Limiting the Influence of Low Quality Information in Community Sensing

Goran Radanovic
Ecole Polytechnique Federale de Lausanne
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
goran.radanovic@epfl.ch

Boi Faltings
Ecole Polytechnique Federale de Lausanne
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
boi.faltings@epfl.ch

ABSTRACT

We consider a community of private sensors that collect measurements of a physical phenomenon, such as air pollution, and report it to a center. The center should be able to prevent low quality reports from degrading the quality of the aggregated information, as there are numerous reasons for operators to inject false sensor data. Hence, it is necessary to track the quality of the sensors over time in order to filter out low quality and malicious reports. To achieve this, we construct a reputation system with a guaranteed bounds on negative impact that malicious sensors can cause, and we evaluate its performance on a realistic dataset.

Keywords

Reputation Systems; Community Sensing; Online Learning

1. INTRODUCTION

Sensors that monitor important environmental phenomena are becoming smaller, cheaper and more ubiquitous. When such sensing units become more affordable and widely adopted by individuals, we arrive at *participatory*, *community* or *crowd* sensing [5, 1], where different entities, public or private, operate sensors and report the measurements to a *center*. The center interprets the reported data and publishes its results.

We consider a community sensing scenario where the center aggregates crowdsensed information in an online manner, from both public and private sensors, to provide real time estimates of air pollution over a certain urban area. In this scenario, the center controls a few accurate sensors that provide spatially or temporally sparse measurements (e.g., very accurate particle sensors are slow; similarly NO_2 can be sensed chemically but it's again slow and expensive), so to properly monitor the localized features of air pollution, it complements its own measurements with those obtained by crowd-participants who own ubiquitous sensor devices.

Figure 1 depicts the particularities of our setting. At the beginning, the center has only prior information about air pollution over an urban area. After some time, the center receives a report and merges it with the current pollution map P using pollution model $\mathcal{M}$. This process repeats until a trusted sensor reports its measurement, after which the center can evaluate the reports of crowd-sensors. We consider this to be one period of sensing and we denote it by t. The crowd sensing process then continues in the same manner until the period $t = T$, which we call *sensing time*.

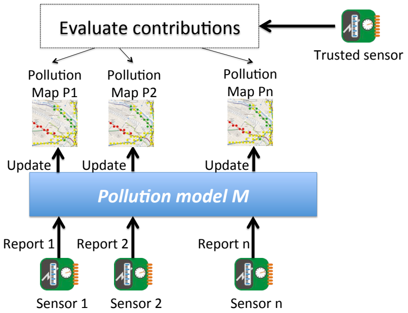


Figure 1: Crowd sensing scenario with online information fusion.

One of the main challenges in the described scenario is how to cope with untrustworthy information. While this issue has been partly addressed by the incentive mechanism design [16, 17], such an approach reaches its limit of effectiveness when a sensor owner intends to be malicious and intentionally misreports values. For example, a factory owner who wants to hide her own pollution traces could install sensors that misreport values of pollution. Therefore, a more rigorous approach is needed in order to identify faulty or malicious sensors. Reputation systems provide such an approach: bad reports lead to low reputation, which limits the influence of the later reports.

Existing reputation systems model the trustworthiness of sensors using reputation scores, and to determine which sensors are trustworthy, they compare the reputation scores with a predefined threshold (e.g. [3, 9]). These reputation systems provide only guarantees for particular reporting strategies, and can be easily manipulated, as we demonstrate in the paper.

Our reputation model is inspired by the *influence limiter* algorithm, primarily designed for recommender systems [20]. The influence limiter is provably resistant to misreporting, but its applicability is limited due to the significant amount of discarded data in the reputation boosting period. In pollution monitoring, sensors are expected to provide a large number of reports, so the information loss plays much smaller role than in recommender systems. However, the design of the influence limiter is not suitable for our setting, which we show in the paper.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

Our main contribution is a novel reputation system, called *Community Sensing Influence Limiter* (CSIL), suitable for real time community sensing scenarios where measurements are aggregated in an online manner. We show that the CSIL algorithm has provable guarantees on the direct negative impact that any malicious strategy could have. To the best of our knowledge, no technique with such guarantees has ever been proposed for the considered setting. Moreover, we evaluate the performance of the CSIL algorithm on a realistic test-bed to confirm its the theoretical properties and show its advantages over the existing reputation systems.

2. PRELIMINARIES

2.1 Sensors

In our setting, the center controls a small number of sensors, that we refer to as *trusted* sensors; these sensors are assumed to report accurate measurements. Other sensors are in control of private entities, and the center does not know their character, i.e., whether they are malicious or not. *Non-malicious* sensors are considered to be strategic — rational agents that aim to maximize their scores — or honest, while *malicious* sensors do not respond to incentives and their goal is to lower the quality of produced pollution maps. In the group of malicious sensors, we can also put *faulty* sensors that are not intentionally malicious, but do provide inaccurate data. Furthermore, notice that a malicious sensor might report accurately in some sensing periods in order to deceive the center. This means that the decision on how to use a sensor's reports in the information fusion process should be done by monitoring the behaviour of the sensor over the whole sensing time T.

Each sensor repeatedly performs a measurement $X \in [0, max]$ and reports value $Y \in [0, max]$, where $max \in \mathbb{R}$ is a maximum value that is reasonable to measure and report. If a sensor is honest, its report is equal to its measurement, i.e., $Y = X$. In general, however, the value of a report depends on both the sensor's measurement and its reporting strategy. Clearly, apart from their measurements, sensors also report their location, which we do not explicitly emphasize in the further text. As mentioned in the introduction, a time period t is defined by the arrival of a trusted report, which we assume to be stochastic. To simplify the description of our algorithm, we impose three conditions for sensors not controlled by the center: a sensor reports one measurement per time period, measurements between two time periods are statistically independent, and reports from different sensors arrive stochastically one at a time.

Strategy space. We make restrictions to the strategic space of malicious agents by assuming that their reports do not have a significant impact on the quality of the information provided by non-malicious sensors. As noted by [20], the restriction to *myopic strategies* is not a trivial assumption, but still allows a large scope of possible misreporting strategies, including strategies where malicious sensors change their reporting behaviour over time. Furthermore, it is likely that non-myopic strategies require complex implementation. For example, an effective malicious strategy that is based on the report sequence would require information about when sensing periods start/end. Since each sensing period ends when a trusted report is submitted, the center can easily obscure the starting point of a sensing period by, for example, not immediately notifying sensors of their reputation change. This also provides a justification for the assumption of stochastic arrival of reports.

2.2 Pollution Model

The center's goal is to construct and publish a pollution map P using the current set of measurements. We are particularly interested in a real-time updating, where pollution map P is updated after receiving each measurement using a pollution model, denoted by $\mathcal{M}$. We keep a general form of pollution model $\mathcal{M}$, where the input is defined by a finite set of reported measurements $\{Y_1, Y_2, ...\}$, while the output is a pollution map. Pollution map P contains either levels of pollution at the points of interest, or in case of probabilistic pollution models, probability distribution functions over the levels of pollution at the points of interest. Pollution model $\mathcal{M}$ is, thus, assumed to capture well correlations among measurements taken at different locations.

Since we want to keep a possibility of having a very general pollution model $\mathcal{M}$, we consider it as a *black box*. This implies that after receiving report Y_s from sensor s, the center should decide whether to publish a new pollution map P_s^{new} obtained by incorporating report Y_s into the existing pollution map P_s^{old}, or to keep the existing pollution map as its output. The rationale behind this is that the pollution map updating should be computationally efficient. For example, the updating procedure of the classical influence limiter is not consistent with this view, as a new output is a linear combination of P_s^{new} and P_s^{old}. However, a proper procedure for obtaining P_s^{new} in the influence limiter has exponential time complexity in the number of sensor, as we argue later on in the paper.

2.3 Evaluating Sensors

We evaluate sensors by their marginal contributions to the quality of produced pollution maps. More precisely, consider a pollution map P_s^{new} obtained by fusing sensor s's report with a pollution map P_s^{old} that preceded the report of sensor s. Furthermore, let $S(P, X_{trust})$ be a general scoring function that evaluates the quality of a pollution map with respect to the report $Y_{trust} = X_{trust}$ of a trusted sensor, and let it take values from a bounded interval $[-c/2, c/2]$. For example, a scoring function can be a quadratic scoring rule defined by equation (2) (in Section 5.2). The score of sensor s is then defined by the gain G_s of the center when it fully incorporates the sensor's report into the existing pollution map P_s^{old}:

$$score_s = G_s = \frac{1}{c}[S(P_s^{new}, X_{trust}) - S(P_s^{old}, X_{trust})]$$

It is easy to see that the score takes values in $score_s \in [-1, 1]$. Notice that the center needs not to publish pollution map P_s^{new} - this decision is separated from the decision on how to score sensor s. The score can further be used to define (monetary) incentives given to the sensor.

2.4 Myopic impact

Following the approach from [20], we use the notion of sensor s's *myopic impact*. Since our main method probabilistically decides whether to accept or discard sensor s's report, we define the *expected myopic impact* of sensor s at time period t as:

$$\bar{\Delta}_{s,t} = \pi_{update} \cdot G_{s,t} + (1 - \pi_{update}) \cdot 0 = \pi_{update} \cdot G_{s,t}$$

where π_{update} is the probability of incorporating sensor s's report into the existing output. The intuition behind the definition is straightforward. Whenever the center *accepts* to fuse sensor s's report into the existing pollution map, the sensor's impact is equal to the center's information gain: $G_{s,t} = \frac{1}{c}[S(P_s^{new}, X_{trust}) - S(P_s^{old}, X_{trust})]$. Otherwise, when the center decides to *discard* sensor s's report, the sensor's impact is 0 because it does not change the center's output P_s^{old}. Finally, we define the expected *total myopic impact* as $\bar{\Delta}_s = \sum_{t=1}^{T} \bar{\Delta}_{s,t}$. Notice that the myopic impacts are functions of $G_{s,t}$. Since $G_{s,t}$ is a random variable, we can associate expected values over $G_{s,t}$ for both $\bar{\Delta}_{s,t}$ and $\bar{\Delta}_s$, which we denote by $\mathbb{E}(\bar{\Delta}_{s,t})$ and $\mathbb{E}(\bar{\Delta}_s)$, respectively.[1]

3. RELATED WORK

The standard approach of dealing with untrustworthy information in sensing is by using reputation systems [14, 3, 9, 4, 24, 6], with the Beta reputation system [8] being the most common way of assigning reputation scores. While in the literature one can find other ways of assigning reputation scores, such as using the Gompertz function [13], the classification of whether a sensor misbehaves is typically based on a simple thresholding principle: if the reputation of a sensor is lower than a certain threshold, the sensor is denoted as misbehaving, otherwise, it is considered to be trustworthy. A thresholding approach is common even among techniques that do not necessarily use reputation systems (e.g., [23]). While such a thresholding principle can cope with simple attacks where malicious sensors report consistently wrong values, it fails to protect the center against deceiving attacks, as we describe it later in the paper.

[22] and [19] take a different approach to fuse information from multiple sensors that are not a priori assumed to be trustworthy. [22] tries to learn the parameters related to the trustworthiness using a maximum likelihood method over the assumed (Gaussian) model with unknown parameters. [19] proposes a two stage Bayesian multi-sensor fusion algorithm that incorporates model of sensors' trustworthiness. Neither of the two multi-sensor fusion methods have provable guarantees on the loss of the system experienced when the majority of sensors is untrustworthy and potentially malicious. As alternatives to reputation systems, we also mention hardware solutions, such as trusted platform modules (e.g., [21, 10]). These approaches, however, require additional hardware on each sensing module, which limits their applicability.

3.1 Current Approach

Let us now describe the thresholding approach. When the center receives a report $Y_{s,t}$ of sensor s, it fuses the report with the existing information if sensor s's reputation is greater than a certain classification threshold Θ, and otherwise discards it. The approach is depicted by Algorithm 1. Function $Update(P, Y_{s,t})$ uses the existing set of *included* reports (the set of reports that produced pollution map P), adds to it report $Y_{s,t}$, and applies model $\mathcal{M}$ to obtain a new pollution map. *RepUpdate* updates the reputation of sensor s using $score_{s,t}$, and has one condition: if $score_{s,t}$ has a strictly positive constant value, the reputation converges to its maximum value; if $score_{s,t}$ has a strictly negative constant value, the reputation converges to its minimum value.

```
Data: Initial reputation $\rho_0$, threshold $\Theta$
begin
    for Sensor s do
        $\rho_{s,1} \longleftarrow \rho_0$;
    end
    for $t = 1$ to $t = T$ do
        Compute prior map $P$;
        Publish $P$;
        for Sensor s do
            Receive $s$'s report $Y_{s,t}$;
            $P_s^{old} \longleftarrow P$ ;
            $P_s^{new} \longleftarrow Update(P, Y_{s,t})$ ;
            if $\rho_{s,t} \geq \Theta$ then
                $P \longleftarrow P_{new}^{s}$;
                Publish $P$;
            end
        end
        Receive report $Y_{trust,t} = X_{trust,t}$;
        for Sensor s do
            $score_{s,t} \longleftarrow$
            $\frac{1}{c}[S(P_s^{new}, X_{trust,t}) - S(P_s^{old}, X_{trust,t})]$;
            $\rho_{s,t+1} \longleftarrow RepUpdate(\rho_{s,t}, score_{s,t})$;
        end
    end
end
```

Algorithm 1: Thresholding

This simple reputation system can be considered to be a part of a large family of reputations systems that use fix thresholds to classify whether a certain sensor misbehaves or not. These reputation systems can cope with simple attacks where malicious sensors report consistently wrong values. For example, they can limit the effectiveness of the malicious strategy that consists of reporting low pollution values. However, they fail to protect the system against deceiving attacks.

One particular deceiving strategy of a malicious sensor could be to report informative values when its reputation is below threshold Θ, while report low quality information when its reputation is above the threshold. The intuition behind this attack is that a sensor reports useful information only when the center does not use it, and when the center uses its information, it deliberately misreports.

Proposition 1. *Consider a pollution model $\mathcal{M}$ that allows arbitrary generation of gains $G_{s,t}$ related to sensor s. Then, there exists a sequence of gains such that the total myopic impact $\bar{\Delta}_s$ of sensor s in Algorithm 1 is negative and monotonically decreases with T, i.e., $\lim_{T\to\infty} \bar{\Delta}_s = -\infty$.*

Proof. Consider a sequence of gains such that whenever $\rho_{s,t} < \Theta$, gain $G_{s,t}$ is equal to $G_{s,t} = g > 0$, while $\rho_{s,t} \geq \Theta$ implies negative gain $G_{s,t} = -g < 0$. In other words, $\pi_{update} = 1$ for $G_{s,t} < 0$ and $\pi_{update} = 0$ for $G_{s,t} \geq 0$. Since reputations converge to the maximum possible reputation if $score_{s,t}$ (i.e., $G_{s,t}$) is fixed to $g > 0$, we know that $\rho_{s,t}$ will infinitely often be greater than Θ for $T \to \infty$. Therefore, $\bar{\Delta}_s$ is negative (because $\pi_{update} = 0$ for $G_{s,t} \geq 0$) and $\lim_{T\to\infty} \bar{\Delta}_s = -\infty$ (because $\rho_{s,t} \geq \Theta$ infinitely often). □

In Section 5.5, we simulate such a behaviour to show that the thresholding does not prevent the center from experiencing an unbounded negative influence.

[1] $\mathbb{E}(\bar{\Delta}_s)$ is the expectation over gains from all time periods.

3.2 The Influence Limiter

The influence limiter, when transformed to our setting, has the same skeleton structure as the thresholding algorithm with the main differences in three components, which we point out in this section. We show, however, that all the three components should be modified in order to obtain a practical algorithm.

Information fusion. The standard version of the influence limiter has a deterministic information fusion component. In particular, the influence limiter incorporates all the reports, but assigns different weights to different reports. In our scenario, this would mean that when a report from a sensor s is received, the new pollution map P_s^{new} is calculated and the published pollution map P is updated to:

$$P \longleftarrow (1 - w_{s,t}) \cdot P_s^{old} + w_{s,t} \cdot P_s^{new} \qquad (1)$$

Here, the weight is equal to $w_{s,t} = \min(\rho_{s,t}, 1)$. The crucial part of the algorithm is how P_s^{new} should be calculated, i.e., the structure of the *Update* function.

In the influence limiter, a sensible updating function has to include the fact that all reports are fused, but with different weights. Since pollution model $\mathcal{M}$ is assumed to be a *black box*, one has to additionally ensure that the reports are properly weighted (limited) when updating pollution map P. For example, consider two reports Y_{s1} and Y_{s2} that arrive sequentially. Initially, P should be set to $\mathcal{M}(\emptyset)$. Once Y_{s1} is reported, the update of P, denoted by P_1, is easy to calculate: we simply make a linear combination of P and $\mathcal{M}(\{Y_1\})$, with weights $1 - w_1$ and w_1 (see (1)). The problem, however, arises when we update the current pollution map P_1 for report Y_{s2}. Namely, the new update should be a linear combination of the current pollution map P_1 and the pollution map P_{s2}^{new} that does not limit Y_{s2}, but does appropriately limit the reports that had arrived before Y_{s2}. In our case, the limited report in P_{s2}^{new} would be Y_{s1}. Since Y_{s1} should in P_{s2}^{new} be limited in the same way as in P_1 (otherwise report Y_{s2} has influence on the limiting process of prior information), we obtain that P_{s2}^{new} is equal to $P_{s2}^{new} \longleftarrow (1 - w_1) \cdot \mathcal{M}(\{Y_2\}) + w_1 \cdot \mathcal{M}(\{Y_1, Y_2\})$. Now, notice that for report Y_{s1} we only needed to query model $\mathcal{M}$ once because there were no prior reports. For report Y_{s2}, we needed to query model $\mathcal{M}$ twice. This can be easily generalized; for example, for the third report Y_{s3}, we would need to query model $\mathcal{M}$ four times to obtain pollution maps: $\mathcal{M}(\{Y_{s3}\})$, $\mathcal{M}(\{Y_{s1}, Y_{s3}\})$, $\mathcal{M}(\{Y_{s2}, Y_{s3}\})$ and $\mathcal{M}(\{Y_{s1}, Y_{s2}, Y_{s3}\})$. By using induction, we prove the following claim:

PROPOSITION 2. *The number of queries to a black box model $\mathcal{M}$ of the influence limiter algorithm in one time period t is $\Omega(2^n)$, where n is the number of reported values.*

Scoring rule. The properties of the influence limiter are proven only for a *quadratic scoring rule* (see Lemma 5 in [20]). Since our goal is not to make restrictions on the form of model $\mathcal{M}$, allowing general scoring techniques is crucial in our design. For example, if a model $\mathcal{M}$ is non-probabilistic, a quadratic scoring rule is not applicable.

Furthermore, the influence limiter uses a binary outcome in its scoring rule (this is a requirement of Lemma 5 in [20]). In our scenario, the report $Y_{trust} = X_{trust}$ of a trusted sensor is a real number, so one needs to transform it into a binary variable in order to apply it in the influence limiter. This can be done by defining a threshold and a binary variable equal to 0 if X_{trust} is smaller than the threshold, and 1 otherwise. An issue with this approach is that the evaluation process is much less accurate. For example, if the threshold is equal to 30, then this scoring technique would assign the same quality evaluations for both $X_{trust} = 35$ and $X_{trust} = 50$.

Reputation update. The reputation updating rule of the influence limiter is defined by $\rho_{s,t+1} \longleftarrow \rho_{s,t} + w_{s,t} \cdot score_{s,t}$, and resembles the information fusion updating.

In our approach, which is described in the following section, we use a *non-deterministic* information fusion to lower the query complexity and we allow general scoring rules based on non-binary outcomes. These changes also imply a different reputation updating rule. All these structural differences point out that the influence limiter is not trivially transformable to our setting.

4. COMMUNITY SENSING INFLUENCE LIMITER

The Community Sensing Influence Limiter (CSIL) is a version of the influence limiter reputation system with an exponential reputation boosting. Furthermore, it shares some similarities with randomized weighted majority algorithms (e.g., see [15]), where the decision making rule is non-deterministic and uses weights that have a multiplicative updating rule.

```
Data: Initial reputation $\rho_0 > 0$
begin
    for Sensor $s$ do
        $\rho_{s,1} \longleftarrow \rho_0$;
    end
    for $t = 1$ to $t = T$ do
        Compute prior map $P$;
        Publish $P$;
        for Sensor $s$ do
            Receive $s$'s report $Y_{s,t}$;
            $P_s^{old} \longleftarrow P$ ;
            $P_s^{new} \longleftarrow Update(P, Y_{s,t})$ ;
            if $rand(0,1) < \frac{\rho_{s,t}}{\rho_{s,t+1}}$ then
                $P \longleftarrow P_s^{new}$;
                Publish $P$;
            end
        end
        Receive report $Y_{trust,t} = X_{trust,t}$;
        for Sensor $s$ do
            $score_{s,t} \longleftarrow$
            $\frac{1}{c}[S(P_s^{new}, X_{trust,t}) - S(P_s^{old}, X_{trust,t})]$;
            $\rho_{s,t+1} \longleftarrow \rho_{s,t} \cdot (1 + \frac{1}{2} \cdot score_{s,t})$;
        end
    end
end
```

Algorithm 2: Community Sensing Influence Limiter

The exact description of CSIL can be found in Algorithm 2, and it has the following steps. Initially, sensors' reputations are set to $\rho_0 > 0$. At time period t, upon the arrival of a sensor s's report, the reputation system calculates pollution map P_s^{new} using function $Update(P, Y_{s,t})$, which adds report $Y_{s,t}$ to the existing set of *included* reports (the set of reports that produced pollution map P) and applies model $\mathcal{M}$ to obtain a new pollution map. In the next step, the algorithm decides whether the current pollution map should be replaced with the update or not. The decision is probabilistic — with probability equal to $\frac{\rho_{s,t}}{\rho_{s,t}+1}$, the center sets pollution map P to P_s^{new}, while otherwise, it discards sensor s's report. The final step of the repetitive algorithm is to update the reputation of sensor s when the report $Y_{trust} = X_{trust}$ of a trusted sensor is received. The reputation updating rule assigns a new reputation to sensor s by adding to the current reputation $\rho_{s,t}$ the score of sensor s modulated by $\frac{\rho_{s,t}}{2}$.

4.1 Theoretical Analysis

Since the deterministic information fusion rule of the standard influence limiter has an exponential query complexity, we have applied a stochastic information fusion rule in the CSIL algorithm. Because of that, CSIL has a significantly lower query complexity, in particular, it makes only a constant number of queries per report.

THEOREM 1. *(Query Complexity) The number of queries to a black box model $\mathcal{M}$ of the CSIL algorithm in one time period t is $O(n)$, where n is the number of reported values.*

PROOF. The CSIL's function *Update* is simple: it uses the set of reports that produced P, say $\{Y_1, ..., Y_k\}$ where $P \longleftarrow \mathcal{M}(\{Y_1, ..., Y_k\})$, adds to it the report $Y_{s,t}$ of sensor s and calculates $P_s^{new} \longleftarrow \mathcal{M}(\{Y_1, ..., Y_k\} \cup \{Y_{s,t}\})$. Therefore, CSIL makes $O(1)$ queries to $\mathcal{M}$ for $i - th$ sensor, thus, for n sensors in one time period t we have $O(n)$ queries. □

An important characteristic of CSIL is that the probabilistic decision making rule allows a possibility of incorporating reports of sensors that are not necessarily considered to be reliable. To make the procedure sound, the probability of fusing a report of a sensor with low reputation is low. For example, a sensor with reputation 0.1 can affect the current pollution map, but only with probability $\frac{0.1}{0.1+1}$. This way, one makes deceiving malicious strategies less effective. In particular, their overall impact cannot be highly negative, meaning that the sum of a sensor's contributions, which can be positive and negative, is bounded from below.

THEOREM 2. *(Limited Damage) The expected total myopic impact $\bar{\Delta}_s = \sum_{t=1}^{T} \bar{\Delta}_{s,t}$ of sensor s is in the CSIL algorithm bounded from below by:*

$$\bar{\Delta}_s > -2 \cdot \rho_0$$

where ρ_0 is the initial reputation of sensor s.

PROOF. The expected myopic impact $\bar{\Delta}_{s,t}$ is equal to $\frac{\rho_{s,t}}{\rho_{s,t}+1} \cdot G_{s,t} = \frac{\rho_{s,t}}{\rho_{s,t}+1} \cdot score_{s,t}$. On the other hand, for reputation $\rho_{s,T+1}$ we have:

$$\ln(\rho_{s,T+1} + 1) = \ln(\rho_{s,T} \cdot (1 + \frac{1}{2} \cdot score_{s,T}) + 1)$$
$$= \ln((\rho_{s,T} + 1) \cdot (1 + \frac{\rho_{s,T}}{\rho_{s,T} + 1} \cdot \frac{1}{2} \cdot score_{s,T}))$$
$$= \ln(\rho_{s,T} + 1) + \ln(1 + \frac{1}{2} \cdot \bar{\Delta}_{s,T}) = ...$$
$$= \ln(\rho_0 + 1) + \sum_{t=1}^{T} \ln(1 + \frac{1}{2} \cdot \bar{\Delta}_{s,t})$$
$$\leq \ln(\rho_0 + 1) + \frac{1}{2} \sum_{t=1}^{T} \bar{\Delta}_{s,t} = \ln(\rho_0 + 1) + \frac{1}{2} \cdot \bar{\Delta}_s$$

where we used the fact that $\ln(1 + x) \leq x$ for $x > -1$. By noting that the updating rule for reputations keeps the reputations positive, i.e., $\rho_{s,t} > 0$, we have that $\ln(\rho_{s,T+1} + 1) > 0$, so $\bar{\Delta}_s$ is lower bounded by:

$$\bar{\Delta}_s > -2 \cdot \ln(\rho_0 + 1) \geq -2 \cdot \rho_0$$

were we again applied $\ln(1 + x) \leq x$ for $x > -1$. □

The consequence of Theorem 2 is that the direct damage of a group of m malicious sensors can be controlled by setting the sensors' initial reputation to a low value. Namely, the impact $\bar{\Delta}_{s,t}$ of sensor s at time period t is measured by its marginal contribution, so the total myopic impact of all malicious sensors over sensing period T is by Theorem 2 at least $-2 \cdot m \cdot \rho_0$ (i.e., the absolute value of the negative impact is at most $2 \cdot m \cdot \rho_0$). By choosing a small value of ρ_0, one can make the (negative) impact of malicious sensors close to 0, regardless of the reporting strategies they use and their reporting time frame. This also implies that when average over a longer sensing period, their negative impact is negligible.

Theorem 2, however, is not sufficient to state that CSIL performs well against malicious strategies. For example, a simple reputation system that discards all the reports completely limits the negative influence of malicious sensors, but in doing so, it discards all the valuable information coming from non-malicious sensors.

The CSIL decision making procedure also induces a certain information loss due to the fact that valuable information might be discarded. This is especially true for the initial sensing periods where all sensors have relatively low reputations, including the ones that are not malicious. For example, if the reputations are set to $\rho_0 = 0.1$, the probability of including a report from an honest and accurate sensor is initially equal to $\frac{0.1}{0.1+1}$. Since only information that comes from sensors with large reputation scores has a good chance of being considered, accurate sensors should build up their reputation quickly, which is indeed the case for the CSIL algorithm because the reputation increase is exponential. Namely, the increase in the reputation is equal to $\frac{1}{2} \cdot \rho_{s,t} \cdot score_{s,t}$, which for a non-malicious sensor with predominantly positive scores implies an exponential reputation growth. Therefore, by using the exponential reputation boosting, CSIL is capable of limiting the negative influence of malicious sensors, while not discarding too many reports of non-malicious sensors.

We measure the expected information loss for partially limiting an accurate sensor s as the difference between the total score of the sensor $\sum_{t=1}^{T} score_{s,t} = \sum_{t=1}^{T} G_{s,t}$ and its impact $\bar{\Delta}_s$. The rationale is that a sensor's scores reflect its contributions — information gains — that the sensor would have made had it not been limited. The following theorem formally shows that if a sensor reports accurate and precise measurements, i.e., its scores are positive in expectation and have small variances, then there is a bound to the amount of sensor s's information discarded by CSIL.

THEOREM 3. *(Bounded Information Loss) Consider a sensor s whose reporting strategy does not depend on its reputation $\rho_{s,t}$ and that has:*

- *Expected scores greater than 0: $\mathbb{E}(score_{s,t}) > 0$*
- *Variance of the scores bounded from above by: $Var(score_{s,t}) < \mathbb{E}(score_{s,t})$.*

Furthermore, let us denote: $g_{s,t} = \ln(1 + \frac{1}{2} \cdot score_{s,t}) \in [g_{min,t}, g_{max,t}]$ *and* $h_{s,t} = \mathbb{E}(g_{s,t})$. *Then the expected information loss* $\sum_{t=1}^{T}(\mathbb{E}(score_{s,t}) - \mathbb{E}(\bar{\Delta}_{s,t}))$ *of the CSIL algorithm for potentially discarding sensor s's reports is bounded from above by:*

$$\sum_{t=1}^{T}(\mathbb{E}(score_{s,t}) - \mathbb{E}(\bar{\Delta}_{s,t})) < z \cdot \left[\frac{e^{-\frac{1}{2} \cdot d}}{1 - e^{-\frac{1}{2} \cdot d}} + \frac{2 \cdot \ln \frac{\rho_0 + 1}{\rho_0}}{h} \right]$$

where $z = \max_{1 \leq t \leq T} \mathbb{E}(score_{s,t}) \leq 1$,
$h = \min_{1 \leq t \leq T}(\frac{1}{t} \sum_{\tau=1}^{t} h_{s,\tau}) \geq \min_{1 \leq t \leq T} h_{s,t} > 0$ *and* $d = \min_{1 \leq t \leq T} \frac{1}{t} \frac{\left(\sum_{\tau=1}^{t} h_{s,\tau}\right)^2}{\sum_{\tau=1}^{t}[g_{max,\tau} - g_{min,\tau}]^2} > \frac{h^2}{2}$.

PROOF. The expected value of the myopic impact is:

$$\mathbb{E}(\bar{\Delta}_{s,t}) = \mathbb{E}(\frac{\rho_{s,t}}{\rho_{s,t} + 1} \cdot score_{s,t})$$

Since scores are stochastically generated (they are independent of reputation $\rho_{s,t}$), we obtain that:

$$\mathbb{E}(\bar{\Delta}_{s,t}) = \mathbb{E}(\frac{\rho_{s,t}}{\rho_{s,t} + 1}) \cdot \mathbb{E}(score_{s,t})$$

Furthermore, Markov's inequality gives us:

$$\mathbb{E}(\frac{\rho_{s,t}}{\rho_{s,t} + 1}) \geq Pr(\rho_{s,t} \geq \rho_0 \cdot a_t) \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1}$$

where we used: $a_t = e^{\frac{1}{2} \sum_{\tau=1}^{t} h_{s,\tau}}$, $h_{s,\tau} = \mathbb{E}(\ln(1 + \frac{1}{2} \cdot score_{s,\tau}))$. Let us also denote: $h = \min_{1 \leq t \leq T} \frac{1}{t} \sum_{\tau=1}^{t} h_{s,\tau}$. Using $\ln(1 + x) \geq x - x^2$ for $x \geq -\frac{1}{2}$, it follows that $h_{s,t} \geq \frac{1}{2} \cdot \mathbb{E}(score_{s,t}) - \frac{1}{4} \cdot \mathbb{E}((score_{s,t})^2)$. By applying the conditions of the theorem, and the fact that $\mathbb{E}((score_{s,t})^2) - (\mathbb{E}(score_{s,t}))^2 = Var(score_{s,t}) < \mathbb{E}(score_{s,t})$ and $0 < (\mathbb{E}(score_{s,t}))^2 \leq \mathbb{E}(score_{s,t}) \leq 1$, we obtain that $h_{s,t} > 0$ (and, hence, $h > 0$). Now, notice that:

$$\begin{aligned} &Pr(\rho_{s,t} \geq \rho_0 \cdot a_t) = Pr(\ln \rho_{s,t} \geq \ln(\rho_0 \cdot a_t)) \\ &= Pr(\ln \rho_{s,t} \geq \ln \rho_0 + \frac{1}{2} \cdot \sum_{\tau=1}^{t} h_{s,\tau}) \\ &= Pr(\ln \rho_{s,t} - \sum_{\tau=1}^{t} h_{s,\tau} - \ln \rho_0 \geq -\frac{1}{2} \cdot \sum_{\tau=1}^{t} h_{s,\tau}) \\ &\geq 1 - Pr(\ln \rho_{s,t} - \sum_{\tau=1}^{t} h_{s,\tau} - \ln \rho_0 \leq -\frac{1}{2} \cdot \sum_{\tau=1}^{t} h_{s,\tau}) \\ &= 1 - p_t \end{aligned}$$

where we denoted the last term $Pr(.)$ by p_t. Since $\ln \rho_{s,t} - \ln \rho_0$ is a sum of t independent random variables $g_{s,\tau} = \ln(1 + \frac{1}{2} \cdot score_{s,\tau})$ (with $1 \leq \tau \leq t$) that are in expectation equal to $h_{s,\tau} = \mathbb{E}(g_{s,\tau})$, using Hoeffding's inequality, we obtain:

$$\begin{aligned} p_t &\leq e^{-\frac{2 \cdot (\sum_{\tau=1}^{t} h_{s,\tau})^2}{4 \cdot \sum_{\tau=1}^{t}[g_{max,\tau} - g_{min,\tau}]^2}} \leq e^{-\frac{2 \cdot \left(\frac{\sum_{\tau=1}^{t} h_{s,\tau}}{t}\right)^2 \cdot t}{4 \cdot \frac{\sum_{\tau=1}^{t}[g_{max,\tau} - g_{min,\tau}]^2}{t}}} \\ &\leq e^{-\frac{1}{2} \cdot d \cdot t} \end{aligned}$$

where we put $d = \min_{1 \leq t \leq T} \frac{\left(\frac{\sum_{\tau=1}^{t} h_{s,\tau}}{t}\right)^2}{\frac{\sum_{\tau=1}^{t}[g_{max,\tau} - g_{min,\tau}]^2}{t}}$, which is greater than $d > \frac{h^2}{2}$ because $\frac{1}{2} \cdot score_{s,\tau} \in [-0.5, 0.5]$ (and, hence, $[g_{max,\tau} - g_{min,\tau}]^2 < 2$). The expected information loss (the difference between the sensor's score and its impact) in round t is bounded by:

$$\begin{aligned} &\mathbb{E}(score_{s,t}) - \mathbb{E}(\bar{\Delta}_{s,t}) = \mathbb{E}(score_{s,t}) \cdot (1 - \mathbb{E}(\frac{\rho_{s,t}}{\rho_{s,t} + 1})) \\ &\leq \mathbb{E}(score_{s,t}) \cdot \left[1 - (1 - e^{-\frac{1}{2} \cdot d \cdot t}) \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1} \right] \\ &= \mathbb{E}(score_{s,t}) \cdot \left[\frac{1}{\rho_0 \cdot a_t + 1} + e^{-\frac{1}{2} \cdot d \cdot t} \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1} \right] \end{aligned}$$

Therefore, over time period T, the information loss in expectation upper bounded by:

$$z \cdot \left[\sum_{t=1}^{T} \frac{1}{\rho_0 \cdot a_t + 1} + \sum_{t=1}^{T} e^{-\frac{1}{2} \cdot d \cdot t} \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1} \right]$$

where $z = \max_{1 \leq t \leq T} \mathbb{E}(score_{s,t})$. We examine bounds for each of the terms in the bracket. We have:

$$\begin{aligned} &\sum_{t=1}^{T} e^{-\frac{1}{2} \cdot d \cdot t} \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1} \leq \sum_{t=1}^{T} e^{-\frac{1}{2} \cdot d \cdot t} \\ &= e^{-\frac{1}{2} \cdot d} \cdot \sum_{t=0}^{T-1} e^{-\frac{1}{2} \cdot d \cdot t} < e^{-\frac{1}{2} \cdot d} \cdot \sum_{t=0}^{\infty} e^{-\frac{1}{2} \cdot d \cdot t} = \frac{e^{-\frac{1}{2} \cdot d}}{1 - e^{-\frac{1}{2} \cdot d}} \end{aligned}$$

where we applied $\sum_{t=0}^{\infty} x^t = \frac{1}{1-x}$ for $x \in (0, 1)$. Furthermore, using the fact that:

$$a_t = e^{\frac{1}{2} \sum_{\tau=1}^{t} h_{s,\tau}} \geq e^{\frac{1}{2} \cdot t \cdot h}$$

we obtain:

$$\begin{aligned} &\sum_{t=1}^{T} \frac{1}{\rho_0 \cdot a_t + 1} \leq \sum_{t=1}^{T} \frac{1}{\rho_0 \cdot e^{\frac{1}{2} \cdot t \cdot h} + 1} \\ &\leq \int_{t=0}^{T} \frac{1}{\rho_0 \cdot e^{\frac{1}{2} \cdot t \cdot h} + 1} dt < \int_{t=0}^{\infty} \frac{1}{\rho_0 \cdot e^{\frac{1}{2} \cdot t \cdot h} + 1} dt \\ &= \frac{2}{h} \cdot \ln(\frac{\rho_0 + 1}{\rho_0}) \end{aligned}$$

Which completes the proof. □

The intuition behind this results is fairly simple. If a sensor has mostly positive scores, i.e., the expected scores are positive and their variances are low, it will boost up its reputation rather quickly to the values where its reports are practically no longer limited. Notice that the bound on the total information loss does not (directly) depend on time (i.e., does not monotonically increase with time), which means that when averaged over a long sensing period T (typical for crowdsensing), the information loss becomes negligible. Furthermore, the bound multiplicatively depends on parameter z that represents a sensor's expected score: the better the sensor is, the more quality information the center looses when it discards the sensor's reports. The second multiplicand in the bound describes how quickly a sensor can boost up its reputation, which depends on how *informative* the sensor is: the more useful the sensor's reports are, the higher its score is, and, thus, the greater its reputation increase is. This is captured by parameters h and d, which are related to the performance of a sensor through random variable $g_{s,t} = \ln(1 + \frac{1}{2} \cdot score_{s,t})$. In the next section, we give an example scenario for which we estimate the values of z, h and d. Notice that by Theorem 3, we can set $z = 1$ and $d = \frac{h^2}{2}$ in order to obtain a looser upper bound that does not require estimates of z and d.

Theorem 2 and Theorem 3 provide guarantees on the performance of the CSIL algorithm that depend on the initial reputation. The bounds of the theorems indicate that the value of the initial reputation ρ_0 should be such that it limits the negative impact of malicious sensors, while not discarding too much information from non-malicious sensors. Since for a longer sensing period, accurate sensors have enough time to build up their reputations, the initial reputation ρ_0 can be set to a relatively small value so that the CSIL algorithm is more robust against malicious reporting strategies.

Finally, we analyze the incentive component of the CSIL algorithm. The important property of sensors' scores, which can be used to define monetary payments, is that they incentivize non-malicious sensors to provide reports that maximize the information gain of the center. Notice that the most *useful* information is not necessarily the true measurement. This is due to the presence of malicious sensors, as well as the possible imperfections of pollution model $\mathcal{M}$. In other words, a strategic behaviour is often desirable.

THEOREM 4. *(Informed Reporting) If a sensor s maximizes its expected score* $\mathbb{E}(score_{s,t})$*, then it also maximizes its expected impact* $\mathbb{E}(\bar{\Delta}_{s,t})$.

PROOF. The myopic impact of sensor s, $\bar{\Delta}_{s,t}$, is proportional to its score $\bar{\Delta}_{s,t} = \frac{\rho_{s,t}}{\rho_{s,t}+1} \cdot G_{s,t} = \frac{\rho_{s,t}}{\rho_{s,t}+1} \cdot score_{s,t}$. Hence, a sensor s that aims to maximize its expected score, is also incentivized to submit a report that maximizes its expected impact. □

5. EXPERIMENTAL EVALUATION

Considering that, in a real dataset, one cannot identify upfront the strategies adopted by different sensors, we simulate different malicious strategies to experimentally validate our approach. Our pollution sensing scenario is based on four weeks of hourly measurements of NO_2 concentrations from an area in Strasbourg (France) covering 116 locations, with each week coming from a different season. The data is the output from the physical model ADMS Urban V2.3 [7] collected by ASPA [2], denoting estimations of pollution concentrations calculated from the emission inventory and actual measurements. In total, the dataset contains approximately one month of hourly measurements - the larger sensing periods can be simulated by looping over the dataset several times, which we do 12 times to obtain the sensing time of $T = 12 \cdot 4 \cdot 7 \cdot 24$ hours. Our main reputation system is CSIL with the initial reputation set to $\rho_0 = 0.1$.

5.1 Pollution Model

We use a probabilistic air pollution model that is based on Gaussian process regression, as described in [18]. For any point of interest (in our case 116 locations), the pre-trained Gaussian Process (GP) model produces a probability distribution function over the possible levels of pollution from the reports of sensors placed at different locations. This posterior distribution is a normal distribution $\mathcal{N}(\mu, \sigma)$, with parameters μ and σ derived from the GP model. We are interested in predicting the value of pollution level that a trusted sensor measures at its location, so we denote the corresponding prediction by $p(X_{trust})$.

5.2 Scoring Function

The standard way of measuring the quality of a probability distribution function is by *strictly proper scoring rules* [11, 12], that take as argument both the predicted probability distribution and the outcome of the predicted event. In our case, the event that we aim to predict is measurement X_{trust} of a trusted sensor, and the prediction we want to evaluate is the posterior density function $p(X_{trust})$, which represents the output of the model at the location of the trusted sensor. We focus on a quadratic scoring rule, that, for events whose outcomes take real values, has the form:

$$S(p, X) = p(X) - \frac{1}{2}\int_{-\infty}^{\infty} p(y)^2 dy \qquad (2)$$

Model $\mathcal{M}$ outputs a normal distribution $\mathcal{N}(\mu, \sigma)$ for a point of interest (x, y). Therefore, we apply scoring rule (2) on probability density function p of the form $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ to obtain:

$$S(p, X_{trust}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X_{trust}-\mu)^2}{2\sigma^2}} - \frac{1}{4\sigma\sqrt{\pi}}$$

The score takes values in $[-\frac{1}{4\sigma\sqrt{\pi}}, \frac{1}{\sigma\sqrt{\pi}}(\frac{1}{\sqrt{2}} - \frac{1}{4})]$, and can be further scaled so that $score_{s,t} \in [-1, 1]$. In our case, no specific scaling was needed.

5.3 Sensors

We consider 40 mobile crowd-sensors and 1 trusted sensor that are at each time period placed at one of 116 available locations. The 40 crowd-sensors are either honest (25% of them) or are malicious sensors (75% of them) that report according to one of the following four strategies. In the *Vary* strategy, sensors build up their reputations by reporting honestly for the first 1000 iterations, and from then on, they report only a low level of pollution. In the *Deceive* strategy, sensors report honestly when their reputation is below 0.5; otherwise, they report a low level of pollution. *Vary and Deceive* is a mixed strategy where malicious sensors first build up their reputation by reporting honestly for 1000 iterations, and from then on, they use the *Deceive* strategy. *Cover* is a strategy that mimics a situation where malicious sensors try to boost up their reputation when it is not important for them to misreport, and then, on specific events, they report wrong values. In our case, malicious sensors boost up their reputation for 1000 iterations. Then, they report honestly whenever the pollution is below 35 ppb of NO_2 or their reputation is lower than 0.5; otherwise, they report a low level of pollution. The low level of pollution in the above strategies is defined as 10 ppb of NO_2 plus a Gaussian noise with 0 mean and standard deviation equal to 5.

5.4 Theoretical Bound

By Theorem 2, it follows that $0.75 \cdot 40 = 30$ malicious sensors can cause an immediate damage of at most $2 \cdot 30 \cdot 0.1 = 6$ score units (units used in (2)). To calculate the bound from Theorem 3, one needs to estimate parameters z, h and d. We can approximate these values by investigating averages of $score_{s,t}$, $\log(1+\frac{1}{2} \cdot score_{s,t})$ and $\max_s \log(1+\frac{1}{2} \cdot score_{s,t}) - \min_s \log(1 + \frac{1}{2} \cdot score_{s,t})$ over time t. Assuming that the scores of honest sensors are similar in most of the sensing periods, these averages lead to the estimates:[2] $z \approx 0.002$, $h \approx 0.001$ and $d \approx 0.005$, from which we can estimate the

[2] If $k << T$ sensing periods have significantly different values from the average values, to achieve a higher precision, one can exclude these k periods when estimating the upper

the upper bound from Theorem 3: 10.39. By multiplying the estimate with the number of honest sensors (i.e., 10), we conclude that the total information loss should be no more than 103.9 score units. Notice that the bounds from Theorem 2 and Theorem 3 have different meanings: the bound from Theorem 2 describes how much a malicious sensor could intentionally shift the result, while the bound from Theorem 3 describes an implicit damage whose nature is not controlled by a malicious sensor. Nevertheless, it follows from the bounds that the quality degradation should not be more than 109.9 score units in total. This can be averaged over time, so that at each time step t, we have an average degradation of at most $\frac{109.9}{t}$ score units. The average goes to 0 as time increases, implying a no-regret property in terms of sensors' myopic impact.

5.5 Simulations

5.5.1 Baseline: Beta Reputation System

In the Beta reputation system, we quantify the behaviour of a sensor using two parameters, α and β, which represent the parameters of beta distribution $B(\alpha, \beta)$. In the setting we analyze, the parameters can be updated as follows (e.g., see [8]). If the marginal information gain $G_{s,t} = score_{s,t}$ of updating the current pollution map with a sensor s's report is positive, parameter $\alpha_{s,t}$ is updated to $\alpha_{s,t+1} = \alpha_{s,t} + G_{s,t}$. Otherwise, parameter β is updated to $\beta_{s,t+1} = \beta_{s,t} + G_{s,t}$. The reputation of sensor s is at time t calculated as the mean of beta distribution $B(\alpha_{s,t}, \beta_{s,t})$, i.e., $\rho_{s,t} = \frac{\alpha_{s,t}}{\alpha_{s,t}+\beta_{s,t}}$. In other words, the reputation of sensor s characterizes the fraction of the positive impact that the sensor had on the system. The decision on whether to include the report of sensor s is based on its reputation and determined using the thresholding principle. We set the initial values of α and β parameters to 0.01 and 0.1, respectively, with threshold $\Theta = 0.5$.

5.5.2 Evaluation Metric

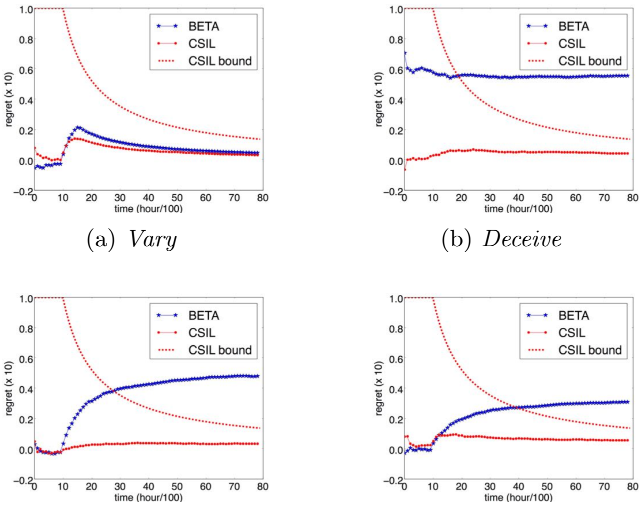
We define a measure of an average regret that evaluates the quality of the aggregates produced by the center with respect to the aggregates obtained by fusing the reports of honest sensors. More precisely:

$$AvgRegret_t = \frac{Score_{honest,t} - Score_{center,t}}{t}$$

where $Score_{honest,t}$ is the total score (until time period t) of the aggregates obtained from the reports of honest sensors, and $Score_{center,t}$ is the total score of the center (with a particular reputation system) until time period t. Both scores are calculated using the quadratic scoring rule, as described by the previous subsections, applied on the pollution map published prior to the report of a trusted sensor. Therefore, the regret is measured in the same score units as the theoretical bound computed in Section 5.4.

5.5.3 Results

Figures (2a, 2b, 2c, 2d) show the performance of the CSIL algorithm and the Beta reputation system in terms of the average regret for four different misreporting strategies. Along with those results, we put the theoretical estimate of the upper bound on the regret of CSIL algorithm ($\frac{109.9}{t}$), which is truncated to 0.1 for large values. The Beta reputation system is able to limit the negative influence of malicious sensors that use the *Vary* strategy. However, in the *Vary* strategy, malicious sensors misreport in a simple and consistent way. For the other three misreporting strategies, the Beta reputation system experiences an average regret that is clearly away from 0, and in two of the cases, the regret is increasing, which means that the total negative impact of malicious sensors is not bounded. The CSIL algorithm is much better in dealing with malicious sensors: its average regret over a longer sensing period is for all the malicious strategies close to 0, as expected by the theoretical results. Finally, the strategy independent upper bound on the CSIL's regret is often below the regret of the Beta reputation system.



(a) *Vary*

(b) *Deceive*

(c) *Vary and Deceive*

(d) *Cover*

Figure 2: Average regrets (times 10) for different strategies

6. CONCLUSION

We discus a problem of having malicious sensors in community sensing with online information fusion. Due to the abundance of crowdsourced data, one can partially discard useful information to limit the overall negative influence of malicious participants. We have designed a novel reputation system, called CSIL, that has a manageable complexity and puts an upper bound on the total negative impact that malicious sensors can have on the fused result, regardless of their reporting strategy. This is in contrast to the standard reputation systems proposed for sensing which do not provide any theoretical guarantees and for which the total negative impact of malicious sensors can increase with time. We have empirically confirmed that the theoretical results hold in a realistic air pollution sensing scenario, and have shown that in an average-case simulation, CSIL outperforms a state of the art reputation system for sensing, whose performance is often worse than the worst case performance of CSIL.

Acknowledgments

The work reported in this paper was supported by Nano-Tera.ch as part of the OpenSense2 project. We thank Jason Jingshi Li for providing a testbed and the anonymous reviewers for useful comments and feedback.

bound from Theorem 3 and simply add to the calculated bound $k \cdot \max_{\tau \in kPeriods} \mathbb{E}(score_{s,\tau})$.

REFERENCES

[1] K. Aberer, S. Sathe, D. Chakraborty, A. Martinoli, G. Barrenetxea, B. Faltings, and L. Thiele. Opensense: Open community driven sensing of environment. In *ACM SIGSPATIAL International Workshop on GeoStreaming*, 2010.

[2] ASPA. l'association pour la surveillance et l'etude de la pollution atmosphérique en alsace. `www.atmo-alsace.net`, 2013.

[3] S. Buchegger and J.-Y. L. Boudec. A robust reputation system for mobile ad-hoc networks. Technical report, Proceedings of P2PEcon, 2003.

[4] S. Buchegger and J.-Y. Le Boudec. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2002.

[5] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications*, 2006.

[6] H. Chen. Task-based trust management for wireless sensor networks. *International Journal of Security and Its Applications*, 2009.

[7] R. N. Colvile, N. K. Woodfield, D. J. Carruthers, B. E. A. Fisher, A. Rickard, S. Neville, and A. Hughes. Uncertainty in dispersion modelling and urban air quality mapping. *Environmental Science and Policy*, 5(3):207–220, 2002.

[8] B. E. Commerce, A. Josang, and R. Ismail. The beta reputation system. In *Proceedings of the Electronic Commerce Conference*, 2002.

[9] S. Ganeriwal and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, 2004.

[10] 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*, 2011.

[11] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 2007.

[12] R. D. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1):1–15, 2007.

[13] K. L. Huang, S. S. Kanhere, and W. Hu. On the need for a reputation system in mobile phone based sensing. *Ad Hoc Networks*, 12:130–149, Jan. 2014.

[14] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the Sixth Joint Working Conference on Communications and Multimedia Security*, 2002.

[15] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

[16] A. Papakonstantinou, A. Rogers, E. H. Gerding, and N. R. Jennings. Mechanism design for the truthful elicitation of costly probabilistic estimates in distributed information systems. *Artificial Intelligence*, 175(2):648–672, 2011.

[17] G. Radanovic and B. Faltings. Incentive schemes for participatory sensing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015.

[18] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[19] S. Reece, S. J. Roberts, C. Claxton, and D. Nicholson. Multi-sensor fault recovery in the presence of known and unknown fault types. In *12th International Conference on Information Fusion*, 2009.

[20] P. Resnick and R. Sami. The influence limiter: Provably manipulation-resistant recommender system. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, 2007.

[21] S. Saroiu and A. Wolman. I am a sensor, and I approve this message. In *Proceedings of the Eleventh Workshop on Mobile Computing Systems and Applications*, 2010.

[22] M. Venanzi, A. Rogers, and N. R. Jennings. Trust-based fusion of untrustworthy information in crowdsourcing applications. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, 2013.

[23] W. Wang, H. Li, Y. Sun, and Z. Han. Catchit: Detect malicious nodes in collaborative spectrum sensing. In *Proceedings of the 28th IEEE Conference on Global Telecommunications*, 2009.

[24] K. Yadav and A. Srinivasan. iTrust: An integrated trust framework for wireless sensor networks. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, 2010.