that such a site is the most responsible for their dissatisfaction (if any) and in turn reducing the detrimental impact that satisfaction prediction errors have on the detection process. To conclude, since an operator is able to evaluate both the actual GT users density available and the characteristics of users mobility in its own network, it can also take a decision on whether or not to predict users satisfaction. This answers to Q4.

4.3.3. Random vs Optimized delivery

Finally, we discuss the obtained results in order to find an answer for Q.5. We observe from Figures 10(a) and 11(a) that the OD strategy always outperforms RD strategy. Moreover, in Scenario S1, using the OD strategy has the effect of moving the critical points (yellow star) towards lower GT users densities compared to the RD strategy. The reason of such a better performance is clearly due to the higher coverage that the OD strategy is able to reach. Figures 10(b) and 11(b) show the network coverage for the different scenarios: as one can see, the OD strategy allows to greatly increase the network coverage at different GT users density, which in turns impact on the achievable $R@\Omega$. However, we recall that in case of the OD strategy the operators may need to put in place incentive strategies for receiving the answers from the users selected by the optimization problem, thus incurring in higher costs.

5. Related Works

Many works in literature recognize the importance for cellular operators to monitor service levels at end hosts such to better understand which network events hamper users experience [13, 14, 8, 12, 10]. On the one hand, the computational power embedded in today's mobile devices let them be a powerful means for data collection, that can be then processed by the operators for diverse purposes [13, 14]. On the other hand, the analysis of users experiences in the network and of their corresponding subjective perceptions have become a fundamental benchmark for network operators, which often adopt crowdsourcing strategies to monitor and collect both objective and subjective users side information [8, 12]. In fact, Quality of Experience (QoE) models can be very helpful to quantify the relationship between users experience and network quality of service [10], considering that the more users share the same perception about similar network events the more likely those events share similar QoS characteristics [12]. Often, in order to augment the set of users reached by a data collection campaign, operators introduce rewards for users responses to encourage their participation. The goal of such incentive-based crowdsourcing strategies is to increase users participation while maximizing the Quality of Information (QoI) requirements of the reference application [23, 24, 25, 26]. A popular QoI requirement is the one of maximizing the data granularity, i.e., the area covered during the process of data gathering. In [23], the authors propose a recurrent reverse auction incentive mechanism that selects a representative subset of users according to their location given a fixed budget, augmenting the covered area by more than 60% while keeping fixed the number of collected samples. In [24], authors improve up to 80% the quality of a crowdsourcing mechanism in terms of data quantity and data coverage designing a proper incentive scheme for deep data gathering. Also, in [25] and [26] authors exploit crowd-workers predicted mobility traces to match spatial tasks with appropriate workers through an incentive-based crowdsourcing algorithm which maximizes the coverage probability under pre-defined budget constraints.

Regardless of the choice about using incentives or not, a common way for network operators to collect users QoE evaluations is to issue satisfaction surveys where the customers are asked what is their likelihood regarding the experienced mobile services. Then, operators can for example leverage the collected QoE feedbacks to plan actions to minimize the churn-rate of their customers,i.e., the percentage of customers who stop their contributions and move to a different operator due to unsatisfactory service [7, 9, 11]. In [28] the authors identify four main categories that influence the satisfaction of cellular users, namely *context*, *user profile*, *system* and *content*. The context considers factors like the purpose of using the service, the user's cultural background and the environment in which the user uses the service while the user profile considers individual psychological factors and memory. Finally, system and content address respectively technical influence factors (such as device-related problems) and resolution/format related issues.

However, a common problem found by cellular operators to assess the QoE of their customers through crowdsourced surveying campaigns is that few users usually respond to satisfaction surveys. To counteract this problem without incurring in additional costs (e.g., due to the use of rewarding mechanisms), usually operators implement techniques to estimate or predict users QoE feedbacks from objective network mesurements. Many works in literature tackle the (complex) issue of predicting users QoE in mobile networks, differentiating between short-term ([29, 30, 2, 31, 21]) and long-term ([7, 9, 11, 32, 5] users experiences. On the one hand, a short-term network experience refers to the case in which a user is first requested to interact with a mobile application under variable (and manually controlled) QoS network levels and secondly asked to provide a QoE evaluation of the experience. In [29], authors use in-smartphone measurements to feed several ML algorithms and predict users cellular users QoE with respect to several mobile applications. Leveraging a dataset comprised of 30 users, which were requested to watch short videos and give QoE feedbacks for each session, they obtain 91% and 98% accuracy on users feedbacks and service acceptability level respectively. A similar work is described in [30], where the authors conduct both lab tests and on field trials to analyse the impact of many network related features (e.g., bandwidth, latency, etc.) on users QoE of common mobile applications. Interestingly, in both [2, 21] the authors show that users QoE of video streaming applications is primarily influenced by the frequency and duration of stalling events, i.e., the longer the video playback re-buffering time the more likely the user will stop watching it. Similarly, authors in [31] recognize from subjective users QoE assessments that i) long video re-buffering and loading time are perceived as highly disturbing by the users and ii) fluent playbacks are preferred with respect to other video-related service indicators (such as resolution, frame rate or bit-rate). In other words, the longer the users experience disturbing network events the more likely their QoE will decrease.

On the other hand, the prediction of long-term users satisfaction is a much more challenging task to address. This is because a long-term user experience in a mobile network composes of many and different network events which together influence her QoE of the received cellular service. This means that users memory plays an important role in long-term QoE assessment processes, as discussed in [5]. Memory effects are also investigated in [7], where the authors leverage a large volume of network data regarding the experience of users in the network of one of the biggest mobile operator in China over several months, with the final aim of implementing a churn prediction system. They also integrate the prediction system in a closed loop automatic retention mechanism, with the aim of both acquiring new customers while retaining potential churners. Their results show that such a system improved the recharge rate of potential churners of more than 50%. With the same aim, in [9] authors introduce a modified random forest algorithm able to estimate a cellular customer's churn rate yielding an AUC value of 91.5%. Similarly, in [11] and [32] the authors correlate user-side network measurements with corresponding QoE feedbacks to train several ML algorithms and predict users satisfaction about network coverage and video streaming services, comparing also the prediction performance with the case where only radio access network measurements are used to train the ML classifiers. Moreover, considering that different users visit several network areas/elements and that the same area/element is usually visited by many different users, they point out that i) the information about ground truth and predicted users QoE feedbacks together with network measurements data can be used to recognize what in the network causes users dissatisfaction and ii) the impact of misclassification errors on such process could be somehow reduced when users QoE information are grouped on a single network area/element.

To conclude, considering that users QoE feedbacks are by definition subjective, an important issue regards the *reliability* of users answers to satisfaction surveys. Many works [3, 4, 6, 28] show that gathering reliable information from a crowd is a very challenging task. In [3] the authors give a probabilistic approach for supervised learning in a situation when there are possibly noisy replies collected from multiple users and there are no absolute gold standards (i.e. standard questions used to evaluate the level of reliability of experts). Similarly, authors in [4] propose an iterative algorithm for deciding best survey allocation and calculating a weighted estimate of the correct survey answer. Interestingly, in [6] it is shown how an incentive-compatible compensation algorithm together with approval-voting mechanisms successfully convert a significant fraction of incorrect answers to correct replies at the price of little increase in net expenditures.

6. Concluding Remarks

In this work we considered the process of crowdsourcing-based network monitoring, which may be used by cellular operators to detect problems in their network on the basis of users satisfaction feedbacks. We observe that several aspects need to be considered by an operator that decides to leverage such an approach. On the one hand, the heterogeneous reactions of users to service issues can hamper the detection of malfunctions in the network. On the other hand, it is not trivial to understand which network site is the main responsible of a user feedback, considering that each user visits many network sites for different amounts of time. Moreover, often very few users participate in the crowdsourcing process, thus forcing the operator to implement ML algorithms able to predict users satisfaction on the basis of objective measurements, in order to enlarge the knowledge base usable for monitoring purposes. This introduces a further aspect, which regards the impact of prediction errors on the detection of issues in the network. For all these reasons, we implemented a simulation framework that can be used by a cellular operator to analyse the application of a crowdsourcing-based network monitoring process in different realistic scenarios and investigate the related aspects. From the results we obtained, the following conclusions can be drawn:

- Under the reasonable assumption that users satisfaction depends on the performance of the visited network sites, it is possible for a network operator to rank/detect malfunctioning sites leveraging users satisfaction feedbacks with good detection performance (as shown in Figure 7);

- The detection process works regardless of the satisfaction profile of the visiting users, which in this work is represented by a random variable that controls users tolerance to bad network events. In particular, Figure 7 shows the robustness of the process with respect to the average users tolerance μ, its standard deviation σ and the threshold ξ;

- If a binary classifier $f(\cdot)$ is included in the detection process, working at low FPR rather than high TPR is more rewarding in terms of detection performance (as observed in Figures 8, 9, 12 and 13);

- When the coverage of the network ensured by GT users is low, it is convenient for an operator to leverage a ML classifier to predict the satisfaction of non-GT users such to augment the knowledge base usable for detection purposes. Conversely, for higher coverage values, the impact of the use of a ML classifier on the detection process depends on the way customers move in the network. On the one hand, if the users visit many network sites for similar times, it is better for the operator to rely only on GT users for detecting under-performing sites in the network. On the other hand, when the distribution of users visit times in the network is skewed towards few favourite sites, it is still convenient for an operator to predict unknown satisfaction levels. These results are summarised in Figures 10 and 11. Note that the above observations are true even when the classification performance of the ML classifier are modest, as shown in Figures 8, 9, 12 and 13;

- The implementation of delivery strategies that optimally allocate satisfaction surveys to users such as to maximize the network coverage increases the detection performance, as observed in Figures 10 and 11;

We believe these observations can be useful for a network operator willing to adopt crowdsourcing-based network monitoring.

Appendix A. Maximum Coverage Problem Formulation

In this Section, we describe the optimization problem that can be run by the Survey Delivery block to optimize the delivery of the surveys in order to maximize the network coverage. The optimization problem is the budgeted version [33] of a family of well-known problems known as *Maximum Coverage* (MC) problems. Given a collection $\mathcal{S}$ of items with associated costs defined over a domain of weighted elements and a budget B, the (budgeted) MC problem aims to find a subset $\mathcal{S}' \cup \mathcal{S}$ such that the total cost of items in $\mathcal{S}'$ does not exceed B and the total weight of the *covered* elements is maximized. In our case, we want to deliver satisfaction surveys to users such that the number of *covered* network sites is maximised, where a network site is covered if (i) it is visited by at least n users and (ii) each user spends more than ξ percentage of its own time in the site. Table A.4 summarizes the parameters that are leveraged by the optimization program. Let:

- x_i be a binary variable equals to 1 if a satisfaction survey is delivered to user i and zero otherwise;
- c_j be a binary variable equals to 1 if network site j is *covered* and zero otherwise;

Table A.4: Parameters considered in the MC problem.

Parameter	**Definition**
M	Number of Network Sites
$\mathcal{J}$	Set of Network Sites
$\mathcal{I} = \{1, .., N\}$	Set of Users
B	GT Users Budget
T	Time Horizon
$t_{i,j}$	User-site visit time
ξ	Percentage of time a user needs to spend in a site for covering it
n	Minimum number of visitors to consider a site covered

- $h_{i,j}$ be a binary association variable which equals 0 if the time that user i has spent in site j is lower than ξT (i.e., if it is not sufficient for coverage), while it can be both 0 or 1 otherwise.

Under these definitions, we propose an Integer Linear Programming (ILP) formulation for our version of the budgeted MC problem as it follows:

$$\max_{x_i} \quad \sum_{j \in \mathcal{J}} c_j \tag{A.1}$$

subject to:

$$\sum_{i \in \mathcal{I}} x_i \leq B \tag{A.2}$$

$$t_{i,j} \geq \xi \cdot T \cdot h_{i,j} \qquad \forall (i,j) \in \mathcal{I} \times \mathcal{J} \tag{A.3}$$

$$\sum_{j \in \mathcal{J}} h_{i,j} \geq 1 - M \cdot (1 - x_i) \qquad \forall i \in \mathcal{I} \tag{A.4}$$

$$\sum_{j \in \mathcal{J}} h_{i,j} < 1 + M \cdot x_i \qquad \forall i \in \mathcal{I} \tag{A.5}$$

$$\sum_{i \in \mathcal{I}} h_{i,j} \geq n - M \cdot (1 - c_j) \qquad \forall j \in \mathcal{J} \tag{A.6}$$

$$\sum_{i \in \mathcal{I}} h_{i,j} < n + M \cdot c_j \qquad \forall j \in \mathcal{J} \tag{A.7}$$

Equation A.1 represents the objective function, which aims at maximizing the number of distinct covered sites. Constraint A.2 limits the number of distinct users answering a survey to be lower or equal then the GT users budget B. Note that for large population sizes N the number of users which ensures full coverage could be smaller than the budget. Constraint A.3 controls the minimum time needed for a user i to contribute to the coverage of site j. Note that while the variable $h_{i,j}$ is forced to be 0 when the time spent by user i is not sufficient to be a covering visitor of site j, it is not constrained to be 1 if the coverage condition is met, so that the solver can decide which user is more convenient to activate to maximize the objective function. Constraints A.4 and A.5 control the selection of a generic user i for the delivery of the survey (i.e. the activation of user i), which arises from the activation of the corresponding time variable $h_{i,j}$ for at least one of the visited network sites. In particular, A.4 forces the variable x_i to be 0 in the case in which the summation on the left is 0, whereas A.5 forces the same variable to be 1 in the case in which the corresponding summation is strictly greater than 0. Note that when A.4 forces x_i to be 0, then A.5 deactivates, while the opposite happens when A.5 forces x_i to equal 1. Finally, constraints A.6 and A.7 set the requirements for considering a site as covered and work similarly to constraints A.4 and A.5.

References

[1] Cisco, Cisco Annual Internet Report (2018–2023), Technical Report, 2020.

[2] H. Nam, K.-H. Kim, H. Schulzrinne, Qoe matters more than qos: Why people stop watching cat videos, in: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE, 2016, pp. 1–9.

[3] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, Journal of Machine Learning Research 11 (2010) 1297–1322.

[4] D. R. Karger, S. Oh, D. Shah, Iterative learning for reliable crowdsourcing systems, in: Advances in neural information processing systems, 2011, pp. 1953–1961.

[5] T. Hoßfeld, S. Biedermann, R. Schatz, A. Platzer, S. Egger, M. Fiedler, The memory effect and its implications on web qoe modeling, in: 2011 23rd international teletraffic congress (ITC), IEEE, 2011, pp. 103–110.

[6] N. Shah, D. Zhou, Y. Peres, Approval voting and incentives in crowdsourcing, in: International conference on machine learning, 2015, pp. 10–19.

[7] Y. Huang, F. Zhu, M. Yuan, K. Deng, Y. Li, B. Ni, W. Dai, Q. Yang, J. Zeng, Telco churn prediction with big data, in: Proceedings of the 2015 ACM SIGMOD international conference on management of data, 2015, pp. 607–618.

[8] L. Tong, Y. Wang, F. Wen, X. Li, The research of customer loyalty improvement in telecom industry based on nps data mining, China Communications 14 (2017) 260–268.

[9] P. Swetha, S. Usha, S. Vijayanand, Evaluation of churn rate using modified random forest technique in telecom industry, in: 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, 2018, pp. 2492–2497.

[10] E. Boz, B. Finley, A. Oulasvirta, K. Kilkki, J. Manner, Mobile qoe prediction in the field, Pervasive and Mobile Computing 59 (2019) 101039.

[11] A. Pimpinella, A. E. Redondi, I. Galimberti, F. Foglia, L. Venturini, Towards long-term coverage and video users satisfaction prediction in cellular networks, in: 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), IEEE, 2019, pp. 146–153.

[12] D. R. Choffnes, F. E. Bustamante, Z. Ge, Crowdsourcing service-level network event monitoring, in: Proceedings of the ACM SIGCOMM 2010 conference, 2010, pp. 387–398.

[13] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, A. Vecchio, Smartphone-based crowdsourcing for network monitoring: opportunities, challenges, and a case study, IEEE Communications Magazine 52 (2014) 106–113.

[14] J. Ren, Y. Zhang, K. Zhang, X. Shen, Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions, IEEE Communications Magazine 53 (2015) 98–105.

[15] E. Hyytiä, J. Virtamo, Random waypoint mobility model in cellular networks, Wireless Networks 13 (2007) 177–188.

[16] K. Lee, S. Hong, S. J. Kim, I. Rhee, S. Chong, Slaw: A new mobility model for human walks, in: IEEE INFOCOM 2009, IEEE, 2009, pp. 855–863.

[17] A. Munjal, T. Camp, W. C. Navidi, Smooth: a simple way to model human mobility, in: Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, 2011, pp. 351–360.

[18] C. Song, T. Koren, P. Wang, A.-L. Barabási, Modelling the scaling properties of human mobility, Nature Physics 6 (2010) 818–823.

[19] I. Orsolic, D. Pevec, M. Suznjevic, L. Skorin-Kapov, A machine learning approach to classifying youtube qoe based on encrypted network traffic, Multimedia tools and applications 76 (2017) 22267–22301.

[20] J. Zhang, F. Ye, Y. Qian, A distributed network qoe measurement framework for smart networks in smart cities, in: 2018 IEEE International Smart Cities Conference (ISC2), IEEE, 2018, pp. 1–7.

[21] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, R. Schatz, Quantification of youtube qoe via crowdsourcing, in: 2011 IEEE International Symposium on Multimedia, IEEE, 2011, pp. 494–499.

[22] A. Balachandran, V. Aggarwal, E. Halepovic, J. Pang, S. Seshan, S. Venkataraman, H. Yan, Modeling web quality-of-experience on cellular networks, in: Proceedings of the 20th annual international conference on Mobile computing and networking, 2014, pp. 213–224.

[23] L. G. Jaimes, I. Vergara-Laurens, M. A. Labrador, A location-based incentive mechanism for participatory sensing systems with budget constraints, in: 2012 IEEE International Conference on Pervasive Computing and Communications, 2012, pp. 103–108.

[24] F. Ma, X. Liu, A. Liu, M. Zhao, C. Huang, T. Wang, A time and location correlation incentive scheme for deep data gathering in crowdsourcing networks, Wireless Communications and Mobile Computing 2018 (2018).

[25] L. Wang, Z. Yu, Q. Han, B. Guo, H. Xiong, Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks, IEEE Transactions on Mobile Computing 17 (2018) 1637–1650.

[26] Z. Song, C. H. Liu, J. Wu, J. Ma, W. Wang, Qoi-aware multitask-oriented dynamic participant selection with budget constraints, IEEE Transactions on Vehicular Technology 63 (2014) 4618–4632.

[27] A. Furno, M. Fiore, R. Stanica, C. Ziemlicki, Z. Smoreda, A tale of ten cities: Characterizing signatures of mobile traffic in urban areas, IEEE Transactions on Mobile Computing 16 (2016) 2682–2696.

[28] T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, P. Tran-Gia, Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing, IEEE Transactions on Multimedia 16 (2013) 541–558.

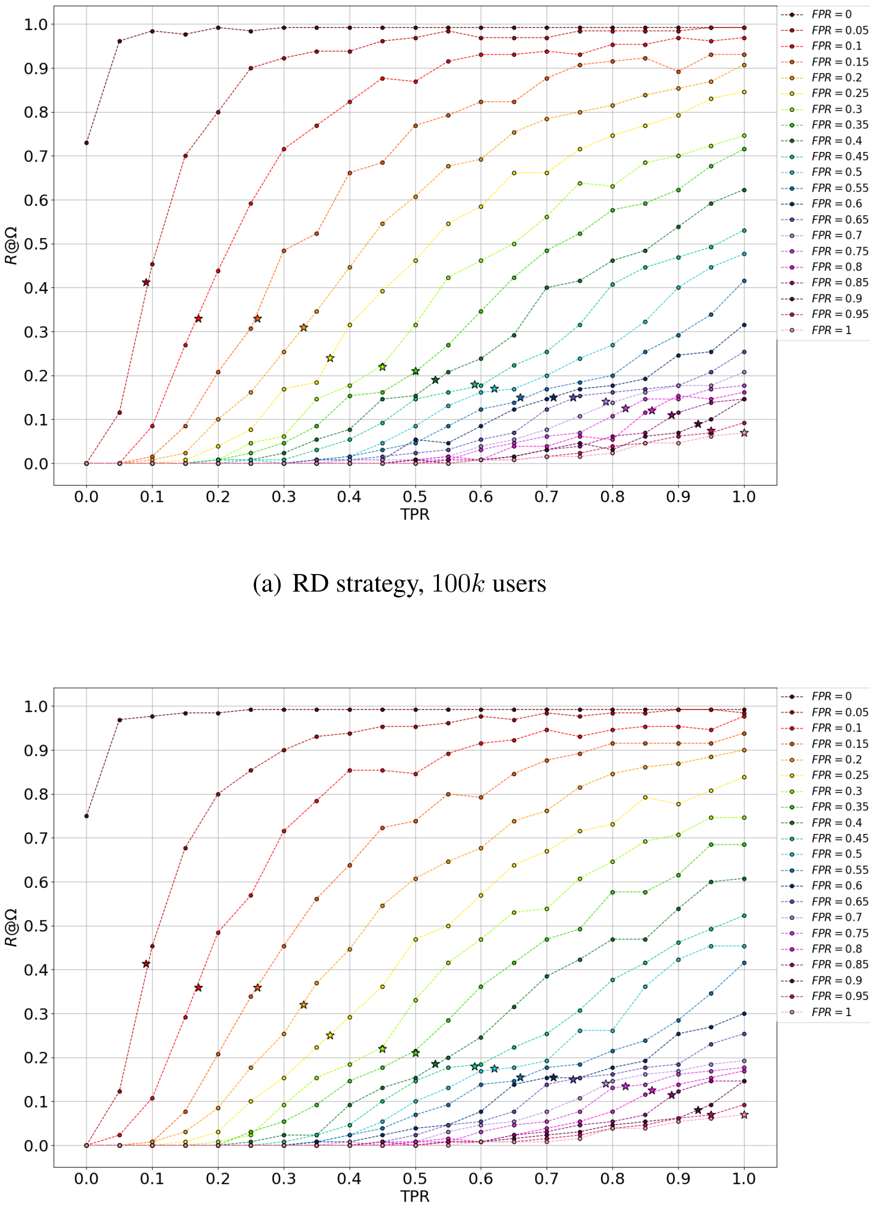
[29] P. Casas, A. D'Alconzo, F. Wamser, M. Seufert, B. Gardlo, A. Schwind, P. Tran-Gia, R. Schatz, Predicting qoe in cellular networks using machine learning and in-smartphone measurements, in: 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX), IEEE, 2017, pp. 1–6.

[30] P. Casas, M. Seufert, F. Wamser, B. Gardlo, A. Sackl, R. Schatz, Next to you: Monitoring quality of experience in cellular networks from the end-devices, IEEE Transactions on Network and Service Management 13 (2016) 181–196.

[31] T. De Pessemier, K. De Moor, W. Joseph, L. De Marez, L. Martens, Quantifying the influence of rebuffering interruptions on the user's quality of experience during mobile video watching, IEEE Transactions on Broadcasting 59 (2012) 47–61.

[32] A. Pimpinella, A. Marabita, A. E. C. Redondi, Crowdsourcing or network kpis? a twofold perspective for qoe prediction in cellular networks, in: 2021 IEEE Wireless Communications and Networking Conference (WCNC), 2021, pp. 1–6. doi:10.1109/WCNC49053.2021.9417464.

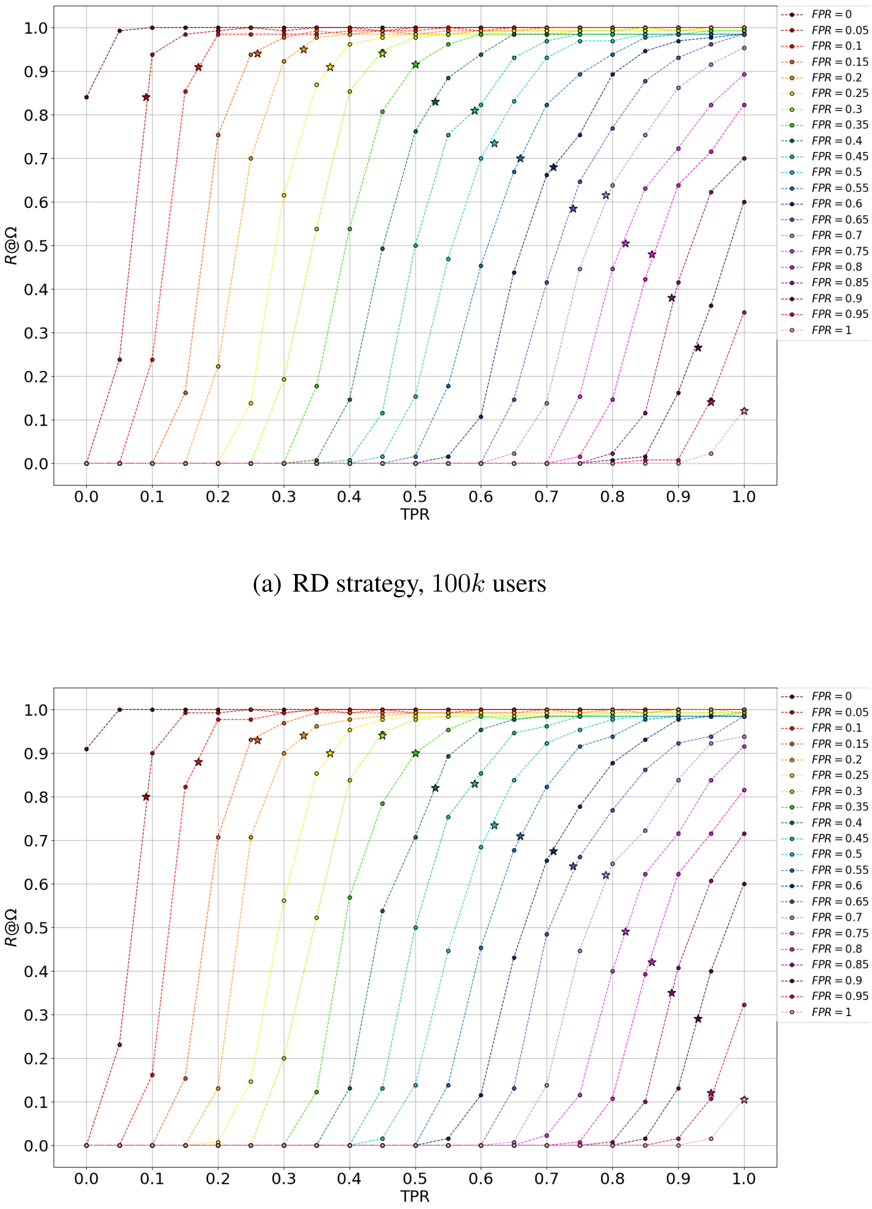
[33] S. Khuller, A. Moss, J. S. Naor, The budgeted maximum coverage problem, Information processing letters 70 (1999) 39–45.



(a) RD strategy, $100k$ users

(b) OD strategy, $100k$ users

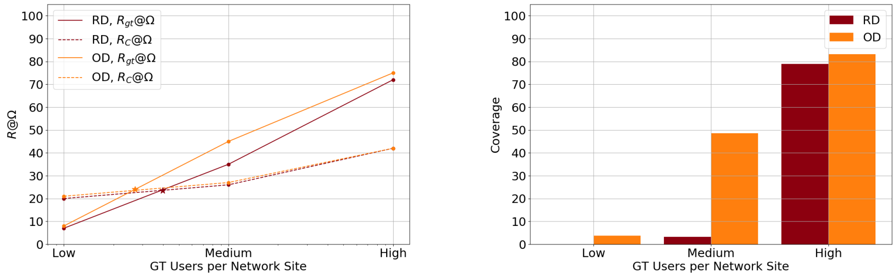
Figure 8: Performance Clouds for a population of 100k Users moving according to S1.



(a) RD strategy, $100k$ users

(b) OD strategy, $100k$ users

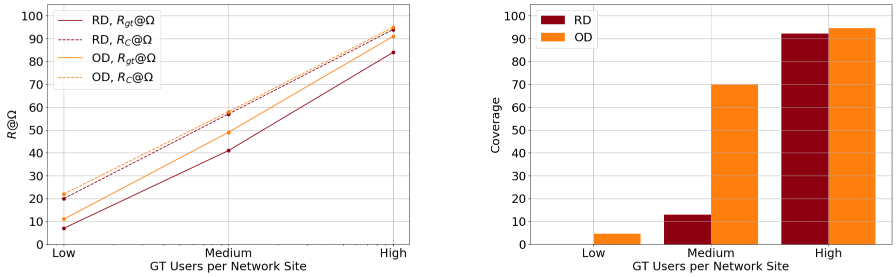
Figure 9: Performance Clouds for a population of 100k Users moving according to S2.



(a) $R_{\text{gt}}@\Omega$ (solid lines) and $R_{\text{C}}@\Omega$ (dashed lines) versus GT users per network site, for RD (red lines) and OD (orange lines) strategies.

(b) Average network coverage versus GT users per network site, for RD (red lines) and OD (orange lines) strategies.

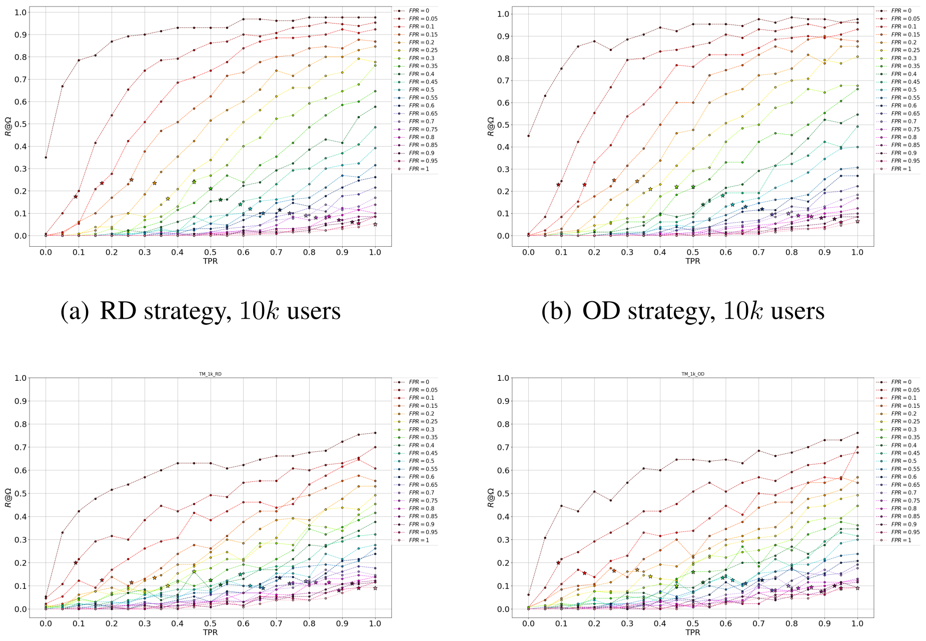
Figure 10: S1 Mobility Scenario.



(a) $R_{\text{gt}}@\Omega$ (solid lines) and $R_{\text{C}}@\Omega$ (dashed lines) versus GT users per network site, for RD (red lines) and OD (orange lines) strategies.

(b) Average network coverage versus GT users per network site, for RD (red lines) and OD (orange lines) strategies.

Figure 11: S2 Mobility Scenario.



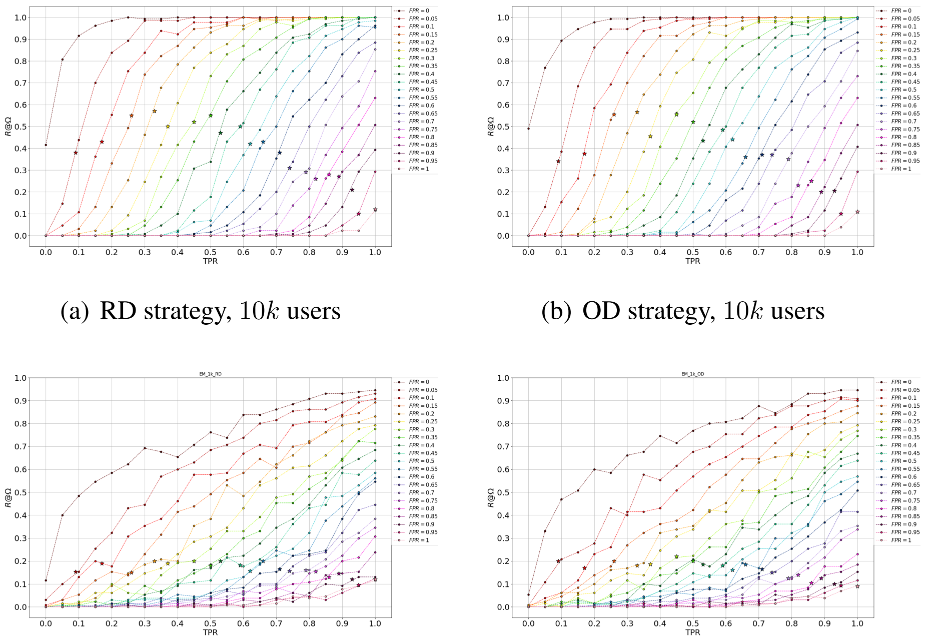
(a) RD strategy, $10k$ users

(b) OD strategy, $10k$ users

(c) RD strategy, $1k$ users

(d) OD strategy, $1k$ users

Figure 12: Scenario S1, 10k and 1k Users: Detection Accuracy versus Classifiers working points, for Random (left) and Optimized (right) surveys deliveries.



(a) RD strategy, $10k$ users

(b) OD strategy, $10k$ users

(c) RD strategy, $1k$ users

(d) OD strategy, $1k$ users

Figure 13: Scenario S2, 10k and 1k Users: Detection Accuracy versus Classifiers working points, for Random (left) and Optimized (right) surveys deliveries.