Fall 2016

Mechanisms for improving information quality in smartphone crowdsensing systems

Francesco Restuccia

Department: Computer Science

MECHANISMS FOR IMPROVING INFORMATION QUALITY IN SMARTPHONE CROWDSENSING SYSTEMS

by

FRANCESCO RESTUCCIA

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2016

Approved by

Dr. Sajal K. Das, Advisor
Dr. Sriram Chellappan
Dr. Abu Sayeed Saifullah
Dr. Simone Silvestri
Dr. Maciej Zawodniok

ABSTRACT

Given its potential for a large variety of real-life applications, smartphone crowdsensing has recently gained tremendous attention from the research community. Smartphone crowdsensing is a paradigm that allows ordinary citizens to participate in large-scale sensing surveys by using user-friendly applications installed in their smartphones. In this way, fine-grained sensing information is obtained from smartphone users without employing fixed and expensive infrastructure, and with negligible maintenance costs.

Existing smartphone sensing systems depend completely on the participants' willingness to submit up-to-date and accurate information regarding the events being monitored. Therefore, it becomes paramount to scalably and effectively determine, enforce, and optimize the information quality of the sensing reports submitted by the participants. To this end, mechanisms to improve information quality in smartphone crowdsensing systems were designed in this work. Firstly, the FIRST framework is presented, which is a reputation-based mechanism that leverages the concept of "mobile trusted participants" to determine and improve the information quality of collected data. Secondly, it is mathematically modeled and studied the problem of maximizing the likelihood of successful execution of sensing tasks when participants having uncertain mobility execute sensing tasks. Two incentive mechanisms based on game and auction theory are then proposed to efficiently and scalably solve such problem. Experimental results demonstrate that the mechanisms developed in this thesis outperform existing state of the art in improving information quality in smartphone crowdsensing systems.

ACKNOWLEDGMENTS

To my family. I would not have come so far without you.

To Lucia, the love of my life. It's because of you I am a better person.

To my advisors and mentors Sajal K. Das, Simone Silvestri, Giuseppe Anastasi, Marco Conti, John R. Lovitt, Bonnie Bachman, and all the other outstanding researchers I've had the pleasure to work with during these years, including Domenico De Guglielmo and Pierluca Ferraro. Your relentless passion and strength will always be a source of inspiration to me. Thanks also for your enormous patience.

To all my friends and enemies, for always encouraging me to work harder.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

In recent years, smartphones have become ubiquitous in our lives. According to 2014 Ericsson's mobility report [26], 8.4 billion smartphones will be active worldwide in 2020. These devices are equipped with rich multi-modal sensors, that provide information such as location, acceleration, temperature, and noise. In the near future, additional capabilities are envisioned, such as detecting pollution, lighting conditions, and more. These technological features, have contributed to the emergence of applications based on a new and promising paradigm known as smartphone crowdsensing*.

1.1. SMARTPHONE CROWDSENSING

The main idea behind smartphone crowdsensing is to enable ordinary citizens to actively monitor various phenomena pertaining to themselves (e.g., health, social connections) or their community (e.g., environment). For example, the cameras on smartphones can be used as video and image sensors [7], the microphone can be used as an acoustic sensor [20], and the embedded global positioning system (GPS) receiver can be used to gather accurate location information, while gyroscopes, accelerometers, and proximity sensors can be used to extract contextual information about the user (e.g., if the user is driving [67]). Further, additional sensors can be easily interfaced with the phone via Bluetooth or wired connections (e.g., temperature sensors [81]). Real-life applications, which can take advantage of both low-level sensor data and high-level user activities, range from real-time traffic monitoring applications like

*For the sake of generality, in this thesis the term *smartphone crowdsensing* will be used to designate applications where participants voluntarily contribute sensor data for their own benefit and/or the benefit of the community by using their phones. Such a notion therefore includes mobiscopes [1], opportunistic sensing [10], and equivalent terms such as mobile phone sensing, participatory sensing [47], or simply crowdsensing. It also covers specific terminologies focusing on particular monitoring subjects, such as urban sensing [10], citizen sensing [9], people-centric sensing [10], [11], and community sensing [45].

Nericell [63] or *Waze* [87] to air [19, 59, 68] and noise pollution [59, 73], social networking [62], crime monitoring [12], smart parking and so on [30, 66, 67, 69, 92]. An excellent survey of applications based on the smartphone crowdsensing paradigm may be found in [42].

Although a clear consensus on the best architecture for smartphone crowdsensing systems has not been reached yet, the majority of the existing smartphone crowdsensing applications utilize a centralized cloud-based architecture, depicted in Figure 1.1.

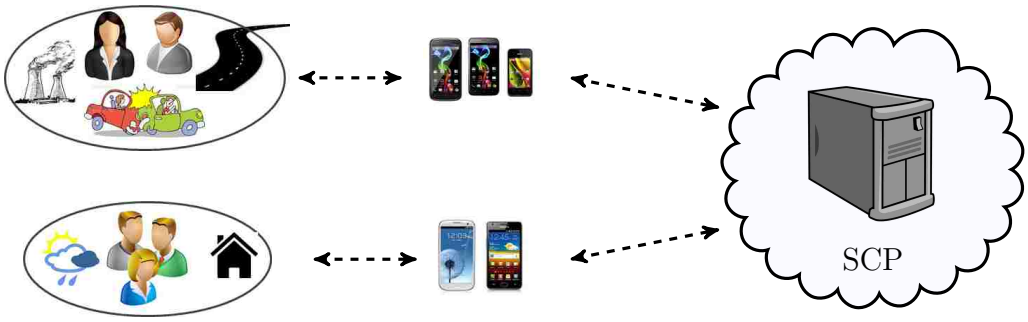


Figure 1.1. Smartphone crowdsensing architecture.

In particular, volunteers[†] use mobile phones to collect sensor data and submit via wireless data communication links to a smartphone crowdsensing platform (SCP) located in the cloud. The crowdsensing tasks on the phones can be triggered manually, automatically, or based on the current context [17]. On the SCP, the data is analyzed and made available in various forms, for example, graphical representations or maps showing the sensing results at an individual or community scale. The results may be displayed locally on the users' mobile phones or accessed by the broader public

[†]In this thesis, we will use the words "users", "participants", and "volunteers" interchangeably to indicate a person contributing to the smartphone crowdsensing campaign.

through web-portals, depending on the application needs. Three are usually the components of a smartphone crowdsensing system.

- The main component of the system are the *users*, whose task is to use their smartphones (and in particular, the sensing application) to capture different kinds of sensor data, such as location, images, sound samples, accelerometer data, biometric data, and barometric pressure. However, *opinions* about the sensing area or an environmental phenomenon, for example, traffic status or weather information, may also be provided by the users. In practical implementations of smartphone sensing systems, users typically register with the system by providing using a username and password [87] that allows their contributions to be uniquely identified.
- The *sensing application* (app), deployed on the users' smartphones, is distributed through common application markets like *Google Play* or *App Store*, or is retrieved from a mobile cloud computing system [27]. It is responsible for providing the users with a friendly user interface for data acquisition and visualization. In particular, data acquisition may be triggered by the users themselves or may be elicited by the app, on a one-time on-demand basis or periodically.
- The backend component of the system is the *smartphone crowdsensing platform* (SCP), responsible for the filtering, elaboration, and redistribution of sensed data, as well as coordinating every operation performed by the system. This component is usually implemented by a set of servers dedicated to the processing of sensed data [74]. The PSP also ensures efficient storage and elaboration of the sensed data coming from the users, which may be stored in relational databases [32], or databases specially adapted to the management of sensor readings, for example, *sensedDB*.

Along with the main elaboration system, the SCP might leverage a *reputation system* and an *incentive mechanism*. Briefly, the target of a reputation system is

to predict the reliability of the data sent by the users based on their past behavior [36, 84] so as to filter out unreliable reports. Conversely, the target of an incentive mechanism is to encourage participation of users by appropriately rewarding the users for their contributions to the smartphone crowdsensing campaign. Since this work will mainly focus on these two components, the state of the art research regarding reputation systems and incentive mechanisms for smartphone crowdsensing systems will be surveyed in details in Section 2.

Smartphone crowdsensing provides a significant number of advantages with respect to previous sensing paradigms (such as, for example, wireless sensor networks):

- The lack of a fixed sensing infrastructure dramatically eases deployment and maintenance costs associated with the administration of the sensing system;
- The sheer number of smartphone users, coupled with the ubiquitousness of WiFi and 3/4G cellular-based Internet connectivity, allows a level of spatio-temporal sensing coverage impossible to achieve in previous sensing paradigms;
- The presence of people in the sensing loop provides the opportunity to acquire *opinions* along with sensor data, allowing the emergence of complex mobile applications such as real-time traffic monitoring [82], [87], [95];
- The widespread availability of software development tools and markets for smartphone applications (apps) makes development and distribution of smartphone crowdsensing software relatively easy.

1.2. MOTIVATION AND CONTRIBUTIONS

The most distinctive characteristic of smartphone crowdsensing is that it relies completely on the voluntary commitment of participants to submit up-to-date and reliable information to the SCP. This implies that one of the key factors for the success of smartphone crowdsensing applications is *determine, enforce, and optimize the Information Quality (IQ) of the reports sent by participants.* IQ is often defined as

a measure of the value which the information provides to the user of that information [8]. In this context, "quality" is often perceived as subjective and the quality of information can then vary among users and among uses of the information. For this reason, the definition of IQ for smartphone crowdsensing still remains an open research issue.

In order to deal with the challenging problem of optimizing the IQ in smartphone crowdsensing, existing work has so far mainly focused on addressing two specific issues that pertain to such broader research topic:

- *Estimate and optimize the reliability of the sensing reports.* A number of reputation-based frameworks [37, 74, 85] have been recently proposed to address this issue. Reputation-based systems associate to each user a *reputation level*, which is estimated and updated over time. The rationale is to improve the reliability of information by filtering out reports coming from users having low reputation, as such reports are most likely to be unreliable. Another approach that has been followed is to use trusted platform modules (TPMs), which are hardware chips that reside on the participants' devices and ensure that the sensed data is captured by authentic and authorized sensor devices within the system [22, 33, 76].
- *Increase the amount of sensing reports received by the participants over time.* A significant number of incentive mechanisms, mostly based on game theory citeTadelis-game2008 and auction theory [46], have been proposed [75] to increase the amount of sensing reports received by users. The rationale is to formulate the incentivization problem as an optimization problem, in which the mechanism selects the participants and computes their reward so as to maximize an objective function defined before (e.g., sensing coverage or information reliability). Rewards usually are given to the users proportionally to their contribution to the smartphone sensing campaign. For example, rewards made may

be based on submitting a report close to a desirable location [38], how a report contributes to the social welfare [55, 79], the number of reports that a user sent, or the time dedicated to collecting and submitting sensing reports [89]

A number of research issues, however, still remain. In particular, the main issue of existing reputation-based frameworks is that user reputation is updated by considering contextual factors, such as location and time constraints. Given user location and timestamp of reports are easily forgeable quantities, the solution already proposed may not perform well in practical smartphone crowdsensing systems, where malicious users can voluntarily tamper with their GPS location and timestamp of reports. Moreover, existing incentive mechanisms assign sensing tasks irrespective of the mobility of users over the sensing area. This approach oversimplifies the problem formalization and subsequent analysis, but may not be applicable to real-world sensing scenarios where sensing tasks are spatio-temporal constrained. These reasons motivated this work and the following novel contributions to the state of the art.

- A novel *Framework to optimize Information Reliability in Smartphone-based participaTory sensing* (FIRST) is developed, which leverages the collective action of mobile trusted participants (MTPs) to securely assess the reliability of sensing reports. FIRST mathematically models and solves the challenging problem of determining before deployment the minimum number of MTPs to be used in order to achieve desired classification accuracy, by also leveraging a novel algorithm based on image processing. FIRST was evaluated through experiments leveraging real-world mobility traces of taxi cabs in San Francisco, Rome, and Beijing, and through an implementation in iOS and Android of a system leveraging human participants to monitor the attendance to various events at the IEEE PerCom 2015 conference. Experimental results demonstrate that FIRST is remarkably effective in optimizing information reliability by reducing the impact of the three considered security attacks, while outperforming

state-of-the-art literature by achieving on the average a classification accuracy of 80% in the considered scenarios.

- The problem of maximizing the likelihood of successful execution of the sensing tasks when participants having uncertain mobility compete for offering their sensing services is studied. The problem is cast in the context of truthful budget-feasible reverse auction design with submodular objective function, where the crowdsensing system is the buyer and the participants are the sellers. After demonstrating that the problem is NP-hard, two incentive mechanisms based on game theory are proposed. To deal with a large number of participants, an implementation on the well-known MapReduce framework is provided. The mechanisms were evaluated by considering a road traffic monitoring application that uses real-world mobility traces of taxi cabs in San Francisco, Rome, and Beijing. Experimental results demonstrate that the mechanisms outperform the state of the art by improving its performance of 30% and are highly scalable, obtaining on the average 12x speedup in the considered experimental setup.

1.3. ORGANIZATION

A survey of state-of-art research work pertaining to this thesis is presented in Section 2. Section 3 presents FIRST, a framework for optimizing IQ in smartphone crowdsensing with minimum amount of mobile trusted participants. Section 4 presents the two auctions.

2. AN OVERVIEW TO SMARTPHONE CROWDSENSING

Smartphones are becoming more and more central to our everyday lives. While early mobile phones were designed to primarily support voice communication, technological advances helped reduce the divide between what we consider conventional phones and computers. As this technological gap further diminished, a new paradigm is fast emerging: people are beginning to replace their personal computers with smartphones. The mobility and power afforded by smartphones allow users to interface more directly and continuously with them more than ever before; smartphones represent therefore the first truly ubiquitous mobile computing device.

A critical component that opens up smartphones to new advances across a wide spectrum of applications domains is founded on the embedded sensors in these devices. Sensor enabled smartphones are set to become even more central to people's lives as they become intertwined with existing applications, such as social networks and new emerging domains such as green applications, recreational sports, global environmental monitoring, personal and community healthcare, sensor augmented gaming, virtual reality, and smart transportation systems. As such, the global density of smartphones will provide ground breaking ways to characterize people, communities, and the places people live in as never possible before.

The advance in smartphone-based applications is enabled not only by embedded sensing, but by a number of other factors as well, including, increased battery capacity, communications and computational resources (CPU, RAM), and new large-scale application distribution channels – also called app stores, such as Apple App Store, Google Android Market, Nokia Ovi Store, to name a few. By mining large scale sensing data sets from applications deployed on smartphones through the app stores and using machine learning techniques to analyze the data, it is now possible

to discover patterns and details about individuals and ensembles of people not possible before. As a result, real-time and historical sensing data from communities of people can be leveraged to make inferences at scale, and advancing the design of new people-centric sensing systems across many diverse application domains [30, 80, 88].

Although a clear consensus on the best architecture for smartphone crowdsensing has not been reached yet, most of the commercially available smartphone crowdsensing application employ an architecture depicted in Figure 2.1.

This figure illustrates the main activities of the sensing and incentivization processes in a smartphone sensing system. Since this collection of activities is performed again and again over the lifetime of the sensing campaign, henceforth, we will use the term *sensing round* to refer to the execution of the following four steps.

Sensing task advertisement. In this phase, the smartphone crowdsensing platform (SCP) communicates to the users the list of sensing tasks that need to be executed during the current sensing round. In particular, each sensing task specifies a series of requirements, such as the sampling rate requested [43], minimum sensing time [41], maximum distance from specified location [83], or task expiration time [48, 94]. For example, a sensing task might be "report the current traffic status near the Golden Gate bridge by 5:00PM". Additional parameters may be sent, such as quality of information requirements [2, 50]. Sensing tasks can be advertised statically [24, 28, 89] or dynamically [29, 74]. In some cases, depending on the application, tasks can be retrieved by the users asynchronously, e.g., each day [49], or whenever requested by the participants [48].

Private information disclousure. After the advertisement of the sensing tasks, the SCP collects information about the participating user, often called the *type* [89] of the user, which can be leveraged by the SCP to make a choice regarding the scheduling of sensing tasks. For example, users may supply a *bidding value*, which is included in the user type for use in auction-based incentive mechanisms. The SCP

may collect temporal information about the user, which can be used to determine the user's availability to perform sensing services [34]. The SCP may also acquire information regarding the characterization of the skills of a person [52, 54] and the incurred cost for the required sensing services, for example, privacy loss [23], energy consumption, or mobile data charges [55].

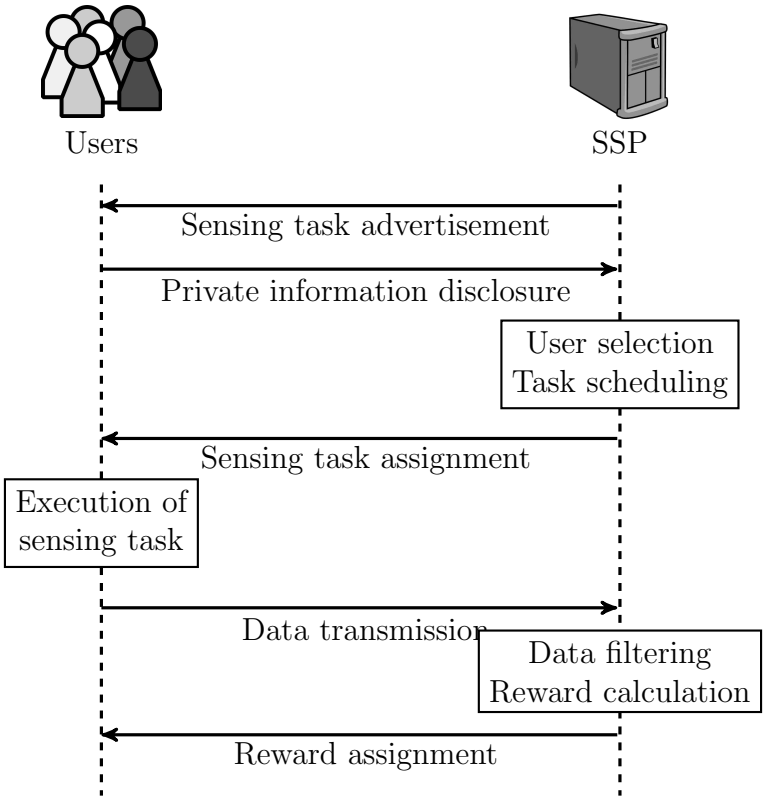


Figure 2.1. Sensing and incentivization activities within a sensing round.

User selection and Task scheduling. After receiving private information describing the users, the SCP selects a subset of users that will submit the sensed information to the SCP, and schedules the sensing tasks for each user. For example, users might be selected according to their geographical position [28, 56, 93], or according to the submitted bid [89], cost of sensing services [24], sensing effort [52]. In case

a reputation mechanism is used, users might be selected according to their reputation value [84, 86]. Furthermore, the SCP may schedule the sensing tasks according to the temporal availability of the users [15, 29].

Execution of sensing task and data transmission. After being selected and instructed on the sensing task to execute, a user is allowed to begin performing the sensing service using the sensing application. The sensing application may be designed to assist in collecting data according to one of the following sensing modes: manual, automatic, or context-aware [17]. Data collection might also be triggered by leveraging the *context*, occurring upon the detection of an event or condition (e.g., clapping of hands, as in [7]). The sensing application handles the transfer of the sensed data from the smartphone to the SCP, making use of communication infrastructure available to the mobile phone, such as WiFi or 3G/4G connectivity. For example, the sensor readings can be transmitted to the server using SMS or TCP connections [32], remote procedure calls [62], or web interfaces [27]. Recently, opportunitistic forwarding coupled with data fusion has been proposed as a viable method for data transmission [56].

Data filtering and Reward assignment. In this phase, unreliable data coming from the selected users is filtered according to the perceived Information Quality (IQ) of the reports themselves. If a reputation mechanism is used, the reports are filtered based on the reputation score of the user submitting the sensing report [84, 85, 86]. Users are rewarded for their services, usually according to the time dedicated to the sensing services. The information quality contained in sensing reports might also be a parameter to decide the amount of reward to assign to users [74].

3. THE FIRST FRAMEWORK

As discussed earlier, the inherent collaborative nature of smartphone crowdsensing implies that its success is strictly dependent on the *reliability* of the information sent by the participants. However, it is well recognized that participants may *voluntarily* submit unreliable information. For instance, participants may be maliciously aimed at *degrading* the received service to the other users of the application by conducting security attacks. In March 2014, to give an example, students from Technion-Israel Institute of Technology successfully simulated through GPS spoofing a traffic jam on *Waze* that lasted hours, causing thousands of motorists to deviate from their planned routes [6]. These (and similar) attacks are made extremely easy by smartphone applications (apps) like *LocationHolic* or *FakeLocation* [51], which allow participants to spoof their current GPS location.

To ease the impact of malicious participants, a limited number of *mobile trusted participants* (MTPs) may be employed to help build reputation scores in a secure manner, and thus 'bootstrap' the trust in the system [13]. Specifically, MTPs are participants that are hired by the sensing application to periodically generate reliable reports that reflect the actual status of the event that is being monitored around their location. This methodology is being successfully used in the *National Map Corps* project [58] developed by the U.S. Department of Geographical Survey (USGS), where MTPs (in this case, USGS employees) are employed to validate crowdsourced data, such as the exact location of schools and cemeteries*. MTPs are also used in the *Crowd Sourcing Rangeland Conditions* project [40], where Kenyan pastoralists are recruited as MTPs by researchers to validate sensed data regarding local vegetation conditions. The advantage of using MTPs with respect to existing approaches is the

*Website at `http://nationalmap.gov/TheNationalMapCorps/`

capability to tackle malicious and unreliable behavior by building *reliable* reputation scores, since MTPs are trusted entities. However, MTPs also inevitably represent an additional *cost* for the smartphone crowdsensing system, as MTPs need to be recruited.

In this section, the following questions will be investigated:

- What is the minimum number of MTPs we need to employ to ensure that the information quality will remain above a certain threshold?
- How does the mobility of the MTPs affect the optimum number of MTPs needed?
- What is the impact of non-trivial security attacks on the information quality when MTPs are employed?

To answer these questions, the *MTP Optimization Problem* (MOP) is formulated, which aims at minimizing the number of MTPs deployed (to minimize hiring costs) while guaranteeing the desired accuracy in classifying the collected reports as reliable or unreliable. However, several aspects make the MOP solution extremely challenging. For example, formalizing the relationship between the number of MTPs deployed and the resulting information reliability is significantly complex, since the latter is heavily influenced by the mobility of MTPs and other users. This motivated the following contributions.

- After describing the system architecture, a novel *Framework to optimize Information Reliability in Smartphone-based participaTory sensing* (FIRST) is proposed, which has three main components. A probabilistic model, called *Computation of Validation Probability* (CVP), calculates the probability that a user report is validated as a function of the number of MTPs deployed and user mobility. A novel image processing algorithm, named *Likelihood Estimation Algorithm* (LEA), leverages geographical constraints of the sensing area to

provide an approximation of the probability that a sensing report will be validated. Finally, an *optimization algorithm* (MOA) efficiently solves the MOP by using the results from CVP and LEA, and computes the minimum number of MTPs required to achieve the desired classification accuracy.

- The performance of FIRST is extensively evaluated by considering a smartphone crowdsensing (SC) application for monitoring road traffic, where real-world mobility traces[4, 72, 91] are used to emulate the mobility of participants. For comparison purpose, the state-of-the-art approaches [37, 74] are implemented. To test their performance, three security attacks previously defined in [64] are considered. Experimental results demonstrate that FIRST outperforms the state of the art and achieves high classification accuracy with relatively low number of MTPs, and is able to tackle effectively all the three considered attacks.
- The performance of FIRST is further evaluated on a practical implementation of a smartphone crowdsensing system, which was conducted at the IEEE PerCom 2015 conference. In this experiment, an app (for both iOS and Android devices) was designed, which was distributed to the interested participants (i.e., volunteers) at the conference. These volunteers sent reports regarding the conference participation, acting as users of the smartphone sensing system. Results show that FIRST outperforms previous approaches and achieves on the average a high classification accuracy of 80%.

The section is organized as follows. Section 3.1 introduces the system model and the MOP. Section 3.4 presents the FIRST framework and its CVP, MEA, and MOA components, while Section 3.5 presents the experimental results and Section 3.7 draws conclusions.

3.1. SYSTEM MODEL

The smartphone crowdsensing architecture taken into account (depicted in Figure 3.1) consists of a smartphone crowdsensing platform (SCP) which can be accessed through 3G/4G or WiFi Internet connection. The data collection process can be summarized as follows. First, participants download through common app markets like *Google Play* or *App Store* the smartphone crowdsensing app, which is responsible for handling data acquisition, transmission, and visualization (step 1).

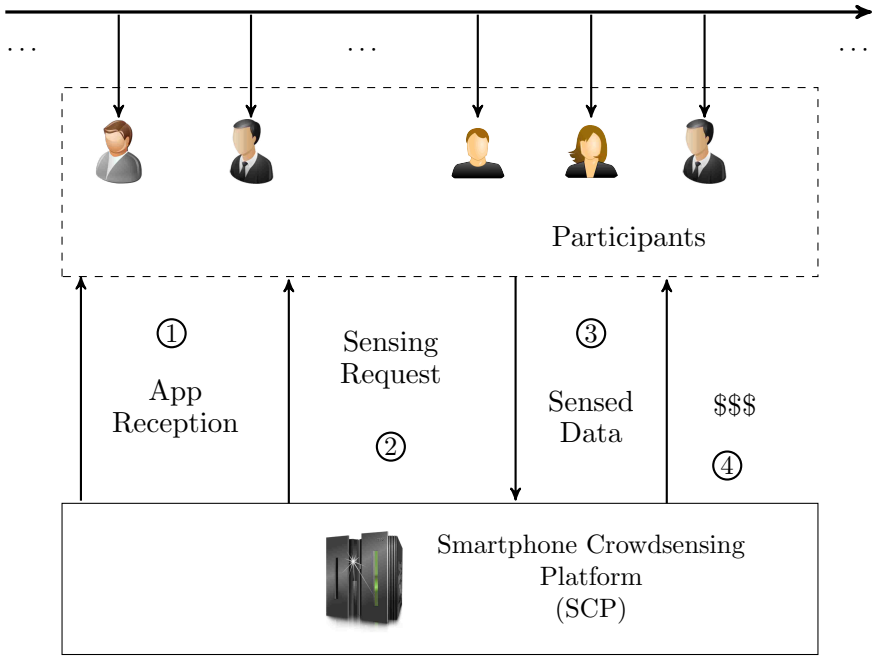


Figure 3.1. System model for the FIRST framework.

Then, the SCP sends (periodically or when necessary) sensing requests through the cloud to registered participants (step 2). The participants can answer such requests by submitting their sensed data (step 3), and eventually receive a reward for their services (step 4). Hereafter, the words "participant" and "user" will be used interchangeably.

As far as the sensing application is concerned, it is considered a sensing system in which the phenomenon being monitored is *(i)* quantifiable, *(ii)* dynamic (i.e., varies over time), and *(iii)* not subject to personal opinion. This includes phenomena measurable with physical sensors, for example, air/noise pollution levels [20], but also quantities that can only be measured by humans, such as occupancy level of parking lots [66], gas prices [21], traffic events (e.g., car crashes and traffic jams) [87], and so on. Furthermore, it is assumed that the *range* of the sensing quantity being monitored may be divided up into *intervals* or *categories*, which are specific to the smartphone crowdsensing application but are properly defined before deployment. For example, in a gas price monitoring system, the range of possible values could be from $2 to $3 dollars per gallon, divided into intervals of 10 cents each. In a traffic monitoring application, a different category for each traffic event (e.g., "Car Crash", "Road Closure", "Traffic Jam", and so on), like in the Waze app [87], could be specified. Moreover, a sensing report is defined as *reliable* if the quantity being reported falls into the interval the phenomenon is currently in (or belongs to that category). For example, if the actual gas price at a station is $2.46, a report is considered reliable if the reported value falls into the range [$2.40, $2.50).

As far as the security assumptions are concerned, the SCP is considered trustworthy in terms of its functionality (such as user registration, issuing credentials, receiving, processing, and redistributing data). Furthermore, confidentiality, integrity, and non-repudiation are assumed to be addressed by using standard techniques such as cryptography and digital signatures. It is also assumed that users may exhibit *malicious* or *unreliable* behavior; such behavior models are detailed below. In the following, users are assumed to be identified by the SCP via username and password and some sort of user-unique information (e.g., credit card information), meaning no sybil/rejoin attacks are possible.

- *Malicious:* These users are willingly interested in feeding unreliable reports to the system; their purpose is to either creating a disservice to other users (e.g., fake road traffic lines [6]), or gaining an unfair advantage w.r.t. other users.
- *Unreliable:* These users are not willingly submitting false information, but they still do it because of malfunctioning sensors or incapability in performing the sensing task [74].

FIRST provides a general approach to determine the reliability of each user depending on his/her behavior. In Section 3.5, three types of attacks are experimentally studied, namely the corruption, on-off and collusion attacks (previously defined in [64]), and it is proven that FIRST is able to quickly detect the malicious behavior and discard unreliable reports.

3.2. MOBILE TRUSTED PARTICIPANTS

In this study, the same approach used by the successful *National Map Corps* [58] and *Crowd Sourcing Rangeland Conditions* [40] projects is employed, and use mobile trusted participants (MTPs) to tackle the attacks described in the previous section. Specifically, MTPs are individuals who are able and willing to submit regularly reliable reports regarding the phenomenon being monitored or observed. These reports are used to *validate* users' sensing reports coming from nearby, and ultimately estimate the reliability of those participants. Such estimate is used to classify reports generated where MTPs are currently not present, as explained in the next sections.

To allow mathematical formulation, the sensing area is divided up into $\mathcal{S} = \{s_1, \ldots, s_n\}$ sectors, which may have variable size and represent the *sensing granularity* of the application. For example, in the gas price app, each gas station could be assigned to a single sector. In an air pollution monitoring application, a sector may be as large as a neighborhood of a city, whereas in a traffic monitoring application,

sectors may be as large as a city block. It is defined as $\mathcal{U} = \{u_1, \ldots, u_z\}$ the set of users contributing to the sensing application.

The MTP report validation process is modeled as follows. In order to validate user reports, it is assumed that the reports sent by MTPs are valid for a time period of T units. The value of T is a system parameter that is dependent on the variance over time of the sensing quantity being measured. For example, in a traffic monitoring application, a good value of T could be 5-10 minutes, while in a gas price monitoring app T can be much longer (in Section 3.5.2, the impact of T on the system performance is evaluated).

Definition 1: *Validation of sensing reports. Whenever a sensing report q is received from a user u_i in sector s_j, the platform checks whether a report from an MTP in sector s_j was received in the previous T time units. If yes, then the report is cross-checked with that coming from the MTP. If q is reliable (i.e., falls into the range of the report sent by the MTP), q is marked as* validated *and classified as reliable. Instead, q is rejected if unreliable. If q is not validated, it is classified reliable or unreliable depending on an algorithm discussed in Section 3.4.*

Figure 3.2 illustrates an example in which an MTP is moving over a sensing area comprising three sectors.

The locations at which the MTP submits a sensing report are marked as white circles, while users are depicted as black dots. The user reports from sector s_1 between $t = 0$ and T units are validated by using the MTP report sent at $t = 0$. Meanwhile, the MTP moves to sector s_2 and generates a new report at time $2T$, which then validates users reports from sector s_2 in the next time window. Similarly, the MTP report at $3T$ validates the user reports from sector s_3 in the time interval $[2T, 3T]$.

Examples of MTPs in urban sensing scenarios include, but are not limited to, professional drivers (i.e., taxi/bus), policemen, employees of the smartphone crowd-sensing application, or people commuting on a daily basis to their workplace. Henceforth, the MTPs will be considered as reliable, in sense that it is implied that their reports reflects the actual status of the event being monitored. This also implies that reports originating from the same sector during the same time window are supposed to be equivalent. The case in which trusted participants can be (up to some extent) unreliable has already been studied in [74].

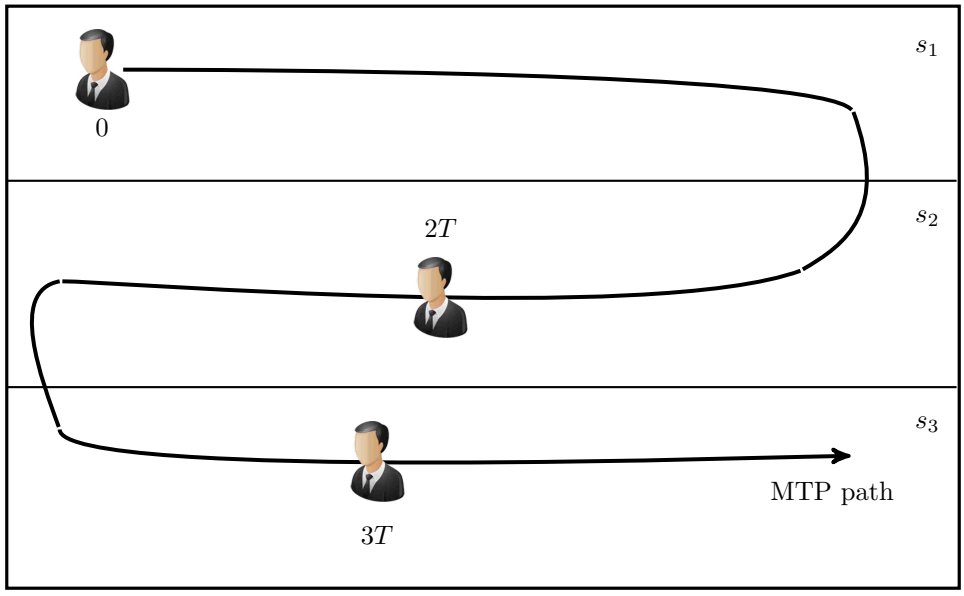


Figure 3.2. An MTP moving over the sensing area.

3.3. MTP OPTIMIZATION PROBLEM

It is intuitive that the number of validated sensing reports (and therefore, information reliability) increases as the number of recruited MTPs increases. However, in practical implementations, it is not feasible to assume unconstrained budget to recruit MTPs; the number of MTPs that can be used by the system will be limited

and therefore, insufficient to guarantee perfect information reliability. The question is then the following: is it possible to find a good estimation of the minimum number of MTPs that will allow the smartphone crowdsensing system to achieve desired classification accuracy? To this end, the MTP Optimization Problem (MOP) is defined and studied. Before that, the metric of classification accuracy is defined.

Definition 2: *Classification accuracy. Let A define the event of the system considering a report as reliable, and let F define the event of a user submitting an unreliable sensing report. Let E define the event of erroneously deeming reliable (resp. unreliable) an unreliable (resp. reliable) report. By definition, it follows that the probability of event E, denoted $\mathbb{P}\{E\}$, can be computed as*

$$\mathbb{P}\{E\} = \mathbb{P}\{F\} \cdot \mathbb{P}\{A \mid F\} + \mathbb{P}\{\overline{F}\} \cdot \mathbb{P}\{\overline{A} \mid \overline{F}\} \tag{3.1}$$

where $\overline{X}$ is defined as the complement of event X. Thus, $1 - \mathbb{P}\{E\}$ represents the classification accuracy of the smartphone crowdsensing system, and will henceforth be used to evaluate its performance.

The proposed FIRST framework will provide the mathematical tools to relate the number m of MTPs to the error probability $\mathbb{P}\{E\}$ and the mobility of users.

Let ϵ^{max} be the desired maximum classification error probability. The MTP optimization problem (MOP) is then defined as follows.

Definition 3. *MTP Optimization Problem.* Minimize m such that $0 \leq \mathbb{P}\{E\} \leq \epsilon^{max}$

3.4. THE FIRST FRAMEWORK

Figure 3.3 introduces the FIRST framework, which is made up by three components:

- Likelihood Estimation Algorithm (LEA): It provides an approximation of the mobility of users and MTPs. LEA is based on an *image processing* technique that produces an approximate likelihood based *only* on geographical information (i.e., the map of the sensing area).
- Computation of Validation Probability (CVP): This component derives the probability $\mathbb{P}\{V\}$ of the event V that a sensing report will be validated by at least one MTP, as a function of the number of MTPs deployed and the approximate mobility produced by the LEA.
- MTP Optimization Algorithm (MOA): It takes $\mathbb{P}\{V\}$ and computes $\mathbb{P}\{E\}$, so as to provide a solution to the MOP to achieve desired maximum error ϵ^{max}.

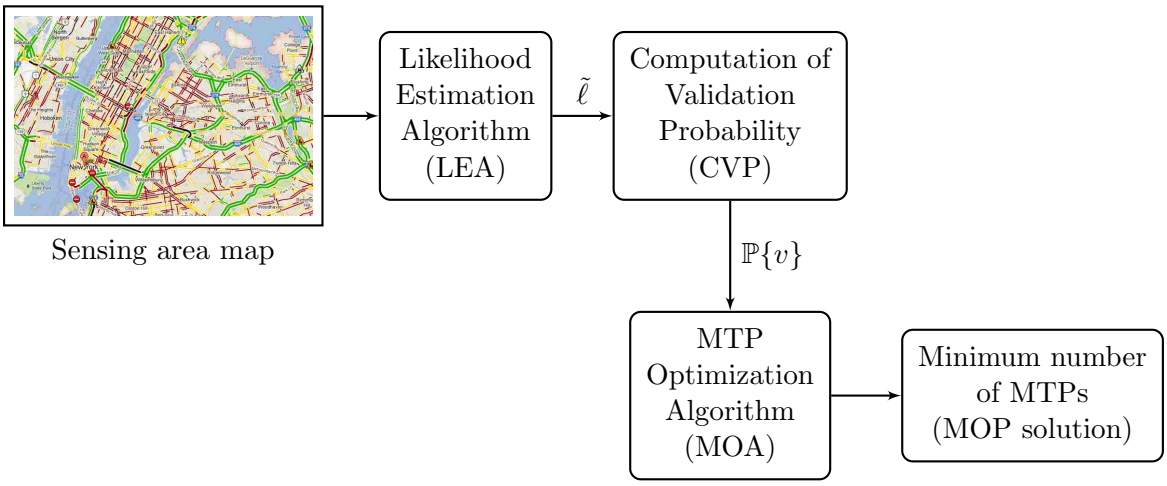


Figure 3.3. Block scheme of the proposed FIRST framework.

3.4.1. Computation of Validation Probability. In this section, the probability $\mathbb{P}\{V\}$ of the event that a sensing report will be validated by at least one MTP is derived. Let $\mathcal{Q}$ be the set of MTPs competing for offering their sensing services, and $\mathcal{U}$ be the set of users of the application. Let $u(i, z, t)$ be the distribution over the sector set $\mathcal{S}$ of the random variable (r.v.) U_z^t describing the location of user z at time

t. Let also $q(i, z, t)$ be the distribution over the sector set $\mathcal{S}$ of the random variable (r.v.) Q_z^t describing the location of MTP z at time t.

Let us calculate the probability $\mathbb{P}\{V_z\}$ that a sensing report coming from user u_z is verified by an MTP, conditioned to the fact that user u_z is currently in sector s_i of the sensing area:

$$\mathbb{P}\{V_z \mid U_t^z = s_i\} = 1 - \prod_{k \in \mathcal{Q}} (1 - q(i, k, t)) \tag{3.2}$$

In the above equation, it is assumed that the mobility of each MTP is independent, which is sound because it is highly unlikely MTPs would influence each other's mobility in any way. The above equation can be explained as follows. The probability that a sensing report is verified is the complement of the probability that no MTP is in the same sector as the user. The probability that a sensing report is verified, irrespective of the location of the user, can thus be computed by using the theorem of total probability, i.e.,

$$\mathbb{P}\{V_z\} = \sum_{i=1}^{n} \mathbb{P}\{V_z \mid U_t^z = s_i\} \cdot u(i, z, t) \tag{3.3}$$

The probability $\mathbb{P}\{V\}$ that on the average a sensing report will be validated can be computed as the average $\mathbb{P}\{V_z\}$ over all the users, which is

$$\mathbb{P}\{V\} = \frac{1}{|\mathcal{U}|} \sum_{z=1}^{|\mathcal{U}|} \mathbb{P}\{V_z\} \tag{3.4}$$

Example. Figure 3.4 shows two sensing areas (S_1 and S_2) divided into the same number $n = 8$ of sectors. It is assumed that a total of $m = 5$ MTPs are present. For simplicity, in this example it is assumed that the mobilities of users and MTPs follow the same distribution.

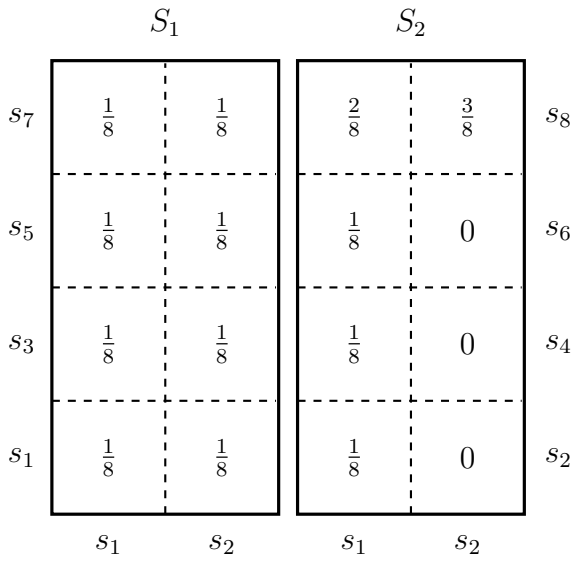


Figure 3.4. Example to illustrate computation of $\mathbb{P}\{V\}$.

For simplicity, let us define as ℓ_i^j as the probability that an MTP will be in sensing area j and sector i. The corresponding mobility distributions ℓ_i^1 and ℓ_i^2 are given as: $\ell_i^1 = 1/8$ for $1 \leq i \leq 8$, while

$$\ell_i^2 = \begin{cases} \frac{1}{8} & i = 1, 3, 5 \\ \frac{2}{8} & i = 7 \\ \frac{3}{8} & i = 8 \\ 0 & i = 2, 4, 6 \end{cases} \tag{3.5}$$

Let us compute $\mathbb{P}\{V\}$ for both sensing areas. First, we need to compute $\mathbb{P}\{V \mid U = s_i\}$ for each s_i, which is

- $S_1 : \mathbb{P}\{V \mid U = s_i\} = 1 - (1 - 1/8)^5 = 0.49$ for every i, since ℓ_i is equal for each sector. Therefore, $\mathbb{P}\{V\} = 1/8 \cdot 8 \cdot 0.49 = 0.49$.
- $S_2 : \mathbb{P}\{V \mid U = s_1\} = \mathbb{P}\{V \mid L = s_3\} = \mathbb{P}\{V \mid U = s_5\} = 1 - (1 - 1/8)^5 = 0.49$. $\mathbb{P}\{V \mid U = s_7\} = 1 - (1 - 2/8)^5 = 0.76$, $\mathbb{P}\{V \mid U = s_8\} = 1 - (1 - 3/8)^5 = 0.90$,

$\mathbb{P}\{V \mid U = s_2\} = \mathbb{P}\{V \mid U = s_4\} = \mathbb{P}\{V \mid U = s_6\} = 0$. Therefore, $\mathbb{P}\{V\} = {}^3\!/\!_8 \cdot 0.49 + {}^2\!/\!_8 \cdot 0.76 + {}^3\!/\!_8 \cdot 0.90 = 0.71$.

The above example suggests that the likelihood that some sectors will be more occupied than others significantly impacts on the report validation probability. Indeed, if the mobility of MTPs and participants is more concentrated, the validation probability will increase with respect to the case when the mobility is uniform.

3.4.2. Bounding Mobility. Estimating the mobility distributions u and q is paramount to compute $\mathbb{P}\{V\}$ and therefore, provide a cost-efficient solution to the MOP. In cases where information about the mobility of users and MTPs is available, for example, mobility traces of MTPs and users are available, an exact computation of u and q may be used. However, prior information about MTP and user mobility may not always be available. Nevertheless, it is need necessary to analyze at least the worst-case scenario of uncertainty and solve the MOP in any case. In this study, the well-known concept of information entropy is used to this purpose.

The entropy $H(L)$ of a random variable (r.v.) L having distribution $\ell_i \triangleq \ell(i) = \mathbb{P}\{L = i\}, 1 \leq i \leq n$, is $H(L) = -\sum_{i=1}^{n} \ell_i \log_2 \ell_i$. If the distributions q and u cannot be estimated otherwise, the theorem of maximum entropy [39] states that, without having any information on the phenomenon, we need to consider the distribution having maximum entropy, which it is known is the uniform distribution.

This theoretical result is particularly interesting, because the worst case performance can be studied regardless of the availability of mobility information. However, in order to obtain better optimization results, it is necessary to find a *tighter bound* on the mobility of users. This is because, if the mobility of the users is more restricted, fewer MTPs will be needed to provide the same information reliability level, which yields a better optimization result.

In this study, a heuristic *Likelihood Estimation Algorithm* (LEA) was designed, to provide a tighter bound on the mobility of users and MTPs, with just knowing the

sensing area location. This heuristic is based on the following rationale: the smartphone crowdsensing systems under consideration are deployed in cities, or anyway close to urban areas. This implies that the mobility of users and MTPs will be likely to be almost restricted to the main arterial roads of the sensing areas, or anyway the zones/roads with the greater amount of traffic (both pedestrian and vehicular). By restricting the possible area of movement of the MTPs and users, the randomness of the movement of users and MTPs can be reduced, and therefore, a tighter bound on the likelihood of sectors can be provided.

To demonstrate this point, Figure 3.5(a) shows the heatmap of the mobility traces[†] of taxi cabs in a section of Downtown San Francisco, where the intensity of the color indicates the popularity of the place. As the figure points out, the mobility of taxi cabs is definitely not uniform, and mostly concentrated on a few popular places. Furthermore, Figure 3.5(b) shows the main arterial roads provided by Google Maps APIs[‡]. From this figure, it emerges that the roads point out (with some degree of approximation) the most popular places as shown in the heatmap of Figure 3.5(a).

Let us now describe the LEA algorithm which works as follows. Let us consider the map M of the sensing area, and divide it into n sectors as required by the application, where $\mathcal{S} = \{s_1, \cdots, s_n\}$ is the set of sectors. Then, information about the most popular places (which may be roads/squares/buildings) and the geographical constraints of the sensing area is acquired.

By using Google Maps APIs[§], the main arterial roads on a specific location area are highlighted. This information is leveraged to mark such places in the map M, the background of which is further removed to get a black-and-white image of the

[†]Published in [72], available at `http://www.crawdad.org`

[‡]APIs publicly available at `https://developers.google.com/maps/documentation/javascript/styling`

[§]Other approaches, such as Open Street Maps (`https://www.openstreetmaps.org`), could be also used for such purpose.

sensing area as shown in Figure 3.5(b), where the black pixels represent the popular places.

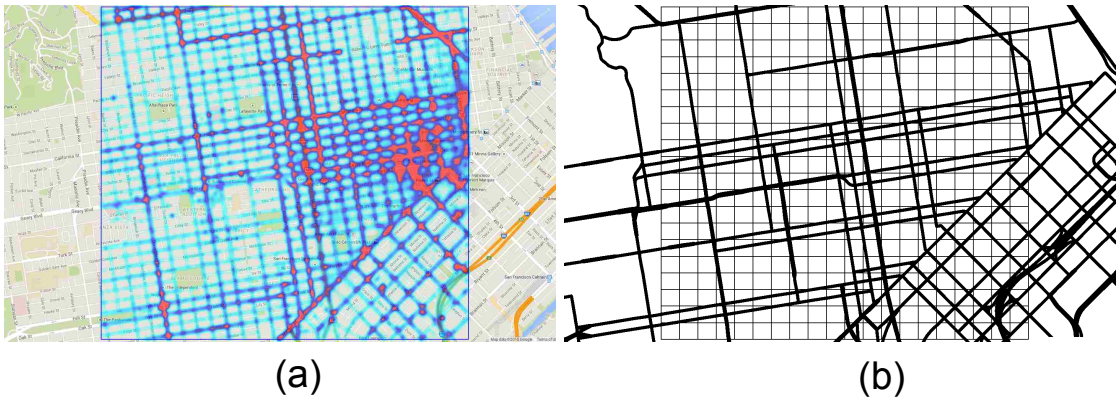


Figure 3.5. (a) Heatmap of mobility traces vs. (b) Arterial roads.

The LEA is described by the pseudo-code in Algorithm 1. In Section 3.5, it is shown that the LEA is remarkably effective in approximating the mobility distribution of users in various settings, by using real-world mobility traces collected in three major cities in three different continents, namely Rome, San Francisco and Beijing.

The implicit assumptions that LEA makes are **(i)** the mobility of users and MTPs is stationary (i.e., does not change over time); and **(ii)** users and MTPs follow the same mobility distributions. Although these are pretty strong assumptions, in the experimental evaluation conducted in Section 3.5 it is shown that LEA provides a pretty good approximation of the likelihood of the sectors, considering that only information only from a map are used.

Indeed, LEA is not a fine-grained mobility estimation algorithm. Instead, it is a simple heuristic that provides before deployment an approximate information regarding the likelihood of certain sectors with respect to others. If more reliable information about the mobility is known, it could be used to complement LEA's analysis and achieve better optimization results.

Algorithm 1 Likelihood Estimation Algorithm (LEA)

Input: M, map of the sensing area
Output: $\tilde{\ell}$, approximate distribution of mobility
1: $\mathcal{S} \leftarrow$ set of sectors $s_1 \cdots s_n$
2: $I \leftarrow$ processed image with most popular areas
3: $B \leftarrow 0$ (sum of black pixels in sensing area)
4: **for** each sector $s_i \in \mathcal{S}$ **do**
5: $\quad B_i \leftarrow$ number of black pixels $\in s_i$
6: $\quad B \leftarrow B + B_i$
7: **end for**
8: **for** each $s_i \in \mathcal{S}$ **do**
9: $\quad \tilde{\ell}(i) \leftarrow {}^{B_i}/_{B}$
10: **end for**
11: **return** $\tilde{\ell}$

3.4.3. Solving the MTP Optimization Problem. This section describes the methodology adopted by FIRST to solve the MTP Optimization Problem (MOP) defined in Section 3.3. In order to solve the MOP, it is needed to compute the error probability $\mathbb{P}\{E\}$. This implies $\mathbb{P}\{A \mid F\}$ and $\mathbb{P}\{\overline{A} \mid \overline{F}\}$, defined in Equation (3.1), must be derived as a function of $P\{V\}$.

Here FIRST solves the MOP by providing the mathematical tools that relate the number m of MTPs to the error probability $\mathbb{P}\{E\}$ and user mobility. In Equation (3.4), it is shown how to compute $\mathbb{P}\{V\}$ given q and u. By applying probability theory, it is obtained $\mathbb{P}\{A \cap F\} = \mathbb{P}\{F\} \cdot \mathbb{P}\{\overline{V}\} \cdot \mathbb{P}\{A \mid \overline{V}\}$ and $\mathbb{P}\{A \cap \overline{F}_j\} = \mathbb{P}\{\overline{F}_j\} \cdot (\mathbb{P}\{V\} + \mathbb{P}\{\overline{V}\} \cdot \mathbb{P}\{A \mid \overline{V}\})$.

The only unknown in $\mathbb{P}\{A \cap F\}$ and $\mathbb{P}\{A \cap \overline{F}\}$ is $\mathbb{P}\{A \mid \overline{V}\}$, which is, the probability of deeming a sensing report reliable in the case it has not been validated by an MTP. Ideally, this probability should be close to 1 when the report being sent is reliable, and close to 0 when the report being sent is not reliable. To this end, FIRST leverages the knowledge provided by the reports submitted by the MTPs,

and computes $\mathbb{P}\{A \mid \overline{V}\}$ as follows.

$$\mathbb{P}\{A \mid \overline{V}\} = \mathbb{P}\{V\} \cdot \mathbb{P}\{\overline{F}\} + \frac{1}{2} \cdot \mathbb{P}\{\overline{V}\} \tag{3.6}$$

This formula can be explained as follows. The first part, $\mathbb{P}\{V\} \cdot \mathbb{P}\{\overline{F}\}$, represents the "degree of belief" we have in the users; it is higher when the user is validated most of the time ($\mathbb{P}\{V\}$ close to 1) and the reports are reliable. The second part, $1/2 \cdot \mathbb{P}\{\overline{V}\}$, represents the "degree of uncertainty" in the users; it is higher when most of the reports have not been validated. Note that, as $\mathbb{P}\{V\}$ increases, the value of $\mathbb{P}\{A \mid \overline{V}\}$ approximates to $\mathbb{P}\{\overline{F}\}$. Also, if $\mathbb{P}\{V\} = 0$ (i.e., no MTPs are present), the system deems as reliable every report with probability $1/2$ (coin tossing), which is sound as there is no reason to be more inclined to accept or reject the report if no information is available.

It is now presented an algorithm to solve the MOP, called the MOP Optimization Algorithm (MOA). The MOA is based on a modified version of binary search algorithm, called Left-most Insertion Point (LMIP). More specifically, LMIP returns the left-most place (i.e., the minimum value) where $\mathbb{P}\{E\}$ can be correctly inserted (and still maintains the sorted order) in the ordered array of the errors corresponding to a particular choice of m. This corresponds to the lower (inclusive) bound of the range of elements that are equal to the given value (if any). Note that LMIP can be applied to solve the MOP due to the fact that $\mathbb{P}\{E\}$ is a monotonically decreasing function of m (demonstration has not been reported here due to space limitations).

The MOA takes as input the approximate distribution $\tilde{\ell}_i$ provided by LEA (equal for participants and MTPs), and also $\mathbb{P}\{F\}$, the desired maximum error ϵ^{max}, and the maximum number m^{max} of MTPs available. It provides as output the optimum number m^* of MTPs to be used to achieve the desired maximum error ϵ^{max}.

Algorithm 2 MOP Optimization Algorithm (MOA)

Input: $\tilde{\ell}_i$, $\mathbb{P}\{F\}$, ϵ^{max}, m^{max}
Output: m^*
1: $\epsilon^{min} \leftarrow$ `CalculateError`$(\tilde{\ell}_i, \mathbb{P}\{F\}, m^{max})$
2: **if** $\epsilon^{max} < \epsilon^{min}$ **then**
3: **return** 'infeasible'
4: **end if**
5: **return** `LMIP`$(\tilde{\ell}_i, \mathbb{P}\{F\}, \epsilon^{max}, 0, m^{max})$

In lines 1-3, the MOA checks with the procedure `CalculateError` (implementing Equation 3.1) whether the minimum error ϵ^{min} obtained with the maximum number of MTPs available is greater than the desired maximum error ϵ^{max}. If this is the case, then the MOA has no feasible solutions and therefore the algorithm terminates immediately. If not, then the routine `LMIP` is invoked, which finds m^* by implementing the LMIP algorithm.

Let us calculate the time complexity of the MOA. `LMIP` is a variation of binary search, therefore its overall complexity will be $O(x \cdot \log m^{max})$, where x is the complexity of `CalculateError`. Such complexity is $\Theta(n)$, given it requires constant time to compute $\mathbb{P}\{E\}$ using Equation (3.1) and n iterations to compute $\mathbb{P}\{V\}$ using Equation (3.4), where n is the number of sectors. Therefore, the overall time complexity of MOA is given by $O((n \cdot \log m^{max})$.

Example 3. In the example of Figure 3.6, it is assumed the ℓ distribution equal to ℓ_i^2 presented in Figure 3.4, $\mathbb{P}\{F\} = 0.01$, $m^{max} = 8$ and $\epsilon^{max} = 0.1$. In this case, the LMIP will return $m^* = 4$, since it is the left-most element that provides $\mathbb{P}\{E\} \leq 0.1$.

3.4.4. Practical Implementation. In this section, it is described how the system, *after deployment*, handles the case in which a report has not been validated by MTPs (i.e., how $\mathbb{P}\{A \mid \overline{V}\}$ is actually computed). For each user u_i, the system

$\mathbb{P}\{E\}$	0.29	0.17	0.11	0.07	0.05	0.03	0.02	0.02
m	1	2	3	4	5	6	7	8

Figure 3.6. Example of `LMIP`.

keeps track of the number k_i of sensing reports submitted, the number k_i^v of sensing reports validated by an MTP, and the number k_i^r of reports that have been validated as reliable.

As soon as a report q is sent by user u_i, if the report has not been validated by an MTP, then the report is classified as reliable with probability

$$\mathbb{P}\{A \mid \overline{V}\} = \frac{k_i^r}{k_i} + \frac{1}{2} \cdot \left(1 - \frac{k_i^v}{k_i}\right) \tag{3.7}$$

After being classified as reliable, reports may be subsequently analyzed by additional algorithms (for example, [60, 61]) to determine the actual status of the sensing area by combining or fusing the information conveyed by the reliable reports.

3.5. EXPERIMENTAL RESULTS

In this section, the experimental results obtained by evaluating the performances of FIRST and comparing it with relevant related work are presented. First, it is reported the performance results obtained by considering an application monitoring vehicular traffic events. Then, results obtained by using the Participatory PerCom application are discussed.

3.5.1. Participatory Traffic Sensing. To implement this experiment, the mobility traces collected from the following datasets were considered:

- *CRAWDAD-SanFrancisco* [72]: This dataset contains mobility traces of approximately 500 taxis in San Francisco, USA, collected over one month's time;

- *CRAWDAD-Rome* [4]: In this dataset, 320 taxi drivers in the center of Rome were monitored during March 2014;
- *MSR-Beijing* [91]: This dataset collected by Microsoft Research Asia contains the GPS positions of 10,357 taxis in Beijing during one month.

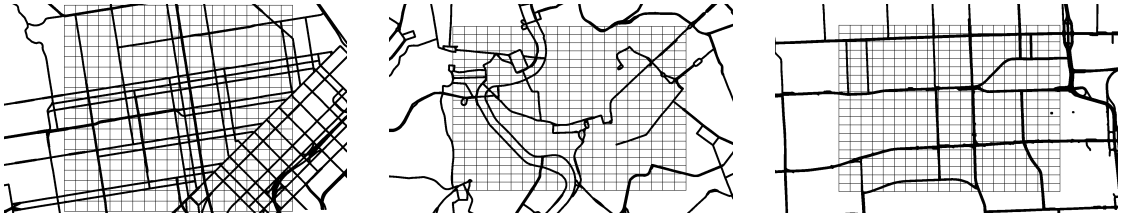
In these experiments, it is considered a traffic sensing application in which taxi cab drivers report traffic anomalies. Consistent with the example mentioned in Section 3.1, it is assumed the reports are divided into 4 categories such as "Car Crash", "Road Closure", "Traffic Jam", "No Event". Furthermore, sensing areas of approximately 4×4km square areas are considered, which characterize the downtown of cities such as San Francisco, Rome, and Beijing. In the chosen scenario, the taxi cabs report every 5 minutes information about their surroundings to the SCP. The application was implemented using the OMNeT++ simulator[¶].

3.5.1.1. Evaluation of FIRST components. The goal of the first set of experiments is to test the efficacy of LEA in computing the likelihood of sectors. To obtain ground-truth information about the actual mobility of taxi cabs, the traces were processed using OMNeT++. It is assumed that such mobility is unknown, and the LEA algorithm was applied to the chosen sensing areas to approximate the mobility. To apply LEA, the sensing area was divided into a grid of 20×20 sectors, with sectors having the same size as a city block. In Figure 4.2, it is shown the maps of the sensing areas after the processing of LEA.

Figure 3.8 shows the distribution of the likelihood of sectors and the one obtained by LEA, respectively. More specifically, the figure shows the actual and estimated probability of a taxi to be in each sector of the sensing area. These experiments conclude that the LEA algorithm approximates well the likelihood of sectors, considering the scarce information available. This result is extremely significant, as it is necessary to provide very precise estimation of the classification accuracy of FIRST

[¶]Available at `https://www.omnetpp.org`

as a function of the number of MTPs. Figure 3.9 shows $\mathbb{P}\{E\}$ as a function of the number of MTPs, calculated analytically by the Computation of Validation Probability (CVP) component of FIRST. For comparison purposes, CVP is evaluated by providing as input (i) the distribution computed by LEA as applied to each considered sensing area (*CVP-LEA*, represented by a dashed line), and (ii) the uniform mobility distribution (*CVP-Uniform*, represented by a dotted continuous line) as the baseline approximation. Such analytical results are compared with the experiments using the traffic datasets.



(a) San Francisco (b) Rome (c) Beijing

Figure 3.7. Maps of the sensing areas after the processing of LEA.

In Figure 3.10, the MTP Optimization Algorithm (MOA) is applied to analyze the number of MTPs that are necessary by FIRST to provide maximum desired error probability ϵ^{max}. Similarly to the experiments shown in Figure 3.9, users sending unreliable reports with three probability values $\mathbb{P}\{F\} = 0.01$, 0.5 and 0.9 are considered. Figure 3.10 confirms that the San Francisco setting requires the highest number of MTPs to achieve given ϵ^{max}. These results also highlight that FIRST is remarkably effective in achieving high accuracy with a low number of MTPs. More specifically, it provides on the average 85% of accuracy with an MTPs per sector density of about 32% in case of Rome and Beijing, and 55% in the case of San Francisco. Note that higher accuracy values require in general a significant number of MTPs, especially

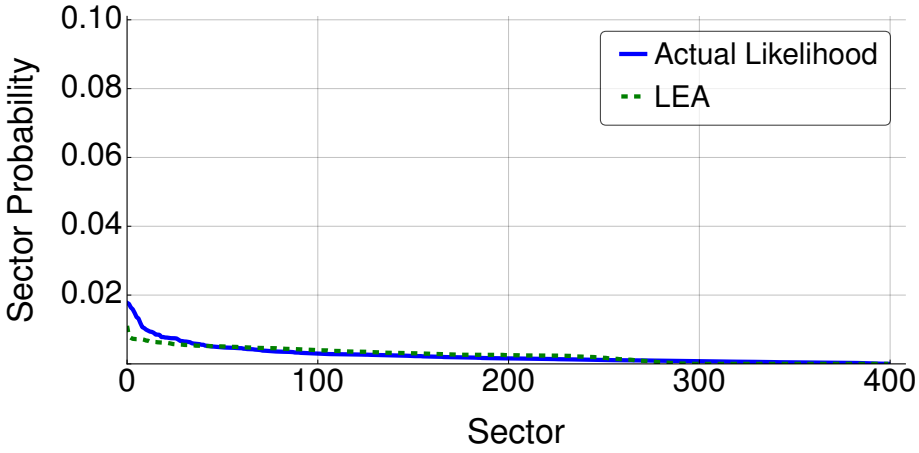
when the behavior of participants becomes hardly predictable (i.e., $\mathbb{P}\{F\} = 0.5$) and the mobility is highly entropic (i.e., in San Francisco setting).

As shown in Figure 3.9, in all three scenarios, *CVP-LEA* computes $\mathbb{P}\{E\}$ with remarkable precision. In particular, the maximum difference obtained is 3.47%, achieved in the Rome setting. Furthermore, Figure 3.9 shows that the accurate estimation of the mobility provided by the LEA translates into an improved prediction accuracy of CVP with respect to the uniform distribution, as *CVP-Uniform* yields a maximum difference of 17.02% in the case of Rome setting. Figure 3.9 highlights that the San Francisco setting requires the largest number of MTPs to achieve a specified maximum error probability.

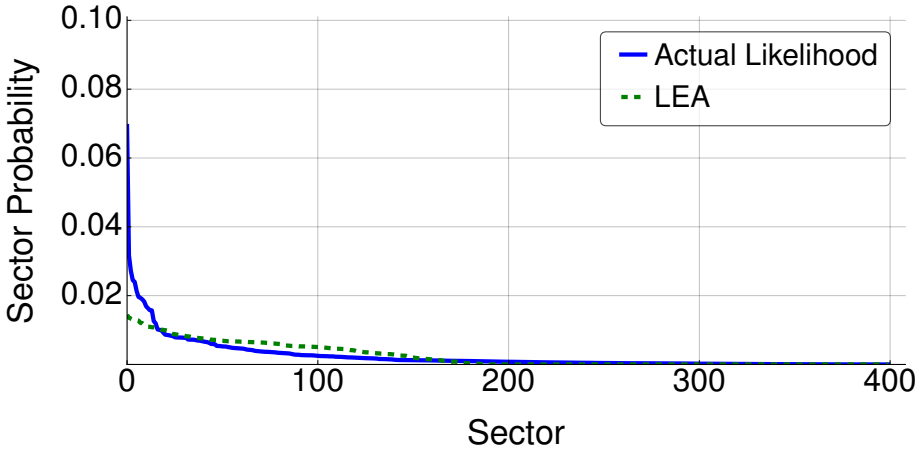
In these cases, only a high number of MTPs can guarantee that a sufficient number of reports are validated and therefore, desired accuracy may be provided. Indeed, Figure 3.10 also shows that fewer MTPs are needed when $\mathbb{P}\{F\} = 0.9$ than when $\mathbb{P}\{F\} = 0.5$.

Intuitively, this is due to the fact that, when participants send reports randomly, it is more difficult to understand their reliability. On the other hand, when their behavior is more "regular" (i.e., consistent over time) it is easier to evaluate their reliability. This intuition is also confirmed by Figure 3.11, which depicts the number of MTPs needed as a function of $\mathbb{P}\{F\}$, for three values of desired maximum error probability ϵ^{max}. Figure 3.11 shows that, irrespective of ϵ^{max}, the highest number of MTPs is necessary when $\mathbb{P}\{F\} = 0.5$.

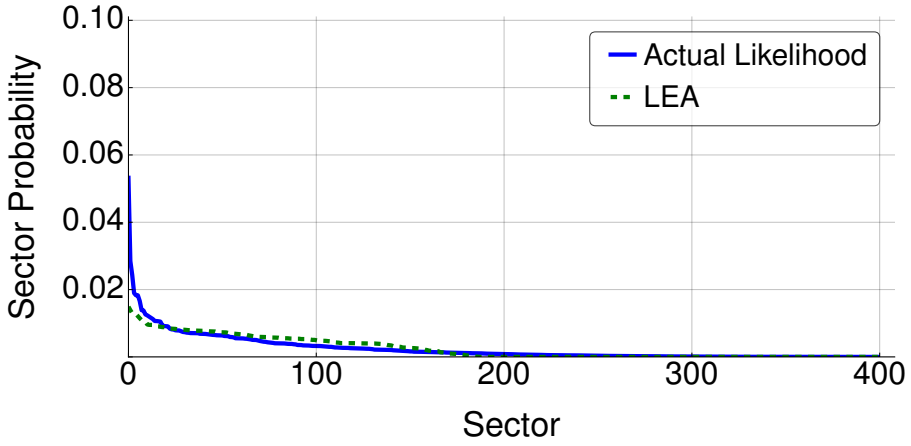
3.5.1.2. Evaluation of attack resiliency. Based on the behavior models defined in Section 3.1, the following security attacks are taken into acocunt, which were defined in other domains and recently cast in the context of smartphone crowdsensing [64]. For simplicity, hereafter we will generically use the word "attacker" for both malicious and unreliable users, and the words "threat" and "attack" interchangeably.



(a) San Francisco

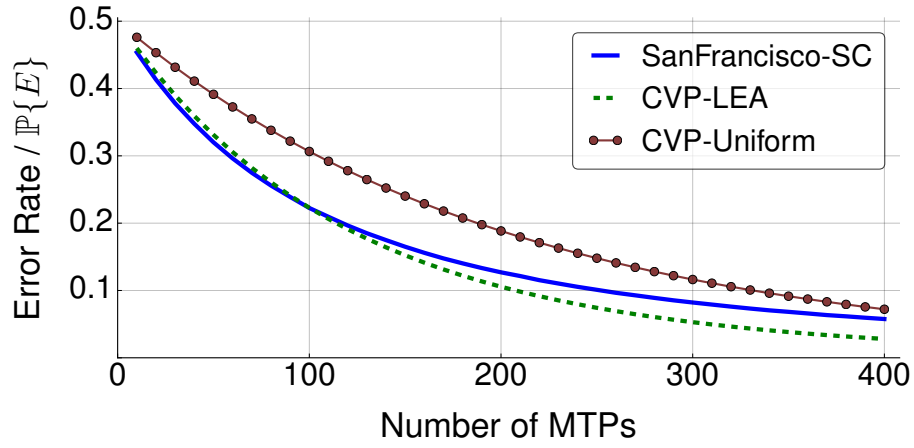


(b) Rome

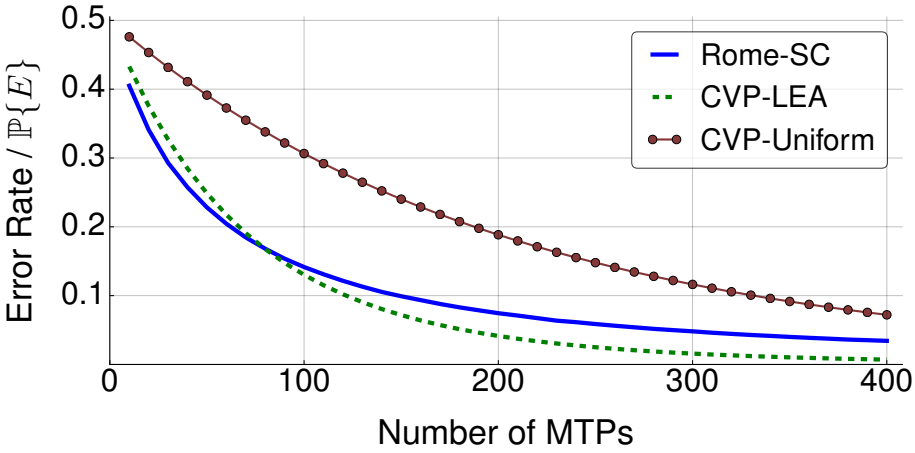


(c) Beijing

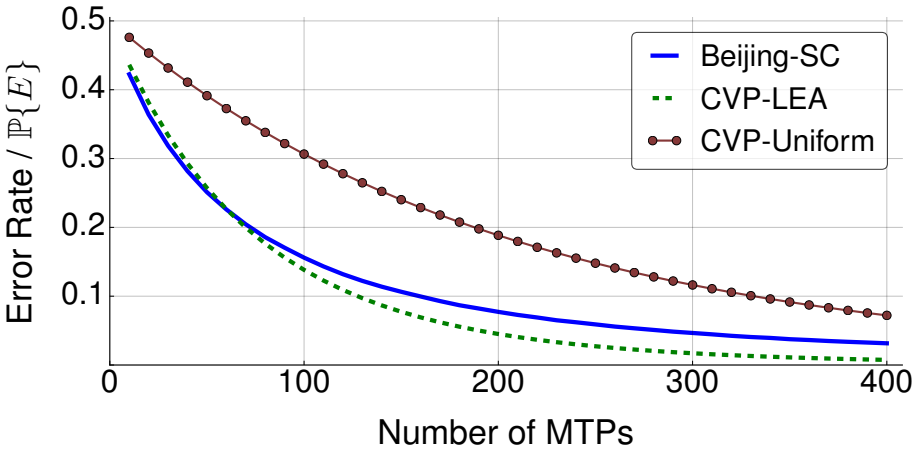
Figure 3.8. Traces vs. Mobility Estimation Algorithm.



(a) San Francisco

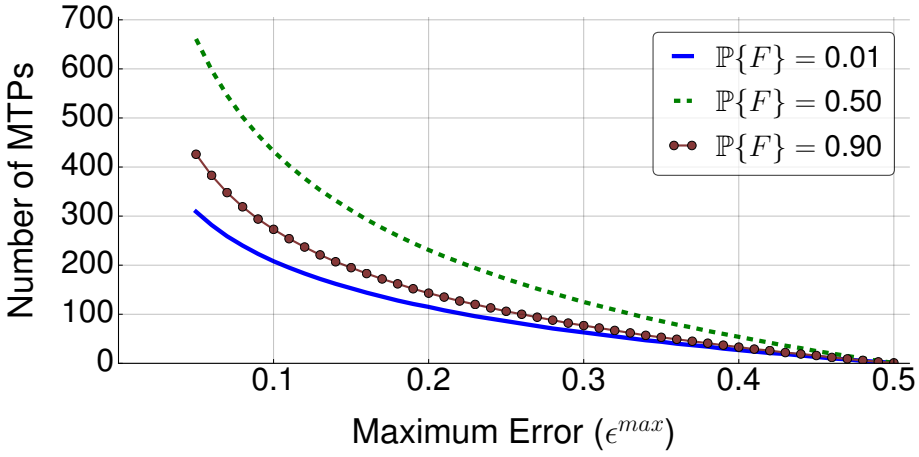


(b) Rome

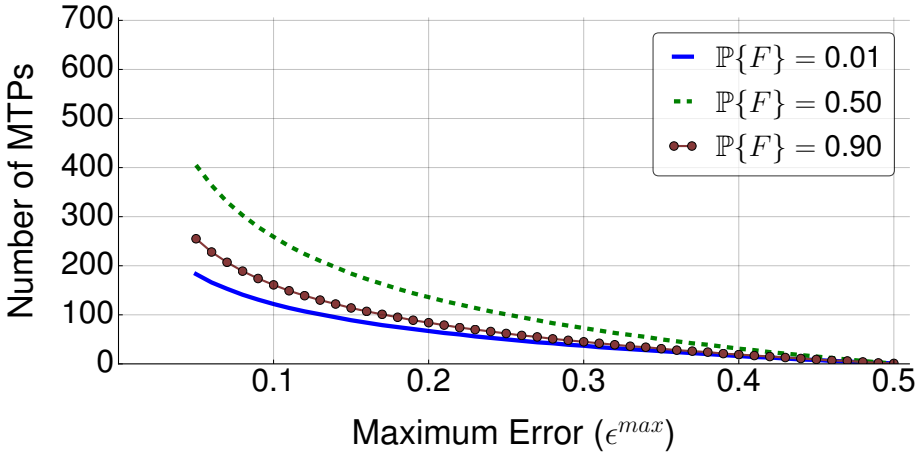


(c) Beijing

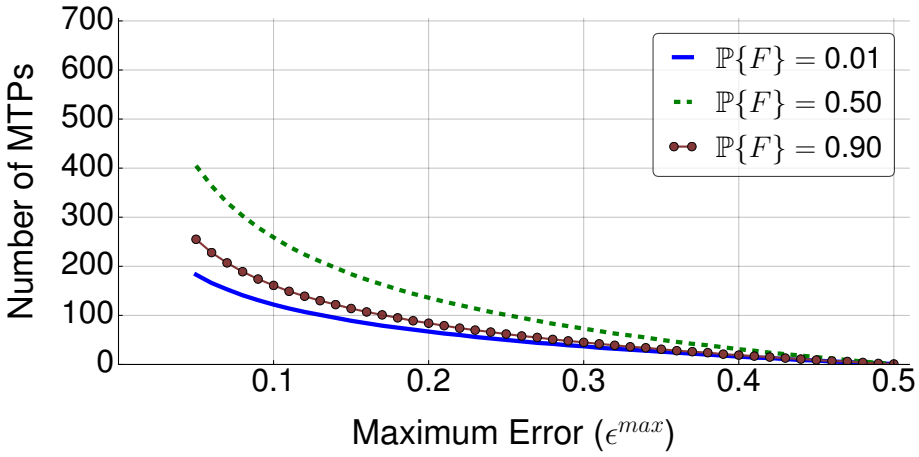
Figure 3.9. Number of MTPs vs Error rate / $\mathbb{P}\{E\}$.



(a) San Francisco



(b) Rome



(c) Beijing

Figure 3.10. MOA results.

1. *Corruption attack*. This threat models the following strategy: for each sensing report, the attacker sends unreliable data with probability p and correct data with probability $1-p$. This attack can be carried out by unreliable and malicious users alike.
2. *On-off attack*. In this attack, the malicious user alternates between normal and abnormal behaviors to conceal her maliciousness. Specifically, the adversary *periodically* sends n reliable reports and then m unreliable reports, and then repeats the process. This attack is extremely easy to carry out but also extremely challenging to detect and contrast [3, 14, 71].
3. *Collusion attack*. In this attack, two or more malicious participants coordinate their behavior in order to provide the same (unreliable) information to the SCP [35, 57]. The malicious behavior may also include GPS location spoofing, so as to mislead the SCP into assuming colluding participants are nearby [74].

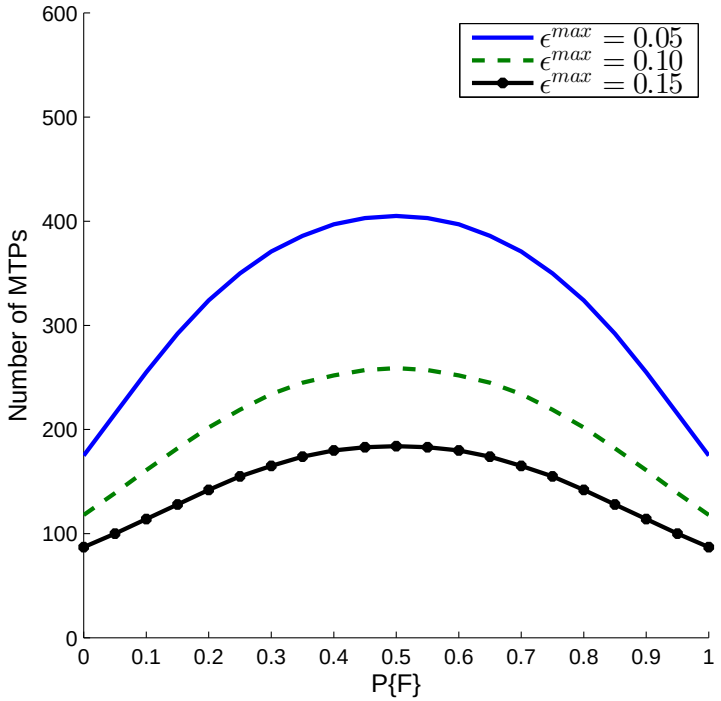


Figure 3.11. Number of MTPs vs. $\mathbb{P}\{F\}$ in case of MOA applied to Rome setting.

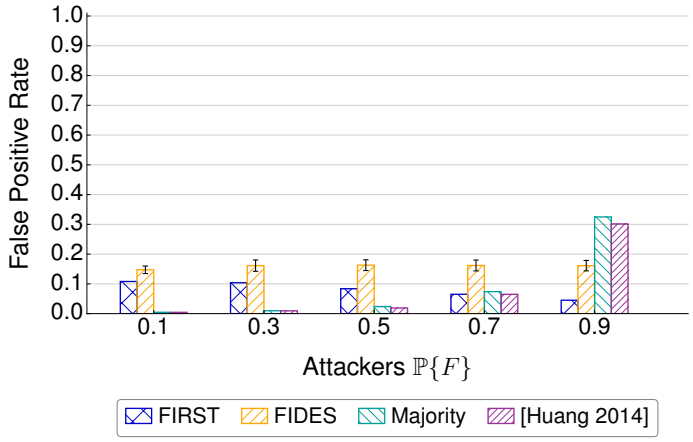
FIRST is compared with the FIDES framework [74] and the reputation-based framework proposed in [37], [Huang 2014]. FIDES uses a modified version of Jøsang's trust model to update the reputation of users. This framework inherits from Jøsang's trust model a strong sensitivity to parameter tuning. On the other hand, [Huang 2014] proposes an approach which is a improved variation of majority vote, and its performance also depends on the choice of parameter setting (Gompertz's function's, among others). For implementation, it is used the parameter settings proposed in the papers, which are reported in Table 3.1. A pure majority vote scheme was also implemented to obtain baseline performance. If not stated otherwise, in the following experiments the parameters reported in Table 3.1 are used. Confidence intervals at 95% are shown only when above 1% of the value.

Table 3.1. Experimental parameters.

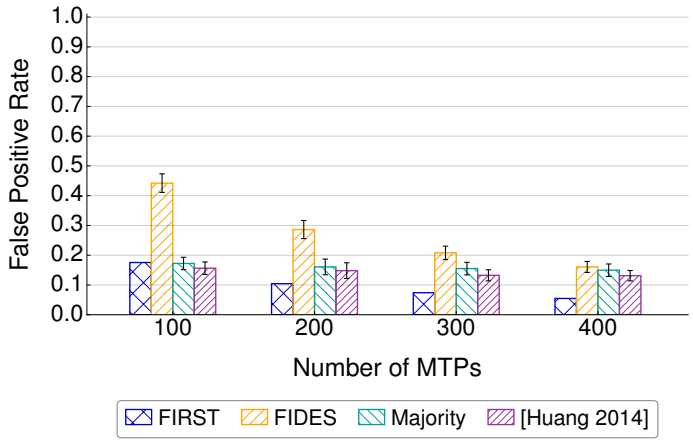
Parameter	**Value**
Experiment length	240mins
Timestep length	5mins
Location	Rome
Number of users	1000
$\mathbb{P}\{F\}$ for non attackers	0.01
a_r (FIDES)	0.7
a_u (FIDES)	0.9
Initial reputation (FIDES)	0.5
Reputation threshold (FIDES)	0.75
Initial weights (FIDES)	[1, 0, 0]
λ_s (Huang)	0.7
λ_p (Huang)	0.8
a (Huang)	1
b (Huang)	-2.5
c (Huang)	-0.85
Initial reputation (Huang)	0.5
Reputation threshold (Huang)	0.5
Number of MTPs (FIRST, FIDES)	400
Number of attackers	500
Attackers $\mathbb{P}\{F\}$	0.8
On-off steps	(10, 10)
Collusion groups	3

Figure 3.12 reports the false positive rate (percentage of false reports accepted w.r.t. the total number of reports accepted) obtained by the frameworks when subject to a corruption attack, as a function of the (constant) attack probability, number of MTPs (applies only to FIDES and FIRST), and number of attackers. Figure 3.12(a) and (c) show that the performance of Majority and [Huang 2014] decreases as the number of false reports and attackers increases. This is reasonable, as both schemes are based on data aggregation and therefore not resilient to large number of malicious users and/or unreliable reports. Figure 3.12(a) Furthermore, Figure 3.13 shows the results obtained under the On-off attack by all the considered schemes. As expected, the performance of FIRST is slightly affected by this attack, especially when the percentage of ON steps is less than the OFF one. This is because, the less the ON steps are, the harder it is for FIRST to decrease the accept probability of malicious users. However, FIRST is able to achieve a FPR of about 10% in the worst case of ON=5. On the other side, [Huang 2014] and Majority are instead more affected when the ON step is greater than the OFF, as it is more likely for them to misclassify sensing reports when the percentage of unreliable reports/number of attackers is higher.

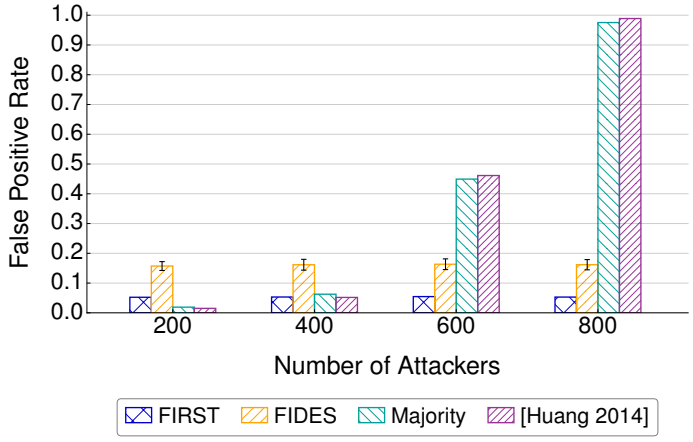
Figure 3.14 shows the results obtained by running the Collusion attack. The experiment has been implemented as follows. We have assumed there are k collusion groups. An attacker belonging to the k-th group coordinates with the other attackers belonging to the same group by implementing together an On-off attack. In such attack, during the ON phase the attackers send false reports pertaining to a chosen sector, the same for every user in the k-th group. The results conclude that [Huang 2014] and Majority are severely affected by this attack, while FIRST tolerates well this attack by keeping the FPR below 10% by using 400 MTPS, regardless of the number of attackers and collusion groups considered. This is because FIRST uses MTPs to validate data and does not rely on data aggregation. Interestingly enough, [Huang 2014] and Majority perform slightly well when the collusion groups are more. This is



(a)

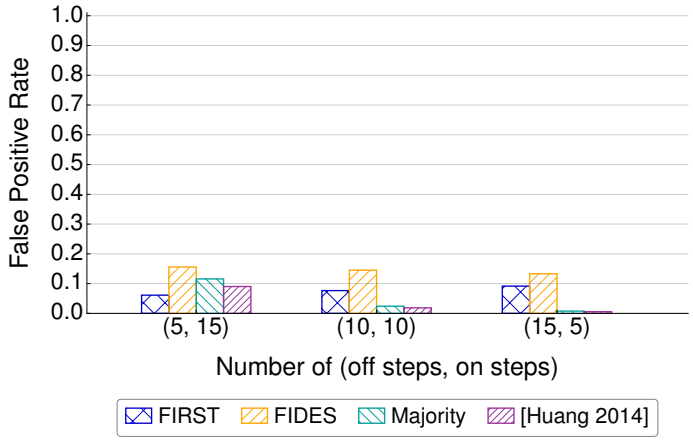


(b)

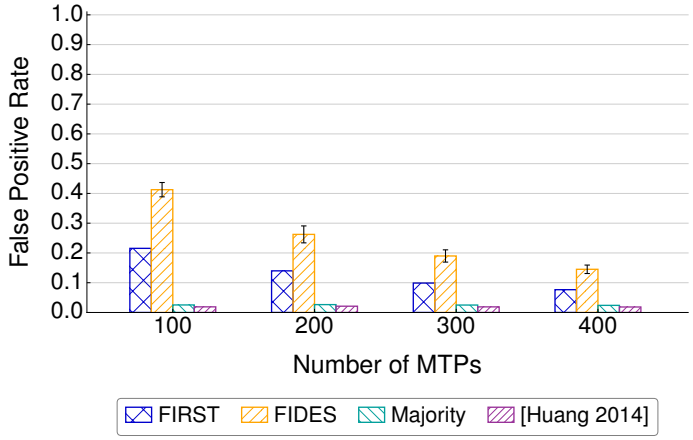


(c)

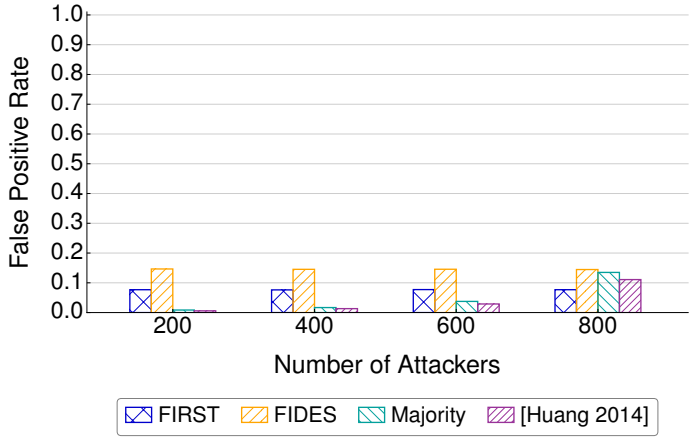
Figure 3.12. Corruption attack: False Positive Rate vs. $\mathbb{P}\{F\}$, MTPs, and attackers.



(a)

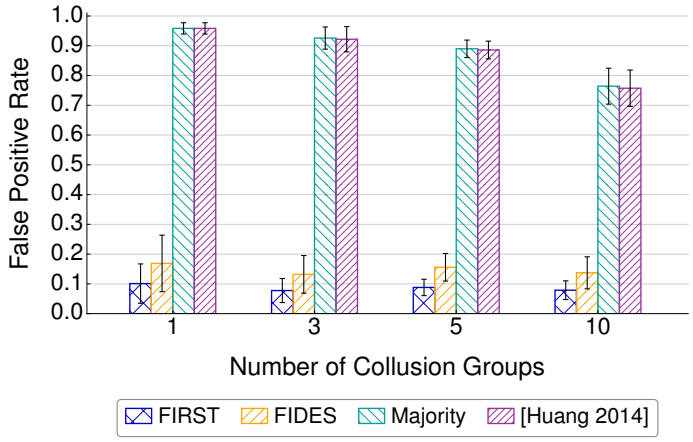


(b)

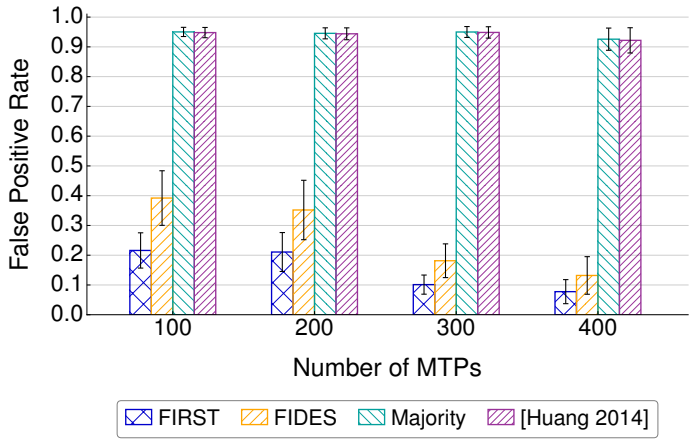


(c)

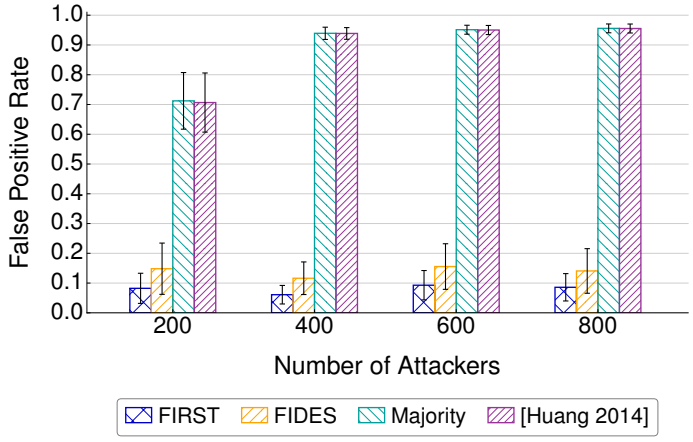
Figure 3.13. On-off attack: False Positive Rate vs. On-off steps, MTPs, and attackers.



(a)



(b)



(c)

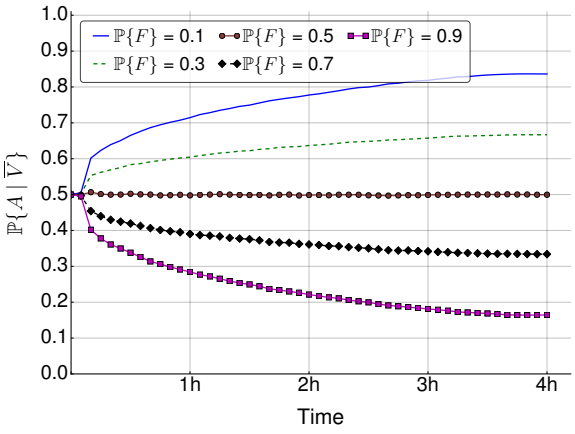
Figure 3.14. Collusion attack: False Positive Rate vs. $\mathbb{P}\{F\}$, MTPs, and attackers.

explained by considering that when the collusion groups are more, less attackers will belong to the same group, and so it is more likely that a scheme based on aggregation may perform better.

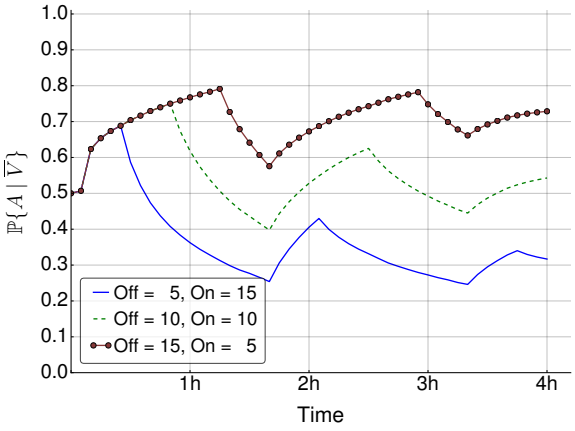
Finally, in Figure 3.15 we report the probability $\mathbb{P}\{A|\overline{V}\}$ of FIRST (i.e., the probability that a report will be accepted when not validated) as a function of time, in all the considered attacks. In the Corruption attack, as expected $\mathbb{P}\{A|\overline{V}\}$ converges to the $\mathbb{P}\{\overline{F}\}$ probability of the attackers. In the On-off attack, FIRST reacts by decreasing the $\mathbb{P}\{A|\overline{V}\}$ probability and increasing it in the OFF phases. Same behavior is also experimented in the Collusion attack, but in this case, the performance is not affected by the number of attackers as explained above. As described in Section 3.4.3, $\mathbb{P}\{A|V\}$ is equal to $P\{\overline{F}\}$, because when reports are verified the probability of misclassification is zero; on the other hand, when the reports are not validated, we would like that $\mathbb{P}\{A|\overline{V}\}$ also tended to $\mathbb{P}\{\overline{F}\}$, and Figure 3.15 shows that FIRST achieves such goal.

3.5.2. Participatory PerCom. In addition to the participatory traffic sensing use-case as described above, the performance of FIRST is also evaluated by implementing a smartphone crowdsensing system designed to monitor the attendance of participants at various events during the IEEE PerCom 2015 conference held in St. Louis, Missouri, USA. In such a system, the voluntary participants were asked to regularly submit **(i)** the conference room they were currently in, and **(ii)** the (approximate) number of participants in that room. The goal of the experiment was to evaluate the accuracy of FIRST in classifying sensing reports sent by participants in a practical scenario.

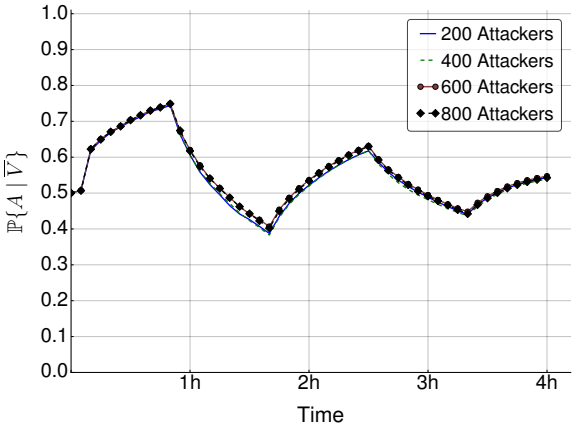
The server-side of the smartphone crowdsensing system handling the storage of sensing reports was implemented by using a dedicated virtual machine on *Amazon Web Services*. Figure 3.16 shows the screenshots of the Android and iOS apps



(a)



(b)



(c)

Figure 3.15. FIRST Acceptance probability $\mathbb{P}\{A|\overline{V}\}$ in corruption, On-off, and Collusion attacks.

distributed to the participants[‖]. The apps provided a simple interface for the participants to report the room they were in (8 choices, from 'A' to 'H'), and the approximate number of people in that room (5 choices, 'Less than 10', 'Between 10 and 20', 'Between 20 and 50', 'Between 50 and 100', and 'More than 100'). In order to recruit participants, the conference and workshop attendees were asked whether they were willing to install our app and participate in the experimental study. This way, 57 participants attending the entire conference and workshops were recruited.

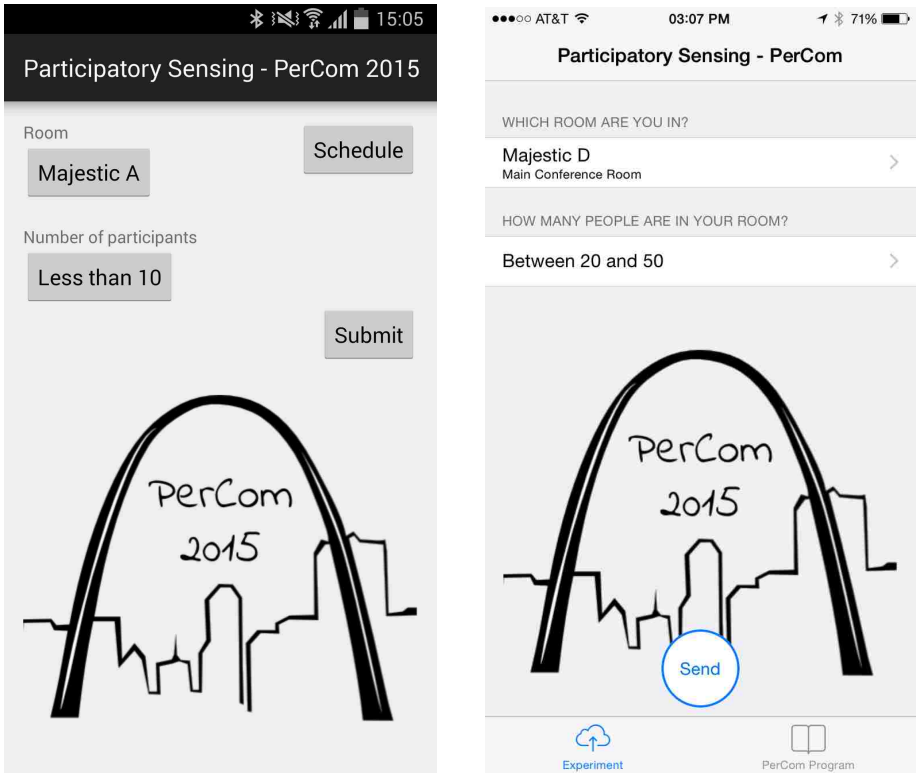


Figure 3.16. Screenshots of the smartphone crowdsensing app, Android and iOS.

In order to acquire ground-truth information about the location of participants, 20 *Gimbal*™ beacon devices[**] were used, which were deployed as in Figure 3.17

[‖]IRB approval of experiments available on file upon request.

[**]Available at `http://www.gimbal.com`

and which emitted periodically *Bluetooth* packets that were received by the smartphone crowdsensing app. Whenever a user sent us a report, the location of the nearest beacon was also automatically included in the report by the SC app. This way, it was possible to acquire ground-truth information on user location.

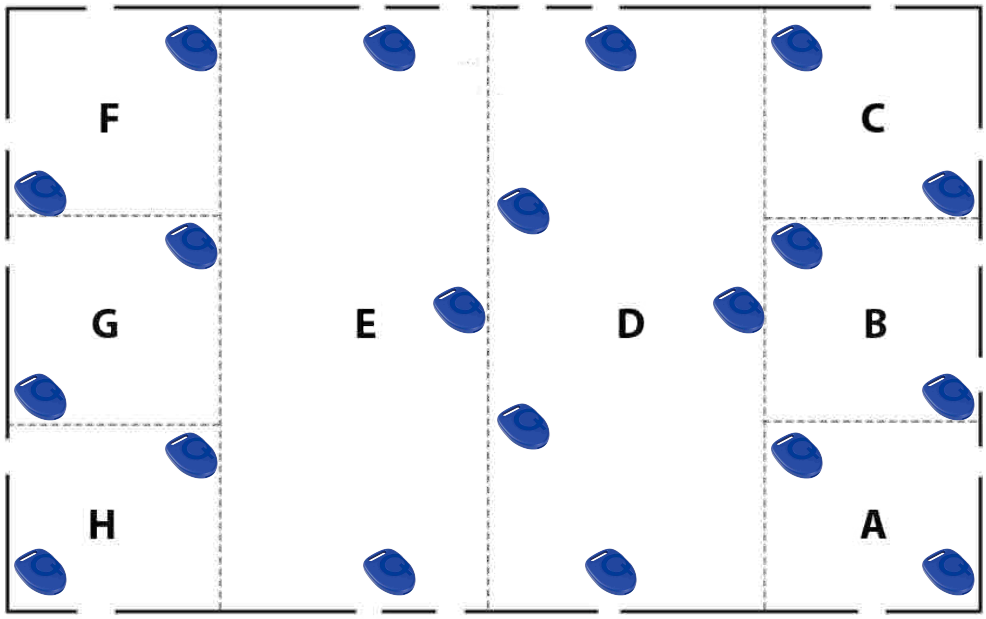


Figure 3.17. Position of Bluetooth beacons.

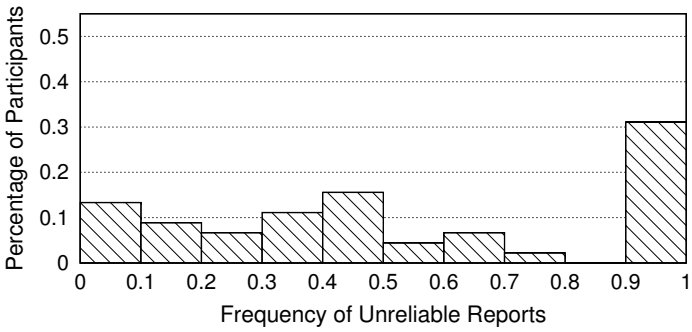
To acquire ground-truth information about the number of people in each room, three people voluntarily acted as MTPs and sent every 5 minutes the actual number of people in each conference room. Prior to the conference, to evaluate the impact of the T parameter (i.e., MTP reporting interval), 4 concurrent, real-time classification processes were implemented, each one taking into account different MTP reporting intervals (10, 15, 20, and 25 minutes), aiming at evaluating the impact of the length of the MTP reporting interval.

During the experiment, it was observed that the number of people attending a particular event was almost constant during a 10-minute time window. Therefore, the MTP reports were used to validate all the reports sent in the following 10-minute time frame. More specifically, a user report was validated as *reliable* if (i) an MTP report r was sent during the 10-minute time frame before the user report was received, *and* (ii) the reported number of people in that room was in the same range as the one sent by the MTP in r. If the number of people in the room reported by the user mismatched the information acquired by the MTP, the report was considered *unreliable*. Otherwise, if no MTP report was available during the previous 10-minute time window, Equation (3.7) was used to decide whether to consider the report as reliable, as explained in Section 3.4.4. After the experiment, the ground-truth information provided by the Bluetooth beacons and the MTPs was used to calculate the classification accuracy.

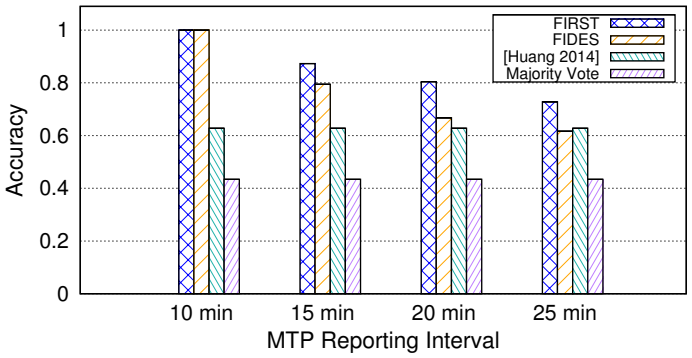
Figure 3.18(a) shows the distribution of the percentage of participants that had submitted unreliable reports with a given frequency. For graphical reasons, frequencies in the x-axis have been grouped into intervals of length 0.1. Figure 3.18(a) points out that about 44% of the participants submitted more than 50% of unreliable reports; moreover, over 30% of participants submitted more than 90% of unreliable reports when participating. These results make this experiment ideal to study the performances of FIRST given the number of unreliable reports is significant.

Figure 3.18(b) illustrates the accuracy of the considered approaches as a function of the MTP reporting intervals implemented in the experiments. These results conclude that FIRST outperforms existing approaches as far as classification accuracy is concerned. In particular, FIRST achieves on the average an accuracy of 80.16%, as compared to FIDES, [Huang 2014] and majority vote which achieve 69.20%, 62.76%, and 43.38%, respectively. The results can be explained as follows. When the MTP reporting interval is 10 minutes, both FIDES and FIRST achieve accuracy of 100%,

because each report is validated by MTPs. As the reporting interval increases, FIDES performs worse than FIRST due to the challenge in finding a parameter setting which achieves good performance in all scenarios. In contrast, FIRST does *not* require any parameter setting to be implemented, and it is able to achieve high accuracy in all the considered scenarios.



(a)



(b)

Figure 3.18. (a) Frequency of unreliable reports vs. percentage of participants. (b) Comparison of FIRST vs. FIDES, Majority Vote and [Huang 2014].

Note that the majority vote and [Huang 2014] schemes do not rely on MTPs, and instead leverage only sensing reports from users to infer their reliability. As a result, the accuracy achieved by majority vote and [Huang 2014] in Figure 3.18(b) does not depend on the MTP reporting interval. As far as performance is concerned, Figure 3.18 concludes that such approaches do not obtain accuracy values close to FIRST

and FIDES. This is due to the fact that approaches based on majority vote are not resilient to large number of unreliable reports, which is the case of the Participatory PerCom experiments, as shown in Figure 3.18(a).

3.6. RELATED WORK

Related work can be divided into two main approaches: *trusted platform modules* (TPMs) and *reputation-based systems*. TPMs are hardware chips that reside on the participants' devices, and ensure that the sensed data is captured by authentic and authorized sensor devices within the system [22, 33, 76]. The main drawback of this approach is that TPMs require additional hardware not currently available on off-the-shelf devices, implying such solutions are not readily deployable. Moreover, TPM chips are tailored to verify data coming from physical sensors (e.g., accelerometer, camera). Thus, they are not applicable to smartphone crowdsensing systems in which the information is directly supplied by the participants.

Most of related work has focused on developing reputation-based systems to increase information reliability. Specifically, they associate each user with a *reputation level*, which is estimated and updated over time. Among related work, [37, 53, 74, 85] are the most relevant. In [85], the authors proposed *ARTsense*, a reputation-based framework that includes a privacy-preserving provenance model, a data trust assessment scheme, and an anonymous reputation management protocol. The main issue of [85] is that user reputation is updated by considering contextual factors, such as location and time constraints. Given user location and timestamp of reports are easily forgeable quantities, the solution proposed in [85] may not perform well in practical smartphone crowdsensing systems, where malicious users may voluntarily tamper with their GPS location and timestamp of reports.

Recently, in [37] the authors proposed a reputation framework which implements an improved version of majority vote. The main limitations of this framework

are (i) the assumption of constant sampling rate, which is not realistic in asynchronous smartphone crowdsensing systems, and (ii) the poor resilience to a large number of malicious users, as the framework uses a modified version of majority vote to update user reputation levels. To overcome such limitation, in [74] the authors proposed *FIDES*, a reputation-based framework that used mobile security agents to classify sensing reports. Similarly to us, FIDES is also resilient to a large number of malicious users. However, as in [37], the necessity to set a significant number of parameters makes the actual performance of the framework hardly predictable in reality. On the other hand, FIRST does *not* depend on specific parameters. Furthermore, in this study it is considered the problem of modeling and optimizing the information reliability [64].

3.7. CONCLUSIONS

In this section it was presented FIRST, a novel framework that models and optimizes the information reliability in smartphone crowdsensing systems. First, the system model, the concept of mobile trusted participants (MTPs), and the MTP optimization problem (MOP) were introduced. Then, the main components of the FIRST framework were discussed in detail, which include a novel likelihood estimation algorithm (LEA) and the MTP optimization algorithm (MOA) that provides optimum solution to the MOP. Furthermore, the framework was extensively evaluated through real mobility traces in the context of participatory traffic sensing, and by a practical implementation of a system that monitored participants' attendance at the IEEE PerCom 2015 conference. Finally, FIRST was compared with state-of-the-art literature. Results have shown that FIRST outperforms existing work in increasing information reliability and is able to capture the performance of the system with significant accuracy.

4. INCENTIVE MECHANISMS FOR CROWDSENSING

The collaborative nature of smartphone crowdsensing implies that its success is strictly dependent on the active participation of users. However, smartphone users invest their personal resources (e.g, time, battery, and bandwidth) while executing sensing tasks. Thus, a user would not be interested in participating in the crowdsensing process unless she receives a satisfying reward. For this reason, a significant number of mechanisms to incentive the users' participation have been proposed, as recently surveyed in [31, 75].

Despite sound mathematical analysis, the main limitation of existing mechanisms is that they implicitly assume that participants will always be able to perform the sensing tasks assigned to them. However, in most smartphone crowdsensing systems, the contributors are pedestrians, drivers, or people commuting to their workplace through public transportation [75]. For example, in the well-known traffic monitoring application *Waze* [25, 87], participants are drivers or commuters traveling from one place to another; thus, their mobility largely depends on current traffic conditions. For example, it may not be always possible for a Waze participant driving in New York City to execute spatio-temporal constrained tasks such as "*Submit road traffic information at Times Square in 5 minutes and Central Park South in 10 minutes*". Therefore, the mobility of the participants should be assumed as *uncertain*, and specific incentive mechanisms should be designed to maximize the likelihood that the sensing tasks will be actually executed. Furthermore, the incentive mechanism should also assume that the budget for recruitment of participants is often *limited*, and thus not every participant can be hired by the system.

In this section, the problem of maximizing the likelihood of successful sensing task execution with participants having uncertain mobility and strict hiring budget

constraint is solved. The problem is cast in the context of *truthful budget-feasible mechanism design*, a branch of game theory that studies how to influence the outcome of a game towards a certain objective [70]. Specifically, the interaction between the smartphone crowdsensing platform (in short, SCP) and the participants is modeled as a *reverse auction*, where the SCP is the buyer (i.e., the auctioneer), and the participants are the sellers (i.e., the bidders) of sensing tasks.

Participants compete with each other by submitting a bid containing the expected payment for executing the sensing tasks advertised by the SCP. The SCP, in turn, will use the auction mechanism to select the winning bidders and compute their payment. The goal is to select participants so as to maximize the probability that the sensing tasks will be executed, and guarantee the total payment will be contained in the budget. In order to avoid the situation in which participants overbid to obtain a greater payment, which leads to inefficiencies in the participants' selection process [70], the auction must also be *truthful*, in sense that participants are not incentivized to overbid by the mechanism.

To effectively solve this problem, the theory of budgeted maximization of submodular functions [44] is leveraged. This theory is used to mathematically model the *Budgeted Value Maximization Problem* (BVM) studied in this section, and show that it is NP-Hard. Two polynomial-time mechanisms are derived, called *Truthful Value Maximization* (TVM) and *Heuristic Value Maximization* (HVM). It is also mathematically proven that the mechanisms guarantee truthfulness, individual-rationality and are budget-feasible. Finally, to provide scalability, it is also presented an implementation of the mechanisms on the well-known *MapReduce* [18] framework for parallel computation.

The mechanisms are extensively evaluated by considering a traffic monitoring application where taxi drivers submit information about traffic events (e.g., accidents,

traffic lines, etc.) during their trips. In order to realistically implement the application, and experiment with different mobility patterns, real-world mobility traces of taxi cabs in San Francisco [72], Rome [4], and Beijing [91] were used. Experimental results demonstrate that our mechanisms outperform the state of the art [16, 77, 89] by improving on the average of 30% with respect to the existing approaches the likelihood of successful task execution, and by achieving a speed-up factor of 12x on the considered experimental setup with MapReduce.

To summarize, the main contributions of this section are as follows:

- Two incentive mechanisms for smartphone crowdsensing systems that consider uncertain mobility of participants and are parallelizable on MapReduce are designed;
- The proposed incentive mechanisms are validated through experiments, and it is demonstrated that they improve significantly the state of the art and able to deal with large number of participants.

The section is organized as follows. Section 4.1 introduces the system model, which Section 4.2 the problem definition. Section 4.3 presents the budget-feasible mechanisms developed in this thesis. Section 4.4 presents the experimental results. Related work is summarized in Section 4.5, while Section 4.6 draws conclusions.

4.1. PROBLEM DEFINITION

This study considers a smartphone crowdsensing architecture consisting of a platform (SCP) which can be accessed through 3G/4G or WiFi Internet connection. The data collection process is detailed as follows. Hereafter, the terms “user”, “participant” and “bidder” will be referred to interchangeably, as well as “system” and “SCP”.

- Before the sensing campaign may begin, users interested in participating to the smartphone crowdsensing campaign download through common application

(app) markets such as *Google Play* or *App Store* the smartphone crowdsensing app (step 1), which is responsible for handling data acquisition, transmission, and visualization.

- When the sensing process begins, the SCP generates *sensing tasks* that need to be executed (step 2). The information pertaining to each sensing task depends on the sensing application, and specifies a series of requirements, such as the sampling rate requested, minimum sensing time, maximum distance from specified location, or task expiration time [75]. For example, a sensing task might be "report the traffic status near the Golden Gate bridge in San Francisco by 5:00PM". Depending on the application, tasks can be retrieved by the users asynchronously, e.g., each day, or whenever requested.
- After the advertisement of the sensing tasks, the SCP collects information that is used to make a choice regarding the scheduling of sensing tasks. Among other information, users supply a *bid*, which is the amount of reward the user would like to receive to perform sensing services (step 3). Bids are used by the auction-based incentive mechanisms developed in this thesis to determine (i) the scheduling of sensing tasks and (ii) the reward to assign to each participating user. If available, the SCP may also collect mobility information about the user, which can be used to determine the user's availability to perform the sensing tasks.
- After receiving the bids from the users, the SCP runs the auction mechanism, and according to the auction's result, selects a subset of users, assigns them the sensing tasks, and communicates the reward assigned to each user (step 4). After being selected and instructed on the sensing task to execute, a user is allowed to begin performing the sensing service using the sensing application (step 5). The data collection may be manual or automatic, depending on the application context [42]). The sensing application handles the transfer of the

sensed data from the smartphone to the PSP, making use of communication infrastructure available to the mobile phone, such as WiFi or 3G/4G connectivity [75]. After the sensing tasks are executed and the data has been transmitted, the SCP assigns the reward communicated in step 4 to each participating user (step 6).

4.2. SYSTEM FORMALIZATION

In order to model users' mobility, the sensing area is divided up into *sectors*, which may have different size and represent the sensing granularity of the application. For example, in a traffic monitoring application, the sectors can be as large as a city block, whereas in air monitoring applications the sector can be as large as a neighborhood. We also assume time is discretized, with j being the j-th time slot between t_j and t_{j+T} .

Let $\mathcal{S}$ define the set of $s = |\mathcal{S}|$ sectors forming the sensing area. Let $\mathcal{Q}$ be the set of $m = |\mathcal{Q}|$ participants competing for offering their sensing services. We define as *sensing task* as a sensing activity that the SCP needs to be performed at a particular place and in a particular moment in time. More formally, since space and time are discretized, a sensing task is modeled as a tuple $\tau_{i,j} = (i, j)$ where $i \in \mathcal{S}$ indicates the sector and $j \in \mathbb{R}^+$ indicates the timestep (e.g., $\tau_{3,4}$ indicates the sensing task involving sector 3 at timestep 4). We will indicate as $\mathcal{Z}$ the set of sensing tasks.

Is is assumed that participants can leave and enter the system at their will. For simplicity, it is defined as $t = 0$ the moment in which the auction is executed. To model the participants' mobility, $p_k^{i,j}$ will indicate the probability that the k-th participant will be in sector i at time j. For the sake of generality, the mechanisms are derived with general $p_k^{i,j}$. In this way, the mechanisms and proofs derived will be general and applicable with any state-of-the-art mobility model.

Participants spend resources when performing sensing services, for example, their personal time, battery, and bandwidth. It is denoted by γ_k the cost of the k-th participant for executing a sensing task, which also includes the minimum profit that the participant desires to earn by participating. This information is considered as personal, which means that γ_k is not revealed to the SCP. In general, the participant may bid a different quantity than γ_k; it is thus defined ν_k as such quantity. The bid is of each participant is therefore the quantity $\{\nu_k\}$. It is also defined as $\mathcal{B}$ the set of bids submitted by the participants for the current auction.

Some terms are now defined, that will be frequently utilized in this section.

Definition 4: (Mechanism). Let $\mathcal{T}$ be set of winning bidders, and $\mathcal{R}$ be the vector of rewards given to the auction participants. A mechanism $\mathcal{M}$ defines a tuple $\{\alpha, \pi\}$, where $\alpha : \mathcal{B} \rightarrow \mathcal{T}$ is defined as the allocation function of the auction, and $\pi : \mathcal{T} \rightarrow \mathcal{R}$ is the payment function of the auction.

Definition 5: (Utility). The utility obtained by the k-th auction participant by bidding ν_k and receiving as reward $\mathcal{R}(k)$ is the quantity $u_k(\nu_k) = \mathcal{R}(k) - \gamma_k$.

It is reasonable to assume that the participants are selfish and are only interested in maximizing their own utility as much as possible. To this purpose, they may *overbid* (i.e., submitting ν_k much higher than γ_k), trying to achieve a higher reward. This may ultimately compromise the auction's efficiency and lead to low performance. To solve this problem, mechanisms that align the users' interests with the system goals are needed. This is why in this thesis *truthful* mechanisms were designed.

A mechanism is called truthful if any user maximizes her utility by bidding her real cost γ_k, no matter how other users may act. It is also required to the mechanisms to satisfy *individual-rationality*, which means that any user always gets a non-negative utility, and to be computational efficient. The formal definitions of these properties is given below.

Definition 6: (Truthfulness) [70]. The mechanism $\mathcal{M} = \{\alpha, \pi\}$ is truthful iff $u_k(\gamma_k) \geq u_k(x)$, for any $x \neq \gamma_k$.

Definition 7: (Individual Rationality) [70]. A mechanism is individual-rational iff $u_k(x) \geq 0$, for each bidder k.

Definition 8: (Computational Efficiency) [70]. A mechanism is computationally efficient iff α and π have at most polynomial complexity in the number of bids.

The user incentivization problem studied in this section is now formalized. Let us define V_{ij} as a quantity indicating the value that sensing task τ_{ij} has for the SCP. Intuitively, V_{ij} models the preference that the system has for some sensing tasks instead of others (e.g., covering some neighborhoods of a city may be more important than covering others). For notation simplicity, $V_{ij} = 0$ if $\tau_{ij} \notin \mathcal{Z}$. It is also defined as $W(i, j, \mathcal{T})$ the following quantity, $W(i, j, \mathcal{T}) = 1 - \prod_{k \in \mathcal{T}}(1 - p_k^{i,j})$, which expresses the probability that at least one participant is in sector i at time j (i.e., the probability that at least one participant will be able to execute the sensing task). Finally, let us define B as the budget available to the SCP for running the auction, and as z the maximum timestamp of the available sensing tasks, hereafter referred to as auction *duration*.

Problem 1: Budgeted Value Maximization (BVM). Given values V_{ij} and function W, let $\mathcal{V}(\mathcal{T})$ be defined as

$$\mathcal{V}(\mathcal{T}) \triangleq \sum_{i=1}^{s} \sum_{j=1}^{z} V_{ij} \cdot W(i, j, \mathcal{T}) \tag{4.1}$$

Find set of bidders $\mathcal{T}^*$ and payments vector $\mathcal{R}^*$ such as

$$\mathcal{T}^* = \underset{\mathcal{T} \subseteq \mathcal{B}}{\operatorname{argmax}} \sum_{i=1}^{s} \sum_{j=1}^{z} V_{ij} \cdot W(i, j, \mathcal{T}), \sum_{\rho_k \in \mathcal{R}^*} \rho_k \leq B \tag{4.2}$$

The optimization function defined in BVM can be seen as a function that expresses the probability that the sensing tasks will be executed (each one weighted with their relative value). It is now demonstrated that BVM is NP-Hard.

Lemma 1: BVM is NP-Hard.

Proof. In order to prove the NP-Hardness of BVM, a reduction from the well-known Bounded Knapsack problem (BKP) is provided. A general BKP instance has a capacity B and a set of items $\Omega = \{s_1 \cdots s_n\}$, where each item $s_i \in \Omega$ has a value v_i and a weight w_i. The goal of BKP is to find a set $S^* \subseteq \Omega$ whose items provide maximum value and do not exceed the capacity B of the knapsack. The general BKP instance is translated into a simpler instance of BVM, where a single time step (i.e., $z = 1$) is considered. It is also assumed each bidder is in a different sector of the sensing area, the probability of being in that sector is equal to one, and that the payment rule is such as $\rho_k = \nu_k$. This way, the optimization function becomes $\sum_{k=1}^{m} V(k) \cdot X_k$, where X_k is an indicator function that is equal to one if a bidder is in sector k, and zero otherwise. Therefore, a solution to BVM can be translated into a solution of the BKP instance by setting the value v_k of each knapsack item to $V(k) \cdot X_k$, its weight w_k to ν_k, and B as the size of the knapsack. As a result, solving BVM is at least as hard as solving BKP, therefore BVM is NP-Hard.

From Lemma 1, it is concluded that it is unlikely to design a mechanism with α^* and π^* in the polynomial computational class, unless $P = NP$. Therefore, the rest of the section will focus on designing mechanisms that provide bounded *approximate* solution to the BVM.

Although proven to be NP-Hard, the BVM problem has a special property that allows the design of approximate solutions. Specifically, it is now proven that its objective function (defined in Equation 4.1) belongs to the family of *submodular*

functions. Submodular function optimization theory provides solutions to many NP-Hard problems [44]. Submodularity is now defined, and some properties necessary to design a greedy algorithm are proven.

Definition 9: (Submodularity). Given ground set S and a function $\mathcal{F} : 2^S \rightarrow \mathbb{R}^+$, then $\forall A \subseteq B \subseteq S$, $\mathcal{F}$ is submodular iff, for any $i \in S$, $\mathcal{F}(A \cup \{i\}) - \mathcal{F}(A) \geq \mathcal{F}(B \cup \{i\}) - \mathcal{F}(B)$.

Lemma 2: The function $\mathcal{V}(\mathcal{T})$, defined in Equation (4.1) is (i) submodular, (ii) non-decreasing, and (iii) $\mathcal{V}(\emptyset) = 0$.

Proof. In order to prove (i), it is shown that, for sets $A \subseteq B \subseteq S$, it is true that $\Delta_{A,k} \geq \Delta_{B,k}$, where $\Delta_{X,k} \triangleq \mathcal{V}(X \cup \{k\}) - \mathcal{V}(X)$ is the increment in value to $\mathcal{V}$ given by the addition to X of a generic element k, hereafter referred to as *marginal value* of k given X.

$$
\begin{aligned}
\Delta_{X,k} = & \sum_{i=1}^{s}\sum_{j=1}^{z} V_{ij} \cdot \left[1 - \prod_{y \in X \cup \{k\}} (1 - p_y^{i,j})\right] - \\
& \sum_{i=1}^{s}\sum_{j=1}^{z} V_{ij} \cdot \left[1 - \prod_{y \in X} (1 - p_y^{i,j})\right] \\
= & \sum_{i=1}^{s}\sum_{j=1}^{z} V_{ij} \cdot \left[1 - (1 - p_k^{i,j}) \prod_{y \in X} (1 - p_y^{i,j})\right] - \\
& \sum_{i=1}^{s}\sum_{j=1}^{z} V_{ij} \cdot \left[1 - \prod_{y \in X} (1 - p_y^{i,j})\right] \\
= & \underbrace{\sum_{i=1}^{s}\sum_{j=1}^{z} V_{ij} \cdot p_k^{i,j}} \cdot \prod_{y \in X} (1 - p_y^{i,j})
\end{aligned}
\tag{4.3}
$$

The braced section in $\Delta_{X,k}$ does not depend on X. Thus, it is only needed to prove the following claim, which is that $\prod_{y \in A} (1 - p_y^{i,j}) \geq \prod_{y \in B} (1 - p_y^{i,j})$ holds for any $A \subseteq B$. It is now derived an equation that proves this claim.

$$\underbrace{\prod_{y \in A} (1 - p_y^{i,j})}_{\triangleq Z} \geq \prod_{y \in B} (1 - p_y^{i,j}) \triangleq \tag{4.4}$$

$$Z \geq Z \cdot \prod_{y \in \{B-A\}} (1 - p_y^{i,j}) \triangleq \tag{4.5}$$

$$1 \geq \prod_{y \in \{B-A\}} (1 - p_y^{i,j}) \tag{4.6}$$

As the quantity $p_y^{i,j}$ is a probability and therefore by definition less or equal to one, the disequality above holds. This proves property (i). Furthermore, property (ii) derives straightforwardly from the fact that $W(i, j, \mathcal{T} \cup \{i\}) \geq W(i, j, \mathcal{T})\ \forall \mathcal{T}$, and (iii) follows from $W(i, j, \emptyset) = 0$, by definition of empty product.

4.3. MECHANISM DESIGN

In this section, two mechanisms are described, namely Truthful Value Maximization (TVM) and Heuristic Value Maximization (HVM) to solve the BVM problem defined in the previous section.

In order to solve the problem of truthful bidding under budget constraints, Truthful Value Maximization (TVM) is designed, which is a mechanism that adopts recent advances in the field of submodular function maximization [44] to provide a solution to the BVM with proven approximation ratio through a greedy strategy.

Algorithm 3 presents the allocation function of TVM. The algorithm incrementally constructs a set of winners $\mathcal{T}$, initially empty (line 1). At each iteration, the algorithm picks an unconsidered bidder k^* having maximum weight, where the weight is defined as the increase in the function $\mathcal{V}$ that k^* provides, divided by its bid (line 5). The bidder k^* is included in $\mathcal{T}$ only if the current sum of bidding values is not exceeded (line 6). and if a condition regarding the new bid ν_k is satisfied (line 7). The algorithm returns the set of winning bidders $\mathcal{T}$.

The payment scheme of TVM, which is reported in Algorithm 4, assigns to each winning bidder a payment corresponding to the *critical value*, which provably ensures truthfulness of the mechanism.

Algorithm 3 Truthful Value Maximization (TVM) allocation function

Input: B, $\mathcal{B}$, $\mathcal{Q}$, $\mathcal{V}$
Output: $\mathcal{T}$, $\mathcal{T}_v$

$\mathcal{T} = \emptyset$
$\mathcal{T}_v = \emptyset$
$\mathcal{T}_c = \mathcal{Q}$
while $\mathcal{T}_c \neq \emptyset$ **do**

$$k^* = \underset{k \in \mathcal{T}_c}{\operatorname{argmax}} \, {}^{\Delta_{\mathcal{T},k}}/_{\nu_k}$$

  if $\nu_{k^*} + \sum_{k \in \mathcal{T}} \nu_k \leq B$ **then**
    if $\nu_{k^*} \leq \frac{B}{2} \cdot \frac{\Delta_{\mathcal{T},k^*}}{\Delta_{\mathcal{T},k^*} + \sum_{v \in \mathcal{T}_v} \Delta_{\mathcal{T},v}}$ **then**
      Append k^* to $\mathcal{T}$
      Append $\Delta_{\mathcal{T},k^*}$ to $\mathcal{T}_v$
    else
      $\mathcal{T}_c = \emptyset$
    end if
  end if
  $\mathcal{T}_c = \mathcal{T}_c - \{k^*\}$
end while
return $\mathcal{T}$,$\mathcal{T}_v$

We point out that the critical value of a bidder is the maximum value that bidder could have bid and still win the auction. Unfortunately, the critical value computation is complicated by the submodularity of the marginal contributions $\Delta_{\mathcal{T},k}$, which implies that the value depends on the sorting order of the greedy allocation policy.

In order to compute payments in an efficient way, the following strategy is used. For each bidder i, the maximum of all the possible bids that i could have declared and

still get allocated are considered, as explained next. Consider running the allocation function without i. For the first j participants in the marginal contribution sorting, using the marginal contribution of i at point j it is possible to find the maximal cost that agent i can declare in order to be allocated instead of the agent in the j-th place in the sorting. For reader's convenience, it is now shown a small example of the steps taken by TVM to compute winners and rewards.

Algorithm 4 Truthful Value Maximization (TVM) payment function

Input: $\mathcal{T}$, B, $\mathcal{B}$, $\mathcal{Q}$, $\mathcal{V}$
Output: $\mathcal{R}$

1: **for** every $i \in \mathcal{T}$ **do**
2: $\quad \mathcal{B}^i = \mathcal{B} - \{\nu_i\}, \mathcal{Q}^i = \mathcal{Q} - \{i\}$
3: $\quad \mathcal{T}^i$,$\mathcal{T}^i_v$ = Algorithm-3$(B, \mathcal{B}_i, \mathcal{Q}_i, \mathcal{V})$
4: $\quad \mathcal{X} = \emptyset$
5: $\quad$ **for** every $j = 1 \dots |\mathcal{T}^i|$ **do**
6: $\quad\quad \nu_{ij} = \frac{\Delta_{\mathcal{X},i} \times \nu_j}{\Delta_{\mathcal{X},j}}$
7: $\quad\quad \rho_{ij} = \frac{B}{2} \cdot \frac{\Delta_{\mathcal{X},i}}{\sum_{j' \leq j} \mathcal{T}^i_v(j) + \Delta_{\mathcal{X},i}}$
8: $\quad\quad$ Append $\mathcal{T}^i(j)$ to $\mathcal{X}$
9: $\quad$ **end for**
10: **end for**
11: **return**

$$\mathcal{R}(i) = \begin{cases} \max\limits_{j \in \mathcal{T}^i} \{\min\{\rho_{ij}, \nu_{ij}\}\} & i \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases}$$

Example. Let us consider for simplicity a sensing area composed of 4 sectors and a time range of only 1 timestep, $t = 1$. Let us consider the case in which 3 bidders are competing to offer their sensing services. The value of each sector is $V = \{.3, .2, .1, .4\}$, while the mobility distributions of the bidders at time $t = 1$ are as follows: $p^1_1 = \{.2, .1, .3, .4\}$, $p^1_2 = \{0, .8, .05, .15\}$, $p^1_3 = \{.4, .2, 0, .4\}$. The bids submitted are $\mathcal{B} = \{10, 8, 12\}$.

Allocation function Input: $B = 20$, $\mathcal{B} = \{10, 8, 12\}$

- *Step 1:* $\mathcal{T} = \{\}$, $\mathcal{T}_v = \{\}$

$$\frac{\Delta_{\mathcal{T},1}}{\nu_1} = \frac{0.2 \cdot 0.3 + 0.1 \cdot 0.2 + 0.3 \cdot 0.1 + 0.4 \cdot 0.4}{10} = 0.027$$
$$\frac{\Delta_{\mathcal{T},2}}{\nu_2} = \frac{0 \cdot 0.3 + 0.8 \cdot 0.2 + 0.05 \cdot 0.1 + 0.15 \cdot 0.4}{8} = 0.028$$
$$\frac{\Delta_{\mathcal{T},3}}{\nu_3} = \frac{0.4 \cdot 0.3 + 0.2 \cdot 0.2 + 0 \cdot 0.1 + 0.4 \cdot 0.4}{12} = 0.0277$$
$$\text{Is } 8 \leq \frac{20}{2} \cdot \frac{0.28125}{0.28125} \text{ ? YES}$$

Append $k^* = 2$ to $\mathcal{T}$

Append $\Delta_{\mathcal{T},k^*} = 0.225$ to $\mathcal{T}_v$

- *Step 2:* $\mathcal{T} = \{2\}$, $\mathcal{T}_v = \{0.225\}$

$$\frac{\Delta_{\mathcal{T},1}}{\nu_1} = \frac{0.2285}{10} = 0.02285$$
$$\frac{\Delta_{\mathcal{T},3}}{\nu_3} = \frac{0.2640}{12} = 0.022$$
$$\text{Is } 10 \leq \frac{20}{2} \cdot \frac{0.02285}{0.02285 + 0.225} \text{ ? NO}$$

Condition on budget (line 7) not fulfilled. Algorithm terminates, and returns $\mathcal{T} = \{2\}$ and $\mathcal{T}_v = \{0.225\}$

Payment function Input: $\mathcal{T} = \{2\}$

- *Step 1:* $i = 2$

  Run auction without 2 (lines 2-3)

  - *Step 1-A:* $j = 1$

$$\nu_{21} = 0.225 \cdot {}^{10}/_{0.27} = 8.33$$
$$\rho_{21} = 10$$

  To win against 1, 2 has to bid 8.33

$$\text{Return } \mathcal{R}(2) = \max_{j \in \{1\}} \{\min\{\rho_{2j}, \nu_{2j}\}\} = 8.33$$

We now prove that TVM is truthful, individual-rational, and budget-feasible. In order to characterize the truthfulness of the mechanism, it is applied the well-known characterization of Myerson [65] in single-parameter domains, which is reported below.

Lemma 3. A mechanism is truthful iff [65] (i) the allocation function is monotone: bidder k wins the auction by bidding ν_k, it also wins by bidding $\nu'_k < \nu_k$; (ii) Each winner is paid the critical value: bidder k would not win the auction if it bids higher than this value.

Lemma 4. TVM is a truthful mechanism, i.e., no bidder can increase her profit by misreporting her true cost.

Proof. Lemma 4 is proven by using Myerson's characterization of truthfulness (Lemma 3). In particular, it is first proven that TVM has a monotone allocation scheme, then it is proven that each winner is paid the critical value.

(*Monotonicity*). The first property is guaranteed by the greediness of the algorithm. By lowering her bid, any allocated bidder would only increase her marginal gain per unit cost and thus be placed ahead in the sorting order considered by the allocation algorithm.

(*Critical value*). According to Algorithm 4, each winning bidder i is paid the following quantity: $\max_j \{\min\{\rho_{ij}, \nu_{ij}\}\}$. Let us consider r to be the index for which $P_i = \min\{\rho_{ir}, \nu_{ir}\}$. Therefore, bidding P_i implies that i would be allocated at position r in the run of the algorithm without i. Four different cases are thus possible.

1. $\nu_{ir} \leq \rho_{ir}$ and $\nu_{ir} = \max_j \nu_{ij}$. Reporting a bid higher than ν_{ir} places bidder i after the first unallocated user k^* in the alternate run of the mechanism, thus i would not be selected.

2. $\nu_{ir} \leq \rho_{ir}$ and $\nu_{ir} < \max_j \nu_{ij}$. Consider some j for which $\nu_{ir} < \nu_{ij}$. Since r has maximality condition, it must be the case that $\rho_{ij} \leq \nu_{ir} \leq \nu_{ij}$. Therefore, bidding higher than ν_{ir} would violate the selection condition (line 7, Algorithm 3) and hence i would not be allocated. For some other j such as $\nu_{ir} \geq \nu_{ij}$, bidding higher than ν_{ir} would place i after j, so i would not be allocated at position j.
3. $\nu_{ir} \geq \rho_{ir}$ and $\rho_{ir} = \max_j \nu_{ij}$. Reporting a bid higher than ρ_{ir} violates the selection condition at each of the indices in $j \in \mathcal{T}^i$, hence i would not be selected.
4. $\nu_{ir} \geq \rho_{ir}$ and $\rho_{ir} < \max_j \nu_{ij}$. Consider some j for which $\rho_{ir} < \rho_{ij}$. Since r has maximality condition, it must be the case that $\nu_{ij} \leq \rho_{ir} \leq \rho_{ij}$. Therefore, bidding higher than ρ_{ir} would put i after j and hence i would not be allocated. For j such as $\rho_{ir} \geq \rho_{ij}$, bidding higher than ν_{ir} would mean i would not be allocated at considered position j.

In all four cases, bidding higher than P_i would cause bidder i to not be selected, which means that P_i is the critical value. Since bidder i is paid P_i, this proves the Theorem.

Lemma 5. TVM is individual-rational, i.e., payments for winning bidders are always greater or equal to their bid.

Proof. Consider the bid that i can declare to be allocated at position $j = i$ (i.e. back at its original position) in the alternate run of the mechanism. Therefore, the payment that i will receive will be $P_i = \min\{\rho_{ii}, \nu_{ii}\}$, We prove that $\nu_i \leq P_i$. First, it is shown that $\nu_{ii} \geq \nu_i$:

$$\nu_{ii} = \frac{\Delta_{\mathcal{X},i} \cdot \nu_i}{\Delta_{\mathcal{X},j}} \geq \frac{\Delta_{\mathcal{X},i} \cdot \nu_i}{\Delta_{\mathcal{X},i}} = \nu_i \tag{4.7}$$

The equality holds because $\nu_j/\Delta_{\mathcal{T},j} \geq \nu_i/\Delta_{\mathcal{T},i}$ since i was selected after $i-1$ in the selection algorithm instead of j. Next, it is shown that $\rho_{ii} \geq \nu_i$:

$$\begin{aligned} \rho_{ij} &= \frac{B}{2} \cdot \frac{\Delta_{\mathcal{X},i}}{\sum_{j' \leq i-1} \mathcal{T}_v^i(j') + \Delta_{\mathcal{X},i}} \\ &= \frac{B}{2} \cdot \frac{\Delta_{\mathcal{X},i}}{\sum_{j \leq i-1} \Delta_{\mathcal{X},j} + \Delta_{\mathcal{X},i}} \geq \nu_i \end{aligned} \tag{4.8}$$

The second equality holds from the fact that the first $i-1$ allocated elements in both the runs of the policies are the same. The third inequality follows from the proportional share criteria used to decide the allocation of i after $i-1$ users were selected already. This proves the Theorem.

Lemma 6. TVM is budget-feasible.

Proof. The maximum payment for a user p is $2 \cdot \frac{\Delta_p}{\sum_{k \in \mathcal{T}} \Delta_k} \cdot B$, where Δ_p is the marginal contribution given by p computed during the run of the allocation function [44]. Summing over all payments, it is obtained $\sum_{i \in \mathcal{T}} \mathcal{R}(i) \leq \sum_{i \in \mathcal{T}} 2 \cdot \frac{\Delta_p}{\sum_{k \in \mathcal{T} \Delta_k}} \cdot \frac{B}{2} \leq B$

Lemma 7. Algorithm 3 has complexity $\theta(m^2 \cdot s \cdot z)$, where m is the number of bidders, s is the number of sectors, and z is the auction allocation span, while Algorithm 4 has complexity $O(m^3 \cdot s \cdot z)$.

Proof. The complexity of Algorithm 3 is dominated by the complexity of the while loop (line 4 through 15). At each iteration, the loop computes the quantities $\Delta_{\mathcal{T},k}$ for each $k \in \mathcal{T}_c$. As every iteration of loop removes one element from $\mathcal{T}_c$, this implies that this computation is performed $m+m-1+m-2+\cdots+1 = \theta(m^2)$ times. The complexity of computing $\Delta_{\mathcal{T},k}$ for a generic k is dominated by the computation of

$$\mathcal{V}(\mathcal{T} \cup \{k\}) = \sum_{i=1}^{s} \sum_{j=1}^{z} V_{ij} \cdot W(i, j, \mathcal{T} \cup \{k\}) \tag{4.9}$$

We now provide a way to recursively derive in constant time $W(i, j, \mathcal{T} \cup \{k\})$ as a function of $W(i, j, \mathcal{T})$. The overall complexity of computing $\mathcal{V}(\mathcal{T} \cup \{k\})$ will be therefore $\theta(s \cdot z)$, which yields a total algorithm complexity of $\theta(m^2 \cdot s \cdot z)$.

$$
\begin{aligned}
W(i,j,\mathcal{T}\cup\{k\}) &= 1-\prod_{q\in\mathcal{T}\cup\{k\}}(1-p_q^{i,j}) \\
&= 1-(1-p_k^{i,j})\cdot\overbrace{\prod_{q\in\mathcal{T}}(1-p_q^{i,j})}^{1-W(i,j,\mathcal{T})} \\
&= 1-(1-p_k^{i,j})\cdot(1-W(i,j,\mathcal{T}))
\end{aligned}
\tag{4.10}
$$

where $W(i,j,\emptyset)=0$ by definition. The complexity is therefore $\theta(m^2\cdot s\cdot z)$. The complexity of Algorithm 4 can be calculated by observing that there are at most m iterations of the main loop, and in each loop the complexity is dominated by the execution of the allocation function. This yields a total complexity of $O(m^3\cdot s\cdot z)$.

4.3.1. Heuristic Value Maximization. As discussed earlier, the stopping criterium of TVM allocation algorithm (line 7, Algorithm 3) limits the efficiency of TVM. This point is demonstrated through an example.

In the following, it will be referred to as *actual budget* (B) the budget available for the current auction provided by the administration, and by *input budget* ($\hat{B}$) the budget that is given as input to the TVM selection algorithm. Figure 4.1 depicts the sum of payments assigned to winning bidders by TVM as a function of the input budget value $\hat{B}$, where 1000 bidders and truncated normal bid distribution (with mean=0.5 and support=1) are considered. As it can be seen from this plot, the sum of payments allocated to the winning bidders is non-linear with $\hat{B}$ and remains significantly below the actual budget.

From this example, it emerges that the performance of TVM can be further optimized by exploring the input budget space and finding the $\hat{B}$ value yielding the highest sum of payments that remains below the actual budget. Although this problem might seem straightforward, it turns out there are several issues to be solved.

First, performing an exhaustive (i.e., linear) search on the input budget space is not practical. This is because, in the worst case, the TVM auction must be run n

times, where n is the size of the input budget search interval. Although the computation complexity of TVM does not directly depend on the input budget, in practice the computational cost of executing the TVM mechanism increases significantly as the input budget increases. Intuitively, this is because the selection algorithm will select more winning bidders, which implies that the payment algorithm must calculate the payment for more winners. Thus, performing the search by running the least amount of auctions is paramount.

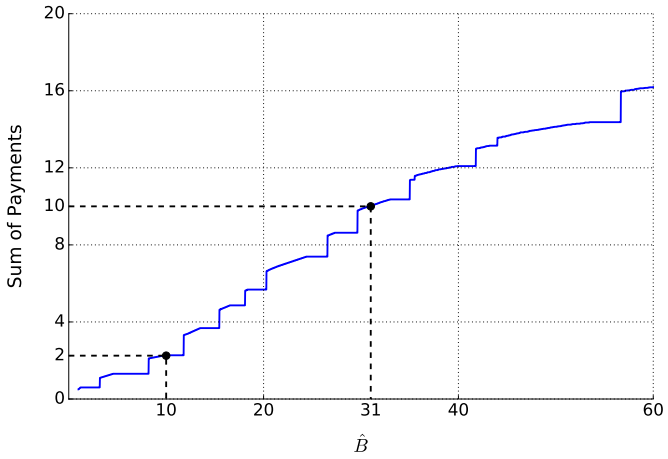


Figure 4.1. TVM: payments vs. input budget.

Second, while the minimum input budget (i.e., B) is known, the maximum input budget must be discovered by some means. Therefore, an efficient algorithm must be designed to define the search interval itself. In order to optimize the computation time of the optimum input budget without renouncing to efficiency, it becomes a necessity to leverage the monotonicity of the sum of payments. To explain the approach, let us assume to have an array A of n elements in which the sum of payments corresponding to all the input budgets from B to $B + n - 1$ are stored. Since payments are monotone with the budget, the array will be sorted. Thus, finding the

optimum input budget corresponds to finding the rightmost place where the given sum of payment can be correctly inserted in array A without compromising the sorted order. Although ordinary binary search may be applied to find such rightmost insertion point, it does not consider that the sum of payments are uniformly distributed (see Figure 4.1).

Algorithm 5 Heuristic Value Maximization

Input: B, $\mathcal{B}$, $\mathcal{Q}$, $\mathcal{V}$
Output: B^*

1: $B_{min} = B$
2: $B_{max} = B + 1$
3: $B_{min}, B_{max}, \mathcal{R}_{min}, \mathcal{R}_{max} = \text{FMMB}(B, \mathcal{B}, \mathcal{Q}, \mathcal{V})$
4: **while** $B_{min} \leq B_{max}$ **do**
5: $\quad B_{cur} = \text{INT}(B_{min}, B_{max}, \mathcal{R}_{min}, \mathcal{R}_{max})$
6: $\quad \mathcal{T}_{cur} = \text{Algorithm-3}(B_{cur}, \mathcal{B}, \mathcal{Q}, \mathcal{V})$
7: $\quad \mathcal{R}_{cur} = \text{Algorithm-4}(\mathcal{T}, B_{cur}, \mathcal{B}, \mathcal{Q}, \mathcal{V})$
8: $\quad$ **if** $\sum_{k \in \mathcal{T}_{cur}} \mathcal{R}_{cur}(k) > B$ **then**
9: $\quad\quad B_{max} = B_{cur} - 1$
10: $\quad\quad \mathcal{R}_{max} = \mathcal{R}_{cur}$
11: $\quad$ **else**
12: $\quad\quad B_{min} = B_{cur} + 1$
13: $\quad\quad \mathcal{R}_{min} = \mathcal{R}_{cur}$
14: $\quad$ **end if**
15: **end while**
16: Return B_{min}

Algorithm 5 introduces *Heuristic Value Maximization* (HVM), which uses interpolation [5] to improve over binary search and find the optimum input budget by computing no more than $O(\log \log n)$ times the TVM auction. First, it finds the upper bound of the search interval B_{max}, by doubling the search interval until the sum of payments is below the actual budget (line 2).

This guarantees that the interval will be found in $O(\log B^*)$ steps. Then, the algorithm computes each step of the search, called B_{cur}, by calculating the line

passing between the two points (B_{min}, R_{min}) and (B_{max}, R_{max}) and computing the next step B_{cur} as the x component of the point passing by the actual budget B. If the payment $\mathcal{R}_{cur}$ yielded by the new point B_{cur} is greater than (less or equal than) the budget, the algorithm explores the left (right) part of the search interval, until the exit condition is not met. In order to further speed up the execution time, the algorithm approximates the $\mathcal{R}_{max}$ and $\mathcal{R}_{min}$ values with the $\mathcal{R}_{cur}$ value.

4.3.2. MapReduce Implementation. Smartphone crowdsensing applications have a large amount of participants. Therefore, it is imperative to take advantage of cloud computing techniques to speed up the execution of large-scale auctions. Among parallel programming models, MapReduce [18] is widely adopted for many data mining tasks on large-scale data. To the best of the author's knowledge, this study is the first to introduce a MapReduce model for incentive mechanisms in smartphone crowdsensing. In this section, it is now discussed the design of a MapReduce model.

In a nutshell, a typical MapReduce model contains two phases: 1) the map phase reads the input data, and converts it into key-value pairs; 2) the reduce phase takes the key-value pairs generated from the map phase as input, and applies an operation to the values belonging to the same key to obtain the desired output.

For TVM, the objective is to adapt Algorithm 3 and 4 (i.e., selection and payment functions) to a parallel version. To this end, it is noticed that the main bottleneck of Algorithm 3 is line 5, which is the computation of the maximum value of $\Delta_{\mathcal{T},k}$ values, for every participant k. However, each of these quantities can be computed separately by different mappers, and the result reduced to obtain the maximum. Regarding the payment function (Algorithm 4), it is noticed that the computation of the payment for each winning bidder (line 1) can be assigned to a different mapper. The same can be applied to compute line 5. As far as HVM is concerned, the MapReduce model can be used for TVM to further parallelize the computation of

the optimum input budget by assigning to different mappers a portion of the search interval, and then apply a reducer to obtain the result.

4.4. EXPERIMENTAL ANALYSIS

In this section,the performance of HVM and TVM is experimentally evaluated and compared with the most relevant existing work. The scalability of the mechanisms has also been evaluated by computing the speed-up factor obtained by running the MapReduce implementation of HVM on a Hadoop cluster.

4.4.1. Experimental Setup. The evaluation has been conducted by emulating an application where taxi cab drivers report vehicular traffic events (similar to Waze [6]). This type of application was chosen since the results may be valid also for similar smartphone crowdsensing applications (e,g., urban air/noise pollution monitoring). In order to obtain real-world participants' mobility and evaluate the performance with different mobility patterns, mobility traces collected from the following datasets were considered:

- *San-Francisco* [72]: This dataset contains mobility traces of approximately 500 taxis in San Francisco, USA, collected during one month.
- *Rome* [4]: In this dataset, 320 taxi drivers in the center of Rome were monitored during March 2014.
- *Beijing* [91]: This dataset collected by Microsoft Research Asia contains the GPS positions of 10,357 taxis in Beijing during one month.

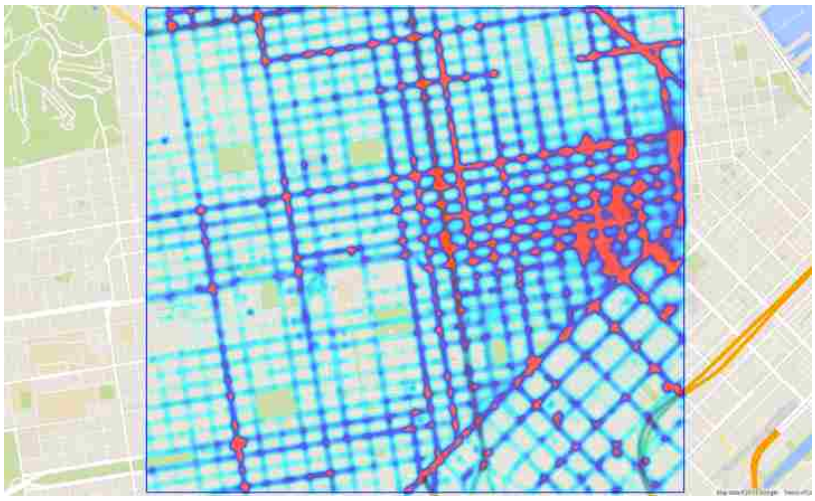
The heatmaps of the datasets are shown in Figure 4.2; the pictures show warmer colors where the mobility is more concentrated. As it can be seen, the mobility is more concentrated in Rome and Beijing, whereas it is more uniform in the San Francisco dataset. Therefore, better performance is expected in the Rome and Beijing experiments (i.e., more users). The sensing area is considered to be approximately

4×4km square areas, which characterize the downtown of cities such as San Francisco, Rome, and Beijing.

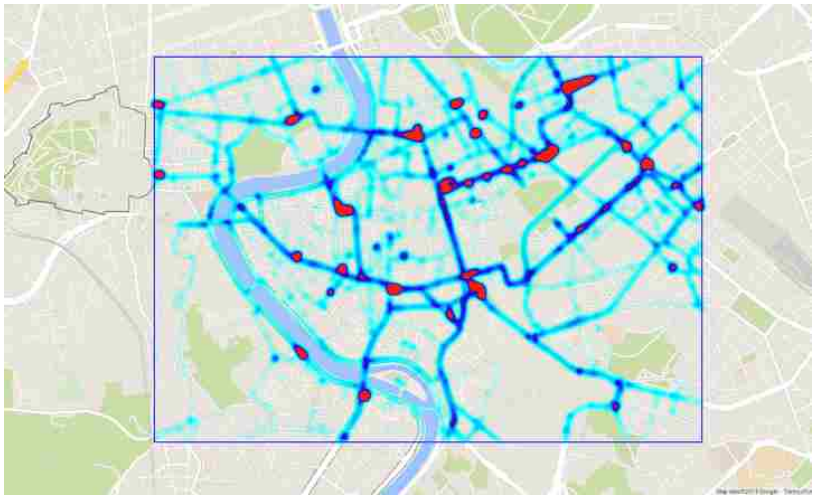
In this application, taxi drivers request to participate to the sensing process by submitting the cost for submitting the sensing report through their smartphone. Whenever the drivers decide to participate, they submit information regarding their current position and the destination of their trip to the SCP by using a smartphone app. Upon reception of such information, the SCP computes the mobility information regarding each participant (i.e., the $p^k_{i,j}$ quantities as described in Section 4.1) by using GraphHopper (`https://www.graphhopper.com`), which provide real-time ETA and routing information. This information, united with the bid information, is used to run the auctions mechanisms as described in this section.

To emulate bidders' behavior, coherently with previous work, the bidding process was emululated by using as a Poisson random variable [28]. In particular, each taxi trip has a duration sampled from a Poisson with $\lambda = 3$ timesteps, while each participant bids happen every $\lambda = 1.5$ timesteps. In all experiments, a timestep of 5 minutes was considered, and the sensing areas were divided into 400 sectors, so as that each sector is approximately as large as a city block. The SCP generates sensing tasks in such a way that each sector must be covered in each timestep. In order to account for the uncertainty in the mobility of the users, a Gaussian-distributed noise with zero mean and std-dev 0.25 to the mobility reported by the participants has been added. For the distribution of the bids' costs, coherently with previous work, it was considered Gaussian-distributed costs with mean 0.5 and std-dev 0.15. In all experiments, 100 bidders were considered, if not stated otherwise. In each experiment, 100 repetitions were performed and 95% confidence intervals were computed, which are not shown when they are below 1% of the average.

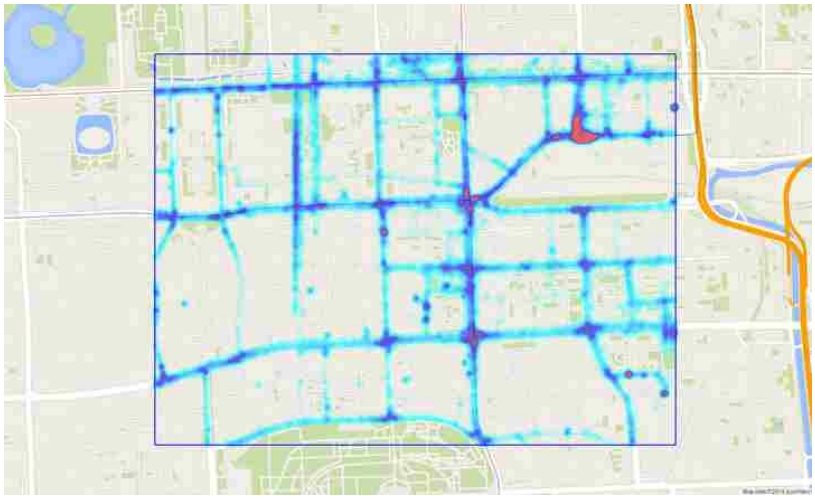
For comparison reasons,the state of the art work due to Singer [77], Chen *et al.* [16], and Yang *et al.* [89] (MSensing auction), recently extended to [90], was



(a) San Francisco



(b) Rome



(c) Beijing

Figure 4.2. Heatmaps of the mobility traces contained in the datasets.

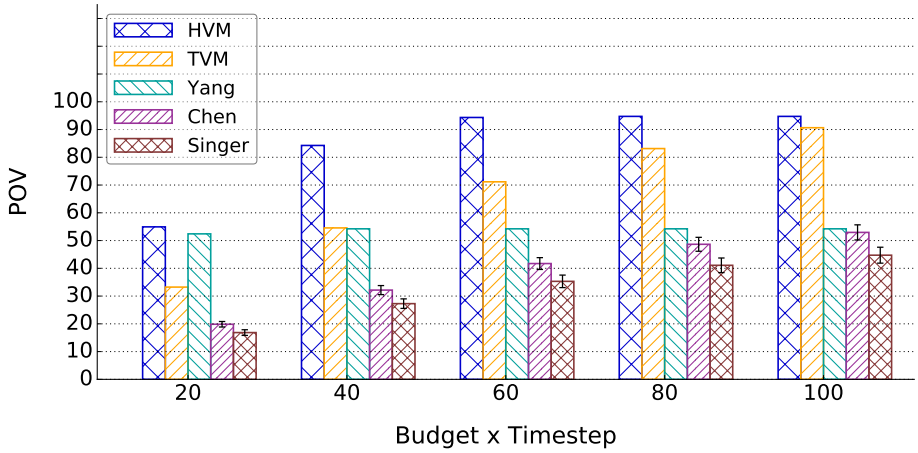
implemented. In the following, the following performance metrics will be used to describe performance:

- *Percentage of Obtained Value (POV):* If P^* is the optimum and P^m is the value obtained by mechanism m, then $POV^m = {}^{P^m}/_{P^*}$.
- *Speed-up:* given execution time E_1 and E_2, the speedup is defined as ${}^{E_1}/_{E_2}$. It will be used to compare the execution time of the mechanisms with the one obtained with running the MapReduce implementation on the Hadoop cluster.

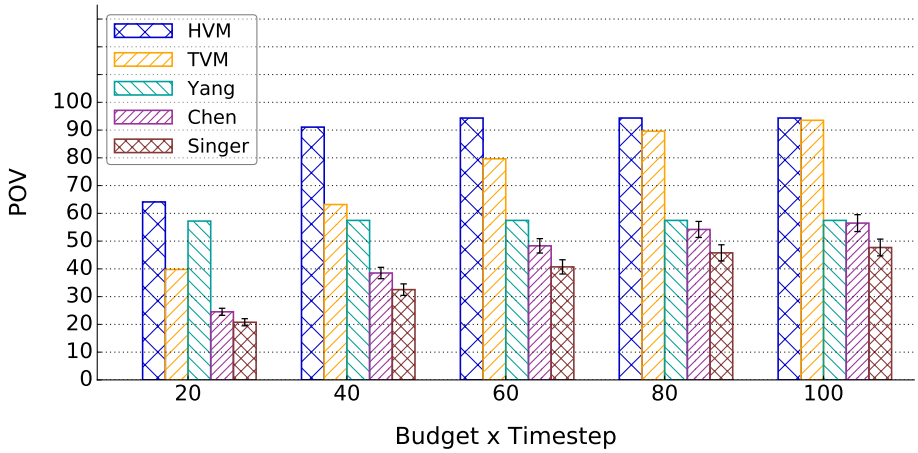
Since the target is efficiency and scalability, these metrics are believed to be sufficient and appropriate to evaluate the mechanisms. All experiments have been performed on a cloud computing system emulated by two Dell Precision T7610 servers, equipped with an Intel(R) Xeon(R) CPU E5-2680 v2 processor (20 cores, 2.80GHz, 64GB RAM), and by four Dell Optiplex(R) 7010 with Intel(R) Core(TM) i7-3770 CPU (7 cores, 3.40GHz, 8GB RAM). An Apache Hadoop cluster has been implemented on top of these machines to evaluate the scalability of the mechanisms.

4.4.2. Experimental Results. Figure 4.3 shows the average percentage of obtained value (POV) experimented by HVM, TVM, Singer, Chen and Yang as a function of the allocated budget per timestep, for the datasets San Francisco, Rome, and Beijing. From the plots the following conclusions can be derived:

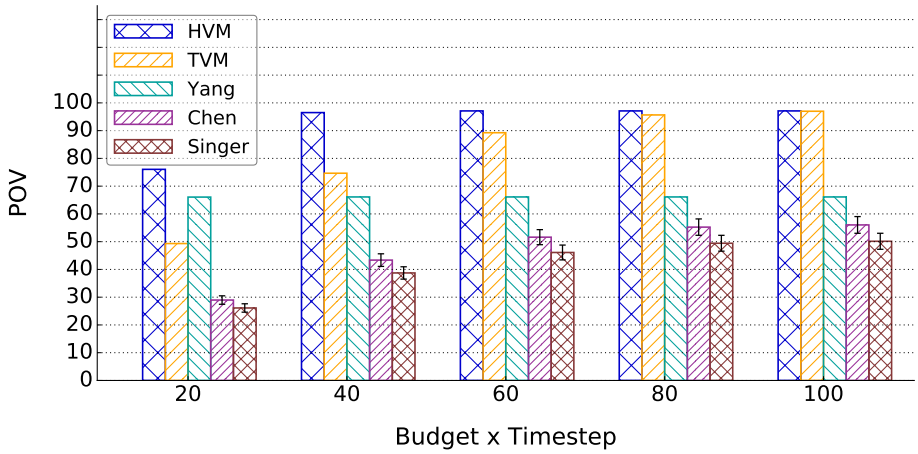
- First, it can be inferred that HVM performs better than the other algorithms, irrespective of the allocated budget per timestep. This is because HVM is significantly budget-effective as it uses almost all the available budget at each execution of the algorithm, while at the same time guaranteeing budget-feasibility. In turn, the other algorithms suffer from the early exit condition necessary to achieve budget-feasibility, which significantly affects the optimization performance.



(a) San Francisco



(b) Rome



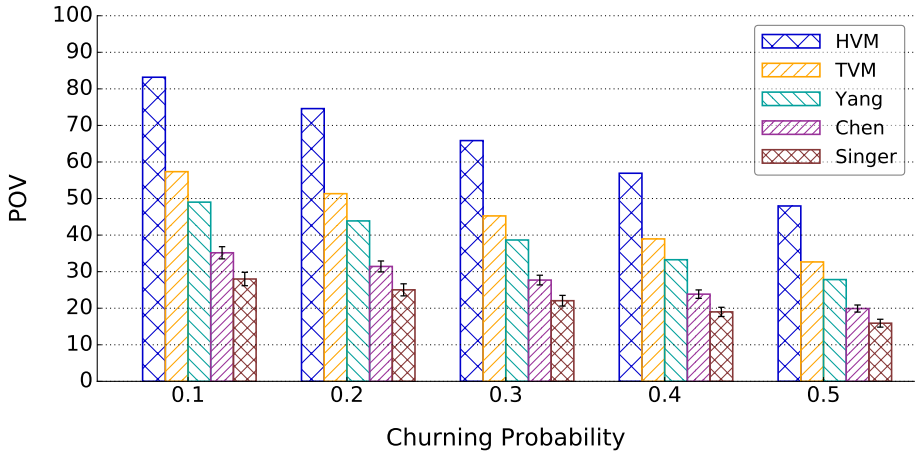
(c) Beijing

Figure 4.3. Budget vs. POV.

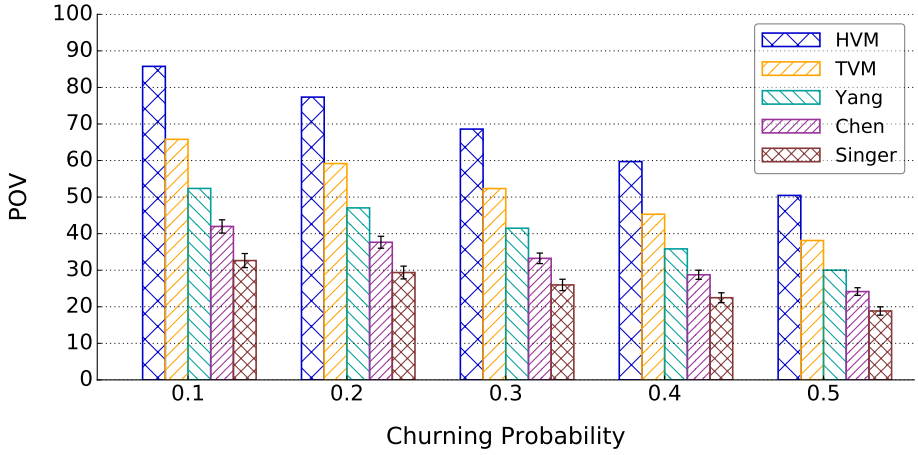
- Second, TVM increases its performance significantly as the available budget increases, which is consistent with what expected as the algorithm will terminate after selecting more participants.
- Third, the performance of Yang, Chen and Singer remains always below 70%, irrespective of the available budget. This is because Chen and Singer use a randomized selection condition to guarantee budget-feasibility, truthfulness and bound on performance, but at the same time decrease significantly their performance. The performance of Yang, in turn, remains almost constant as the algorithm neglects to consider participants' mobility. Overall, HVM increases the performances of the state of the art algorithms by a remarkable 30% (29.1%, average of 30.7%, 30.1.%, 26.7%).

To further validate HVM and TVM efficiency, Figure 4.4 shows the POV as a function of the probability that a bidder will not be able execute the sensing task after been assigned to it, defined for simplicity as churning probability (CP). This might happen, for example, because of problems in the smartphone equipment (i.e., battery exhausted, networking issues, or software issues), or because the participant is in a different sector of the sensing area at the relevant time (or also outside the sensing area). As expected, Figure 4.4 shows that the overall performance of all the algorithms decreases as this probability increases. However, HVM is more resilient than the other algorithms, as the increased efficiency in budget utilization allows to select more bidders and therefore allow more redundancy in the selection process (i.e., hire more participants). On the average, HVM achieves 35.2% more utility than the other algorithms, which makes it ideal in cases when the CP is high (e.g., very uncertain participants' mobility).

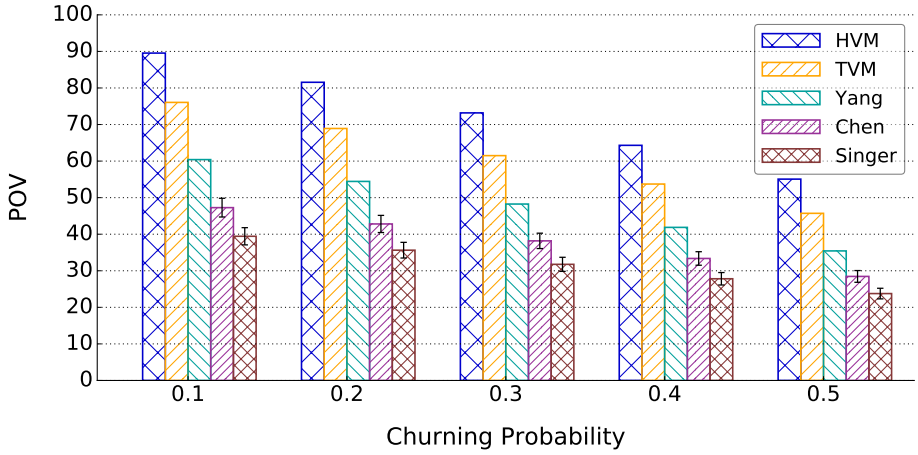
Finally, in order to demonstrate the scalability of HVM, Figure 4.5 shows the computation time as a function of the number of bidders and the number of mappers used in the MapReduce implementation of HVM. For clarity, the execution times



(a) San Francisco



(b) Rome



(c) Beijing

Figure 4.4. Churning Probability vs. POV.

were normalized by the same factor, so as to have 1 computation unit in the case the number of bidders is 200 and 1 mapper job is used.

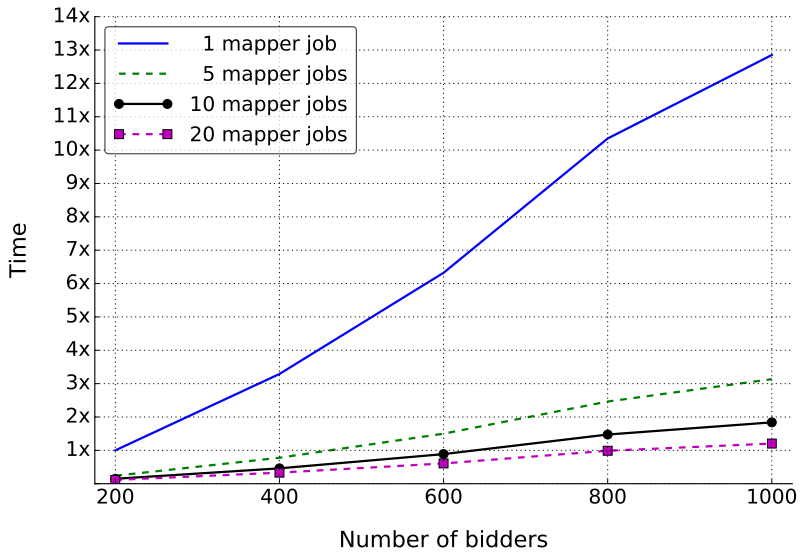


Figure 4.5. Execution time vs. Number of bidders.

Figure 4.5 concludes that the MapReduce framework is effective in providing scalability to the HVM algorithm, as the speedup provided by the additional mappers is linear; on the average, HVM gets a 12.28x speedup by passing from 1 to 20 mappers.

4.5. RELATED WORK

A significant number of recruitment mechanisms (also referred to as incentive mechanisms) for smartphone crowdsensing have been recently proposed, as surveyed in [31, 75]. Research has preferred focusing on designing auction-theoretical mechanisms rather than other strategies because of the provable theoretical results that auction theory offers.

One of the first mechanisms was presented in [38], where Jaimes *et al.* propose a mechanism that addresses the coverage problem with budget constraints (shown to

be NP-Hard) by greedily finding a set of users that covers the greatest possible area within a budget constraint. However, the mechanism fails to consider truthfulness. The problem of guaranteeing a truthful incentive mechanism was explored for the first time in [89]. In this paper, the authors propose a model where the system announces a set of sensing tasks, each one having a certain value to the system. Each user then selects a subset of tasks according to its preference and bids for each of them. The system then selects the participants so as to maximize a submodular value function. Although the mechanism shows important properties and achieves good performances, it does not consider budget-feasibility, participants' location nor mobility pattern.

More recently, Feng *et al.* proposed in [28] an incentive mechanism which takes into account the location of the smartphone users. Specifically, the tasks here are location-based, and users can bid only on tasks which are in the sensing coverage of the smartphone. After proving that optimally determining the winning bids with location awareness is NP-hard, the authors proposed mechanism consists of a polynomial time and near-optimal task allocation algorithm, as well as a payment scheme that guarantees truthfulness. The main limitation of [28] is that the sensing tasks and users' positions are assumed to be known in advance and static. A similar mechanism considering dynamic tasks and users has been proposed by the same authors in [29]. However, in both papers budget-feasibility has not been considered. The closest to this work are [16, 77, 78], where the authors investigated the problem of optimum design of budget-feasible auctions in a more general setting.

4.6. SUMMARY

In this section, mechanisms to incentivize the participation of users with uncertain mobility in smartphone crowdsensing were introduced. First, the system model, the related assumptions, and the Budgeted Value Maximization (BVM) problem were

introduced. It has been demonstrated that BVM is NP-Hard and that its objective function is submodular, and proposed two mechanisms satisfying the desired auction-theoretical property of truthfulness. The algorithms have been evaluated through experimental analysis and compared them with the relevant existing work. Results have shown that the algorithms outperform existing state of the art by more than 30% and are highly scalable.

5. CONCLUSION

Smart devices have revolutionized our lives and the way we interact with the surrounding environment. What was technologically a dream twenty years ago has now become reality, and with more and more users embedding pervasive technologies in their daily lives, an immense number of novel applications that drastically improve people's everyday life are now possible. Among all, smartphone crowdsensing is certainly one of the most promising paradigms, as it allows to decrease dramatically infrastructure costs and obtain detailed information about the phenomenon being monitored.

This work focuses on designing mechanisms for improving information quality in smartphone crowdsensing systems. First, a novel *Framework to optimize Information Reliability in Smartphone-based participaTory sensing* (FIRST) was developed, which leverages the collective action of mobile trusted participants (MTPs) to securely assess the reliability of sensing reports. The framework was evaluated through experiments leveraging real-world mobility traces of taxi cabs and through an implementation in iOS and Android. Experimental results demonstrate that the framework optimizes information reliability and outperforms state-of-the-art literature.

Next, this work studied the problem of maximizing the likelihood of successful execution of the sensing tasks when participants having uncertain mobility compete for offering their sensing services. Two incentive mechanisms based on game theory and having proven approximation ratio are proposed that solve such problem. Such mechanisms were evaluated by experimenting with real-world mobility traces, and experimental results demonstrate that the mechanisms outperform the state of the art.

However, there are a number of open research issues still left to investigate for further improving information quality in smartphone crowdsensing systems. For example, it is still an open research challenge understanding what incentive works *best* to motivate volunteers. More specifically, there is currently a lack of a general, empirical study on the motivations of volunteers to perform smartphone crowdsensing tasks and the effectiveness of smartphone crowdsensing incentives across different contexts. Furthermore, to date, incentives for smartphone crowdsensing have largely been applied in an *ad hoc*, "one size fits all" manner, assuming that all applications have the same requirements and people have all the same needs. Additional study is needed to develop a systematic approach to the design and selection of incentives that are *tailored* to a particular application, and *personalized* to motivate individual volunteers to perform smartphone crowdsensing tasks.

In addition, one of the limitations of existing work is that it does not provide a detailed definition of information quality, nor do they address the problem of *how* to determine the information quality of sensing reports. More specifically, they assume that information quality can be computed by using external functions tailored to such purpose. However, this might not be the case when *multimedia* data (e.g., audio/video feed, pictures) is being collected by the smartphone crowdsensing system. In particular, when the data is coming from physical sensors such as temperature, pressure, or light, the information quality may be defined as the accuracy of that measurement with respect to the real phenomenon being monitored. However, how do we define the information quality of a picture, of a sound, of a video? In such case, the specific context of the smartphone crowdsensing application may significantly influence the definition of information quality. Therefore, additional research is necessary to provide different definitions of information quality that are valid for the spectrum of smartphone crowdsensing systems nowadays available.

BIBLIOGRAPHY

[1] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, "Mobiscopes for human spaces," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 20–29, April 2007.

[2] A. Albers, I. Krontiris, N. Sonehara, and I. Echizen, "Coupons as monetary incentives in participatory sensing," in *Collaborative, Trusted and Privacy-Aware e/m-Services*, ser. IFIP Advances in Information and Communication Technology, C. Douligeris, N. Polemi, A. Karantjias, and W. Lamersdorf, Eds. Springer Berlin Heidelberg, 2013, vol. 399, pp. 226–237. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37437-1_19.

[3] H. Alzaid, E. Foo, J. G. Nieto, and E. Ahmed, "Mitigating On-Off Attacks in Reputation-based Secure Data Aggregation for Wireless Sensor Networks," *Security and Communication Networks*, vol. 5, no. 2, pp. 125–144, 2012. [Online]. Available: http://dx.doi.org/10.1002/sec.286.

[4] R. Amici, M. Bonola, L. Bracciale, A. Rabuffi, P. Loreti, and G. Bianchi, "Performance assessment of an epidemic protocol in VANET using real traces," *Procedia Computer Science*, vol. 40, pp. 92–99, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914014021.

[5] A. Andersson and C. Mattsson, "Dynamic interpolation search in o (log log n) time," pp. 15–27, 1993.

[6] K. D. Atherton, "Israeli students spoof Waze with fake traffic jam," http://tinyurl.com/p4gcgkv.

[7] X. Bao and R. Roy Choudhury, "Movi: Mobile phone based video highlights via collaborative sensing," in *Proceedings of the 2010 ACM International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 357–370. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814468.

[8] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava, "On the quality and value of information in sensor networks," *ACM Transactions on Sensor Networks*, vol. 9, no. 4, pp. 48:1–48:26, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2489253.2489265.

[9] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *Proceedings of the 2006 Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications*, ser. WSW '06, 2006, pp. 117–134.

[10] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing," in *Proceedings of the 2006 ACM Annual International Workshop on Wireless Internet*, ser. WICON '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1234161.1234179.

[11] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, "The rise of people-centric sensing," *IEEE Internet Computing*, vol. 12, no. 4, pp. 12–21, July 2008.

[12] CampusCrime, "Safety on your smartphone for you and your campus," http://www.campussafeapp.com, 2015.

[13] C. Castelfranchi and R. Falcone, *Trust theory: A socio-cognitive and computational model*. John Wiley & Sons, 2010, vol. 18.

[14] Y. Chae, L. C. DiPippo, and Y. L. Sun, "Trust Management for Defending On-Off Attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 1178–1191, April 2015.

[15] C. Chen and Y. Wang, "Sparc: Strategy-proof double auction for mobile participatory sensing," in *Proceedings of the 2013 International Conference on Cloud Computing and Big Data*, ser. CLOUDCOM-ASIA '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 133–140. [Online]. Available: http://dx.doi.org/10.1109/CLOUDCOM-ASIA.2013.99.

[16] N. Chen, N. Gravin, and P. Lu, "On the approximability of budget feasible mechanisms," in *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '11. SIAM, 2011, pp. 685–699. [Online]. Available: http://dl.acm.org/citation.cfm?id=2133036.2133090.

[17] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928 – 1946, 2011, mobile Applications: Status and Trends. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121211001701.

[18] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492.

[19] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time air quality monitoring through mobile sensing in metropolitan areas," in *Proceedings of the* 2^{nd} *ACM SIGKDD International Workshop on Urban Computing*, ser. UrbComp '13. ACM, 2013, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/2505821.2505834.

[20] E. D'Hondt, M. Stevens, and A. Jacobs, "Participatory noise mapping works! An evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring," *Pervasive and Mobile Computing*, vol. 9, no. 5, pp. 681–694, 2013, special Issue on Pervasive Urban Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119212001137.

[21] Y. F. Dong, S. Kanhere, C. T. Chou, and R. P. Liu, "Automatic image capturing and processing for petrolwatch," in *Proceedings of the 2011 IEEE International Conference on Networks*, ser. ICON '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 236–240. [Online]. Available: http://dx.doi.org/10.1109/ICON.2011.6168481.

[22] A. Dua, N. Bulusu, W.-C. Feng, and W. Hu, "Towards trustworthy participatory sensing," in *Proceedings of the 4th USENIX Conference on Hot Topics in Security*, ser. HotSec '09. USENIX, 2009, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855628.1855636.

[23] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *Proceedings of the 2012 IEEE International Conference on Computer Communications*, ser. INFOCOM '12, March 2012, pp. 1701–1709.

[24] ——, "Motivating smartphone collaboration in data acquisition and distributed computing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2320–2333, Oct 2014.

[25] E2z.com, "Waze touches 50M users globally; Malaysia, Indonesia in top 10 list," http://tinyurl.com/lsounox.

[26] Ericsson, "Ericsson mobility report, november 2014," http://tinyurl.com/p3uev7c.

[27] R. Fakoor, M. Raj, A. Nazi, M. Di Francesco, and S. K. Das, "An integrated cloud-based framework for mobile phone sensing," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 47–52. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342520.

[28] Z. Feng, Y. Zhu, Q. Zhang, L. Ni, and A. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proceedings of the 2014 IEEE International Conference on Computer Communications*, ser. INFOCOM '14, April 2014, pp. 1231–1239.

[29] Z. Feng, Y. Zhu, Q. Zhang, H. Zhu, J. Yu, J. Cao, and L. Ni, "Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones," in *Proceedings of the 2014 IEEE International Conference on Distributed Computing Systems*, ser. ICDCS '14, June 2014, pp. 11–20.

[30] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "GreenGPS: A participatory sensing fuel-efficient maps application," in *Proceedings of the 8th ACM International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '10. ACM, 2010, pp. 151–164. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814450.

[31] H. Gao, C. H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, and K. K. Leung, "A Survey of Incentive Mechanisms for Participatory Sensing," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 918–943, Second Quarter 2015.

[32] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt, "Micro-blog: Sharing and querying content through mobile phones and social participation," in *Proceedings of the 2008 ACM International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 174–186. [Online]. Available: http://doi.acm.org/10.1145/1378600.1378620.

[33] P. Gilbert, J. Jung, K. Lee, H. Qin, D. Sharkey, A. Sheth, and L. P. Cox, "Youprove: Authenticity and fidelity in mobile sensing," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '11. ACM, 2011, pp. 176–189. [Online]. Available: http://doi.acm.org/10.1145/2070942.2070961.

[34] K. Han, C. Zhang, and J. Luo, "Truthful scheduling mechanisms for powering mobile crowdsensing," *CoRR*, vol. abs/1308.4501, 2013. [Online]. Available: http://arxiv.org/abs/1308.4501.

[35] D. He, S. Chan, and M. Guizani, "User privacy and data trustworthiness in mobile crowd sensing," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 28–34, February 2015.

[36] K. L. Huang, S. S. Kanhere, and W. Hu, "Are you contributing trustworthy data?: The case for a reputation system in participatory sensing," in *Proceedings of the 2010 ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, ser. MSWIM '10. New York, NY, USA: ACM, 2010, pp. 14–22. [Online]. Available: http://doi.acm.org/10.1145/1868521.1868526.

[37] ——, "On the need for a reputation system in mobile phone based sensing," *Ad Hoc Networks*, vol. 12, pp. 130–149, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870511002174.

[38] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications*, ser. PerCom '12, March 2012, pp. 103–108.

[39] E. T. Jaynes, "Information theory and statistical mechanics," *Physical Review*, vol. 106, no. 4, pp. 620–630, 1957.

[40] N. Jensen, "From pastoralists to mechanical turks: Using the crowd to validate crowdsourced data," http://tinyurl.com/zwbhk3w.

[41] S. Ji and T. Chen, "Crowdsensing incentive mechanisms for mobile systems with finite precisions," in *Proceedings of the 2014 IEEE International Conference on Communications*, ser. ICC '14, June 2014, pp. 2544–2549.

[42] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 402–427, First 2013.

[43] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *Proceedings of the* 32^{th} *IEEE International Conference on Computer Communications*, ser. INFOCOM '13. IEEE, 2013, pp. 1402–1410.

[44] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, p. 19, 2012.

[45] A. Krause, E. Horvitz, A. Kansal, and F. Zhao, "Toward community sensing," in *Proceedings of the 2008 ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '08, April 2008, pp. 481–492.

[46] V. Krishna, *Auction theory*. Academic press, 2009.

[47] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, Sept 2010.

[48] Q. Li and G. Cao, "Providing privacy-aware incentives for mobile sensing," in *Proceedings of the 2013 IEEE International Conference on Pervasive Computing and Communications*, ser. PerCom '13, March 2013, pp. 76–84.

[49] ——, "Providing efficient privacy-aware incentives for mobile sensing," in *Proceedings of the* 34^{th} *IEEE International Conference on Distributed Computing Systems*, ser. ICDCS '14. IEEE, 2014, pp. 208–217.

[50] C. H. Liu, P. Hui, J. W. Branch, C. Bisdikian, and B. Yang, "Efficient network management for context-aware participatory sensing," in *Proceedings of the 2011 Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, ser. SECON'11, June 2011, pp. 116–124.

[51] LocationHolic and FakeLocation, Available respectively on AppStore (iOS) and Google Play (Android) app markets.

[52] T. Luo, S. S. Kanhere, S. K. Das, and T. Hwee-Pink, "Optimal prizes for all-pay contests in heterogeneous crowdsensing," in *Proceedings of the 2014 IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, ser. MASS '14, Oct 2014, pp. 1–9.

[53] T. Luo, S. S. Kanhere, and H.-P. Tan, "SEW-ing a simple endorsement web to incentivize trustworthy participatory sensing," in *Proceedings of the 11th IEEE Annual Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, ser. SECON '14. IEEE, 2014, pp. 636–644.

[54] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *Proceedings of the* 33^{th} *IEEE International Conference on Computer Communications*, ser. INFOCOM '14. IEEE, 2014, pp. 127–135.

[55] T. Luo and C.-K. Tham, "Fairness and social welfare in incentivizing participatory sensing," in *Proceedings of the 2012 Annual IEEE Communications Society Conference onSensor, Mesh and Ad Hoc Communications and Networks*, ser. SECON '12, June 2012, pp. 425–433.

[56] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 29–35, Aug 2014.

[57] C. Marforio, A. Francillon, S. Capkun, S. Capkun, and S. Capkun, *Application Collusion Attack on the Permission-based Security Model and its Implications for Modern Smartphone Systems*. Technical Report, Department of Computer Science, ETH Zurich Zürich, Switzerland, 2011.

[58] E. A. McCartney, K. J. Craun, E. Korris, D. A. Brostuen, and L. R. Moore, "Crowdsourcing the national map," *Cartography and Geographic Information Science*, vol. 42, no. sup1, pp. 54–57, 2015. [Online]. Available: http://dx.doi.org/10.1080/15230406.2015.1059187.

[59] D. Méndez, A. J. Perez, M. A. Labrador, and J. J. Marron, "P-sense: A participatory sensing system for air pollution monitoring and control," in *Proceedings of the* 9^{th} *IEEE International Conference on Pervasive Computing and Communications Workshops*, ser. PERCOM Workshops. IEEE, 2011, pp. 344–347.

[60] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng, "Truth discovery on crowd sensing of correlated entities," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: ACM, 2015, pp. 169–182. [Online]. Available: http://doi.acm.org/10.1145/2809695.2809715.

[61] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: ACM, 2015, pp. 183–196. [Online]. Available: http://doi.acm.org/10.1145/2809695.2809719.

[62] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Cenceme: Injecting sensing presence into social networking applications," in *Proceedings of the 2007 European Conference on Smart Sensing and Context*, ser. EuroSSC'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1–28. [Online]. Available: http://dl.acm.org/citation.cfm?id=1775377.1775379.

[63] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," *Proceedings of the 2008 ACM Conference on Embedded Networked Sensor Systems*, pp. 323–336, 2008.

[64] H. Mousa, S. B. Mokhtar, O. Hasan, O. Younes, M. Hadhoud, and L. Brunie, "Trust management and reputation systems in mobile participatory sensing applications: A survey," *Computer Networks*, vol. 90, pp. 49 – 73, 2015, crowdsourcing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128615002340.

[65] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[66] S. Nawaz, C. Efstratiou, and C. Mascolo, "ParkSense: A smartphone based sensing system for on-street parking," in *Proceedings of the* 19^{th} *ACM Annual International Conference on Mobile Computing & Networking*, ser. MobiCom '13. ACM, 2013, pp. 75–86. [Online]. Available: http://doi.acm.org/10.1145/2500423.2500438.

[67] S. Nawaz and C. Mascolo, "Mining users' significant driving routes with low-power sensors," in *Proceedings of the 12th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '14. ACM, 2014.

[68] N. Nikzad, N. Verma, C. Ziftci, E. Bales, N. Quick, P. Zappi, K. Patrick, S. Dasgupta, I. Krueger, T. v. Rosing, and W. G. Griswold, "Citisense: Improving geospatial environmental assessment of air quality using a wireless personal exposure monitoring system," in *Proceedings of the* 2^{nd} *Conference on Wireless Health*, ser. WH '12. ACM, 2012, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/2448096.2448107.

[69] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao, "MusicalHeart: A hearty way of listening to music," in *Proceedings of the* 10^{th} *ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. ACM, 2012, pp. 43–56. [Online]. Available: http://doi.acm.org/10.1145/2426656.2426662.

[70] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge University Press Cambridge, 2007, vol. 1.

[71] L. F. Perrone and S. C. Nelson, "A Study of On-Off Attack Models for Wireless Ad Hoc Networks," in *2006* 1^{st} *Workshop on Operator-Assisted (Wireless Mesh) Community Networks*, Sept 2006, pp. 1–10.

[72] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, "A parsimonious model of mobile partitioned networks with clustering," in *Proceedings of the* 1^{st} *International Conference on COMmunication Systems and NETworkS*, ser. COMSNETS '09, 2009. [Online]. Available: http://www.comsnets.org.

[73] R. Rana, C. T. Chou, N. Bulusu, S. Kanhere, and W. Hu, "Ear-phone: A context-aware noise mapping using smart phones," *Pervasive and Mobile Computing*, vol. 17, pp. 1–22, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214000273.

[74] F. Restuccia and S. K. Das, "FIDES: A trust-based framework for secure user incentivization in participatory sensing," in *Proceedings of the* 15^{th} *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*, ser. WoWMoM '14. IEEE, 2014, pp. 1–10.

[75] F. Restuccia, S. K. Das, and J. Payton, "Incentive mechanisms for participatory sensing: Survey and research challenges," *ACM Transactions on Sensor Networks*, vol. 12, no. 2, pp. 13:1–13:40, Apr. 2016. [Online]. Available: http://doi.acm.org/10.1145/2888398.

[76] S. Saroiu and A. Wolman, "I Am a Sensor, and I Approve This Message," in *Proceedings of the* 11^{th} *Workshop on Mobile Computing Systems & Applications*, ser. HotMobile '10. ACM, 2010, pp. 37–42. [Online]. Available: http://doi.acm.org/10.1145/1734583.1734593.

[77] Y. Singer, "Budget feasible mechanisms," in *2010* 51^{st} *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2010, pp. 765–774.

[78] A. Singla and A. Krause, "Truthful incentives for privacy tradeoff: Mechanisms for data gathering in community sensing," in *Proceedings of the ICML Workshop: Machine Learning Meets Crowdsourcing*, 2013.

[79] C.-K. Tham and T. Luo, "Fairness and social welfare in service allocation schemes for participatory sensing," *Computer Networks*, vol. 73, no. 0, pp. 58 – 71, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614002679.

[80] J. Thebault-Spieker, "Crowdsourced participatory sensing: applications and motivation of work," 2012.

[81] Thermodo, "Thermodo temperature sensor," 2014. [Online]. Available: http://www.thermodo.com.

[82] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 2009 ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. New York, NY, USA: ACM, 2009, pp. 85–98. [Online]. Available: http://doi.acm.org/10.1145/1644038.1644048.

[83] Y. Ueyama, M. Tamai, Y. Arakawa, and K. Yasumoto, "Gamification-based incentive mechanism for participatory sensing," in *Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communications Workshops*, ser. PERCOM Workshops '14, March 2014, pp. 98–103.

[84] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Artsense: Anonymous reputation and trust in participatory sensing," in *Proceedings of the 2013 IEEE International Conference of Computer Communications*, ser. INFOCOM '13, April 2013, pp. 2517–2525.

[85] ——, "Enabling reputation and trust in privacy-preserving mobile sensing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2777–2790, 2014.

[86] X. Wang, K. Govindan, and P. Mohapatra, "Collusion-resilient quality of information evaluation based on information provenance," in *Proceedings of the 2011 Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, ser. SECON '11, June 2011, pp. 395–403.

[87] Waze, "The Waze traffic monitoring application," http://www.waze.com.

[88] C. Xu, S. Li, Y. Zhang, E. Miluzzo, and Y. farn Chen, "Crowdsensing the speaker count in the wild: implications and applications," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 92–99, October 2014.

[89] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th ACM Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '12. ACM, 2012, pp. 173–184. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348567.

[90] ——, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–13, 2015.

[91] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "T-Drive: Enhancing driving directions with taxi drivers' intelligence," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 220–232, Jan 2013.

[92] A. Zhan, M. Chang, Y. Chen, and A. Terzis, "Accurate caloric expenditure of bicyclists using cellphones," in *Proceedings of the* 10^{th} *ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. ACM, 2012, pp. 71–84. [Online]. Available: http://doi.acm.org/10.1145/2426656.2426664.

[93] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14. New York, NY, USA: ACM, 2014, pp. 703–714. [Online]. Available: http://doi.acm.org/10.1145/2632048.2632059.

[94] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *Proceedings of the 2014 IEEE International Conference on Computer Communications*, ser. INFOCOM '14, April 2014, pp. 1213–1221.

[95] P. Zhou, Y. Zheng, and M. Li, "How long to wait? predicting bus arrival time with mobile phone based participatory sensing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1228–1241, June 2014.

VITA

Francesco Restuccia was born in Locri, Calabria, Italy. He received his B.Eng. and M.Eng. with highest honors in Computer Engineering from the University of Pisa, Tuscany, Italy, in 2009 and 2011, respectively. He has been a research assistant at IIT-CNR, Pisa, Italy, from November 2011 to August 2012. Francesco joined University of Texas at Arlington as a PhD candidate in 2012. In 2013, he transferred to Missouri University of Science and Technology, where he earned his Doctorate of Philosophy in Computer Science in December 2016. As a PhD student, Francesco worked under Prof. Sajal K. Das. His research interests were in mobile and pervasive computing, wireless sensor networks, and modeling of complex systems. Francesco has published several papers in top journals and conferences, some of which are listed with the references of this research. Francesco has been a student member of the Institute of Electrical and Electronics Engineers (IEEE) since 2015.