A Reputation System Resilient Against Colluding and Malicious Adversaries in Mobile Participatory Sensing Applications

Hayam Mousa*†, Sonia Benmokhtar*, Omar Hasan*, Lionel Brunie*, Osama Younes†, Mohiy Hadhoud†
† Faculty of Computers and Information, Menoufia University, Egypt
*LIRIS, INSA Lyon, France

***Abstract*—Participatory sensing is an emerging paradigm in which citizens voluntarily use their mobile phones to capture and share sensed data from their surrounding environment in order to monitor and analyze some phenomena. Participating users can disrupt the system by contributing corrupted, fabricated, or erroneous data. Different reputation systems have been proposed to monitor participants' behavior and to estimate their honesty. There are some attacks that were not considered by the existing reputation systems in this context including corruption, collusion, and on-off attack. In this paper, we propose a more robust and efficient reputation system designed for these applications. Our reputation system incorporates a mechanism to defend against those attacks. Experimental results indicate that our system can accurately estimate the quality of contributions even if a collusion is committed. It can tolerate up to 60% of colluding adversaries involved in the sensing campaign. This enables our system to aggregate the data more accurately compared with the state-of-the art. Moreover, the system can detect adversaries even if they launch on-off attack and strategically contribute some good data with high probability (e.g. 0.8).**

I. Introduction

Everyday, millions of people move around carrying a variety of handheld devices equipped with sensing, computing, and networking capabilities (e.g., smartphones, tablets, music players, GPS watches, in-vehicle sensors, etc.). The advancement and widespread use of such devices have contributed toward the emergence of a new kind of application called *participatory sensing* [1]. These applications exploit both the mobility of participants and the sensing capabilities of their devices to construct opportunistic mobile sensor networks [2]. Participants capture sensed data from their surrounding environment using a variety of sensors (e.g., GPS, camera, microphone, accelerometer, gyroscope, digital compass, etc.) embedded in their devices. Then, they share their collected observations with a backend server, which processes the received data to monitor, map, or analyze some incidents or phenomena of common interest.

Participatory sensing systems can be applied to serve many of our daily life needs, including monitoring health [4], traffic[5], noise [3], weather, commerce, as well as many other applications [6]. In these applications, no restrictions are usually imposed about participants' experience, concern, trustworthiness, and interest. In addition, they are not usually paid for their participation in sensing campaign. Thus, they usually do not have strong motivations to comply with the tasks' requirements. That is, they are not concerned about some parameters which may improve the quality of their contributions (e.g. time, location and/or the position of the device during the sensing process). As a consequence, participatory sensing applications are vulnerable to *erroneous* and *malicious* participants. We define erroneous and malicious participants as those who mislead and disrupt the system measurements by reporting false, corrupted or fabricated contributions either intentionally or non-intentionally. Non-intentional (i.e. erroneous) corruption may originate from a malfunctioning sensor while intended (i.e. malicious) corruption is deliberately committed to alter the system measurements in a specific location. For instance, an adversary can put his device in a non-appropriate position. Alternatively, the participant can modify a contribution before sharing it. Malicious participants may further launch various types of attacks such as Sybil, collusion, on-off attack, etc. Some of these attacks are discussed in Section III-B. Consequently, the need arises for approaches that try to detect erroneous participants and deter or mitigate malicious ones in order to evaluate the veracity and accuracy of participants' contributions and therefore to build robust and reliable application systems.

Different reputation systems have been proposed for participatory sensing applications. We have studied, classified, compared those systems in [7]. It is evident that, those systems are in their infancy, and have several limitations. One of these limitations is the estimation of the quality of contributions in the existence of collusion attack. A few researchers have addressed the estimation of the quality of contributions for example, Huang et al. [8], Wang et al. [9], and Manzoor et.al. [10]. However, such systems are not resistant against malicious colluding adversaries. These systems exploit some consensus and outlier detection algorithms to evaluate the consistency of each contribution (e.g. [11], [12]). Subsequently, such systems are biased if a good participant is surrounded by a number of colluding adversaries. That is, it ends up getting a good participant defined as malicious and vice versa.

In this paper, we propose a novel and efficient reputation system to estimate the trustworthiness of participants' contributions. The system also adopts a methodology to detect adversaries even if there is a large number of colluding ones. It also incorporates other novel parameters, including a proximity factor and users' feedback, to assign a trust score

to each contribution. These trust scores give the system the ability to aggregate more accurate data which may reflect the ground truth more precisely compared with the state-of-the-art. The parameters exploited by the system are collected by most existing participatory sensing applications (e.g. data, location, etc.). Thus, our system is applicable to most of typical participatory sensing applications (e.g. noise, pollution, weather, traffic, etc.).

The rest of this paper is organized as follows: Section II states the previous work and its limitations. We then give an overview about participatory sensing and it's threat model in Section III. We then describe and discuss in details our proposal in Section IV. The experimental results of our reputation system are discussed in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Different reputation systems have been proposed in literature for different participatory sensing applications. A reputation system for noise monitoring application system is presented by Huang et al. in [8]. This system adopts a robust average algorithm through a watchdog module to measure the quality of the recorded noise samples provided by each participant. In [10] and [9], Manzoor et al. and Wang et al measure the quality of participant contribution through a Gaussian membership function and a similarity factor that measures the consistency of each contribution compared with the others respectively. All these systems adopt some outlier detection or consensus algorithms to measure the deviation of each contribution from the common consensus (e.g. [12], [11]). Thus, the results of these systems disrupt if a large number of malicious or colluding adversaries is involved in the sensing campaign.

Other reputation systems have been proposed earlier for social participatory sensing applications in [14], [15], [16]. These systems mainly depend on some social parameters for estimating the trustworthiness of participants. These parameters include friendship duration, interaction time gap, familiarity, etc. However, these parameters are not usually available in all participatory sensing applications. Thus, these systems are not applicable with the wide range of participatory sensing application.

Complementary to reputation based trust systems, researchers suggest to equip smartphone' sensors with an embedded Trusted Platform Module (*TPM*) [17], [18], [19]. Such a module ensures the authenticity of participants' contributions. A major limitation of TPM-based solutions is that they only consider data authenticity regardless of the participant's sincerity and honesty. TPM can not detect contributions from malicious participants who deliberately initiate sensing actions that cause distortion of their contributions (e.g putting the device in non-appropriate position). In addition to erroneous contributions that originate from a malfunctioning sensor.

For more details about reputation systems in participatory sensing, its classification, their merits, limitations, and different research directions in this domain, please refer to the survey presented by Mousa et al. in [7].

III. System Model

In this section, we establish a framework that allows us to analyze the reputation system presented in Section IV.

A. Basic Definitions

Trust and reputation have been defined earlier in the context of participatory sensing by Wang et al. in [9] as follows:

a) Definition 1: **Trust of a contribution:** The trust of a contribution C, denoted as $Trust(C)$, is the probability of C being correct, as perceived by the server.

b) Definition 2: **Reputation of a Participant:** The reputation of a participant p_i, denoted as $\hat{R}_{p_i}$, is the synthesized probability that the past contributions sent by p_i are correct, as perceived by the server.

c) Definition 3: **Participant's Behavior:** The participant p_i is identified as a good participant if he is assigned a reputation score $\hat{R}_{p_i}$ that exceeds a predefined minimum threshold τ. Otherwise, he is identified as a malicious or adversary. The following equation describes this concept.

$$behavior = \begin{cases} \text{Good} & \text{if } \hat{R}_{p_i} >= \tau \\ \text{Malicious} & \text{if } \hat{R}_{p_i} < \tau \end{cases} \quad (1)$$

B. Threat Model

Below, we define the attacks that are mainly considered along this work (e.g. Corruption, collusion, on-off attacks). We treat them here in the context of participatory sensing applications for the first time.

Corruption attack leads to an erroneous contribution. It may arise as a result of a malfunctioning sensor or adversary can deliberately contribute corrupted or forged data. Additionally, a local processing module can be used for modifying the sensed data before sharing it. He can also initiate sensing actions which may corrupt the sensed data by putting the device in non-appropriate positions. For example, in air quality mapping system, the adversary may put his device beside a cigarette flame. The system should have strong capabilities to identify correct contributions in order to identify and exclude corrupted ones.

In *collusion attack*, malicious colluding participants coordinate their behavior in order to provide unified false contributions, and/or false feedback. Multiple malicious participants acting together can cause more damage than each one acting independently. If the majority of participants collude they can mislead the system measurements and decisions. In order to attain robustness against such attack, systems should not rely on consensus algorithms to define good and bad contributions. Otherwise, the system measurements and decisions are biased.

In *On-Off attack*, an adversary alternates between normal and abnormal behaviors. Specifically, he provides false data randomly and irregularly with a probability. Thus, he can keep his reputation above the required threshold. This makes it difficult to be detected.

IV. The Dynamic Trusted Set based Reputation System: DTSRS

A. Overview

In the context of participatory sensing, evaluating the quality of participants' contributions is a crucial task. By the term *quality*, we mean *how much a contribution is close to the ground truth in the sensing area*. In the state of the art, authors measure the consistency of a contribution with the other contributions provided by other participants. However, this measure is usually disrupted especially when there is a large number of adversaries involved in the sensing campaign. In different contexts, the systems rely on a trusted third party that can provide her with the ground truth. However, in the context of participatory sensing, this third party is not available. Thus, we try to propose a more efficient and robust mechanism for evaluating contribution quality depending on a *Trusted Participant* set (TP). We define this set such that it involves the participants with the higher reputation scores $\hat{R}_{p_i}$. Those participants are usually more trusted and have higher probabilities to submit good data. Relying on this set of reliable contributions to evaluate the rest of contributions gives our system better idea about which contribution is correct and which is false. Therefore, our system has better capabilities to detect adversaries who contribute bad data. The trusted set is *dynamic* such that it is updated after each campaign in order to base on the most recent reputation information. Thus, we refer to the proposed system as a Dynamic Trusted Set based Reputation System (DTSRS). In the following, we describe how this trusted set is constructed and updated. Then, we illustrate how this set is exploited to assess trust through the sensing campaign.

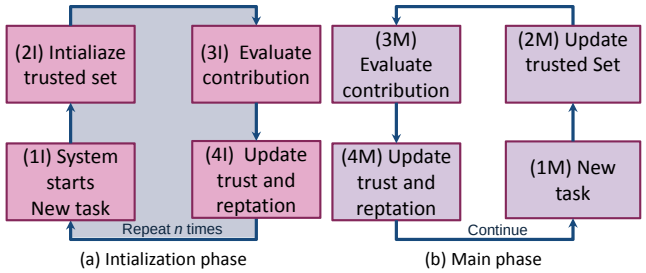


Fig. 1: Trusted set management

B. Trusted set Management

The trusted set management process is depicted in Figure 1. This process is composed mainly of two phases as illustrated in the sub-figures a and b. First, the *initialization phase* which is carried out for n times when the application is started and there is no available reputation data concerning each participant (step 1I). Therefore, we adopt a methodology to use the most consistent contributions in order to evaluate the others (step 2I). Through this phase, involved participants are assigned reputation scores that identify them either as good or malicious ones (step 3I, 4I). The details of this phase is illustrated in the following subsection. After a number of iterations (e.g. n), the system moves to the *main phase* (step 1M) and update the trusted set according to the reputation scores of the participants (step 2M). Then, this set is used to evaluate the current contributions and subsequently calculate new trust and reputation scores (step 3M, 4M). Below, we describe in more detail both the initialization and the update of the trusted set.

1) The Initialization Phase: This phase starts by determining the size of the trusted participants TP as an input. First, we measure the similarity between each contributions and the rest (C_{p_i}, C_{p_l}) using some similarity measures as the ones introduced in the field of data mining in [13]. The output of this measure ranges from -1 for completely conflicting contributions to +1 for contributions which are exactly the same. We then calculate the average consistency of each contribution. Hereafter, the available contributions are arranged in descending order according to this average. Finally, the first contributions that have higher average of consistency are selected. These steps are repeated for n tasks.

2) Update the Trusted Set: In this phase, the trusted set is updated according to the current reputation score of the participants. The current reputation scores of participants are reported in some way depending on the system methodology. If the sensing campaign is non-anonymous (e.g. the identity of participants is known), the reputation scores of those participants are retrieved either from a common database or reputation queries are sent to a reputation server. Otherwise, reputation scores are demonstrated using some anonymous demonstration (e.g. anonymous reputation certificate), if the sensing campaign is anonymous. The contributions of the current task are then arranged according to the reputation scores of their providers. Finally, the first TP contributions are selected.

C. Trust Assessment and Reputation Update

In this subsection, we provide an overview about the methodology exploited for trust assessment and reputation in our DTSRS system. Figure 2 depicts the main trust parameters exploited in our system and identified as blue boxes in the figure. While white boxes are the information sources and the brown ones refer to the modules where these parameters are aggregated for assessing trust and reputation of each contribution. We target to assess the trust of contributions in away that the assigned trust scores reflect the consistency of the contributions with ground truth rather than the consistency of them with each other as considered in the state-of-the art. This provides our system with much more resistance against collusion. A brief definition of the function of each module is described as follows:

First, a *contribution evaluation* module evaluates the quality of a participant's current contribution. It measures the deviation of each contribution from the mean of the set of contributions which are provided by the most trusted participants. First, contributions that belong to the same task $Task_t$ are grouped together (step 1 and 2). The trusted set is defined according to the methodology described in the previous subsections. Then, the deviation of each contribution from the mean of those

trusted contributions is calculated and assigned a score θ_{p_i} (Step 3).

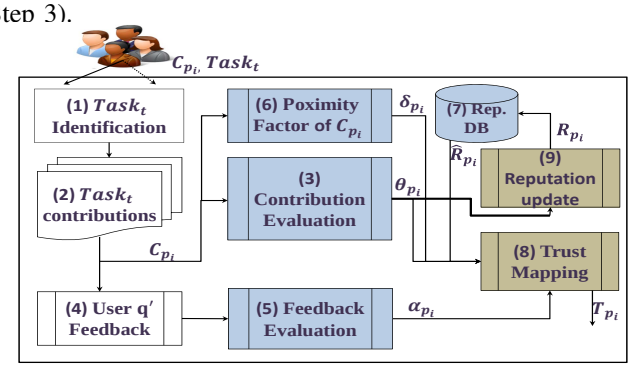


Fig. 2: The Framework of our DTSRS system

Second, sensed data are published through a public server. Thus, end users query these data. Those users are themselves a subset of the participants who are involved in the sensing area. They are sometimes permitted to provide a feedback for the received contributions. An accurate feedback usually reflects how much the rated contribution agrees with ground truth as perceived by the user. User q report a feedback about the contribution of a participant p_i noted as $F_q(p_i)$ (step 4). A user may report a feedback that does not reflect his genuine opinion about the target contribution. This is considered as unfair rating attack [22]. Thus, the feedback is evaluated to mitigate the effect of such attack and aggregated to assign a feedback score α_{p_i} to the target participant (step 5).

Third, participatory sensing are usually interested in a specific sensing area. Additionally, sensed data are affected by the distance from the sensing area. For instance, a noise sample recorded by a participant is significantly affected by a near by sound source such as train station, crowd, etc. This noise is considered to attenuate by going away from its' source [23]. Subsequently, the closer a participant to the considered sensing area, the more accurate his contribution. Here, we propose to define a proximity factor δ_{p_i} that measures the vicinity of a participant to the center of the sensing area. The contribution is subsequently assigned a score which reflects its possible decay according to the nature of the application (step 6).

Finally, the reputation score $\hat{R}_{p_i}$, which is previously assigned to the participant according to his previous contributions, is also considered (step 7). This score describes the historical behavior of the participant. Thus, it gives an indication of the participant's expected behavior during the subsequent tasks. Incorporating the historical reputation score of a participant enables to trace the behavior of this participant and help to detect the ones who launch on-off attack.

In the trust mapping module, the collected measures including θ_{p_i}, α_{p_i},δ_{p_i}, $\hat{R}_{p_i}$ concerning the contribution of the target participant p_i are integrated to assign a trust score to his current contribution $Trust$ (step 8). The reputation score of the participant p_i is then updated to R_{p_i} (step 9). In the following subsection, we discuss the details of these modules.

1) Contribution Evaluation: Consider np is the number of participants who joined the sensing campaign and p_i is one of them who submits a contribution C_{p_i} such that

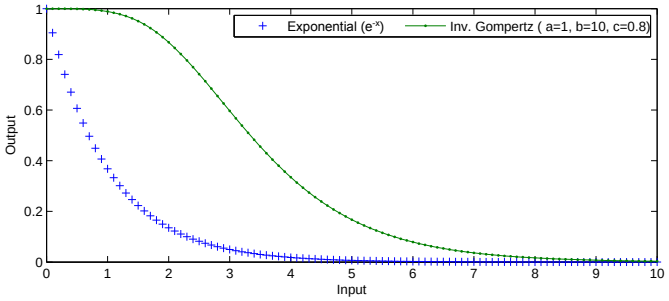


Fig. 3: The output of both exponential and Inv. Gompertz

$i \in \{1, 2, 3, ..., np\}$. First, the trusted set of participants TP is defined according to the methodology illustrated above. The mean of the contributions provided by this set ($C_j \in \{C_1, C_2, C_3, ..., C_{TP}\}$) is calculated and is noted as $\mu(C)$ (Equation 2). The higher the similarity of a contribution C_{p_i} with this mean, the more reliable it is considered. Thus, the deviation of each contribution, from this mean, is calculated and is noted as *Contribution Deviation* d_{p_i} as depicted in Equation 3.

d_{p_i} ($\forall i \in 1, 2, 3, ..., np$) is then normalized to the range [0,1]. The normalized deviation is noted as $d^{no}_{p_i}$ as shown in Equation 4. $d^{no}_{p_i}$ of 0 means that the contribution of p_i is exactly the same as the mean of the trusted contributions. Whereas, 1 means it completely contradicts with this mean. The participant's contribution C_{p_i} is assigned a score θ_{p_i} which reflects its quality by feeding the normalized deviation as an input to an exponential distribution. The output of this distribution is depicted in Figure 3. Using this distribution, participants are assigned scores which commensurate with their normalized deviation. For example, a contribution with a normalized deviation of 0 is assigned the maximum score 1. The output of the distribution is defined according to Equation 5 and depicted in Figure 3.

$$\mu(C) = \frac{\sum_{j=1}^{j=TP} C_{p_j}}{TP} \quad (2)$$

$$d_{p_i} = abs(C_{p_i} - \mu(C)), i \in \{1, 2, 3, ..., np\} \quad (3)$$

$$d^{no}_{p_i} = \frac{d_{p_i} - min\{d_{p_i}\}_{i=1}^{np}}{max\{d_{p_i}\}_{i=1}^{np} - min\{d_{p_i}\}_{i=1}^{np}} \quad (4)$$

$$\theta_{p_i} = \exp^{-d^{no}_{p_i}} \quad (5)$$

where $abs(.)$ is the absolute function, np is the total number of contributions provided by np participants for the considered task. We also normalized the input (i.e. the deviation) to the range [0,1]. Thus, we have the quality score θ_{p_i} output in the range $[1, \exp^{-1}] \Rightarrow [1, 0.37]$.

We run the experiments different times to determine the most suitable range for normalization. Using the exponential distribution, we found that the normalization of the input to the range [0,1] and getting an output in the range [1,0.37] allows for more accurate data aggregation. That is calculating the weighted sum of the available contributions according to these scores are more close to the ground truth.

2) Feedback Processing: This module targets to verify the user's feedback $F_q(p_i)$ which lies in the range [0,1]. For this, if the reputation score of the rater q exceeds the reputation score of the target participant p_i (i.e. $\hat{R}_q > \hat{R}_{p_i}$) the reputation $\hat{R}_q$ of the rater q will be used as a weight for his provided rating, as depicted in Equation 6. Otherwise, the rater's feedback is excluded. Consequently, the rate provided by a poor reputation user is less considered, and vice versa. Different feedback scores which assigned for the same contribution are aggregated. An average feedback score α_{p_i} is then calculated according to the Equation 7, where F is total number of feedback providers. The aggregated feedback score lies also in the range [0,1].

$$F_{q\,Evl}(p_i) = F_q(p_i) \times R_q \quad \text{if } \hat{R}_q > \hat{R}_{p_i} \quad \forall q \in 1, 2, ..., F \tag{6}$$

$$\alpha_{p_i} = \frac{\sum_{q=1}^{q=FP} F_{q\,Evl}(p_i)}{FP} \tag{7}$$

3) A Proximity Factor: The proximity factor, as we mentioned earlier, measures the vicinity of a participant to the sensing area. As a first step towards the calculation of this measure, the distance between the center of the *Target Sensing Area* (TSA) and the *Sensing Location* ($SL(p_i)$) where the contribution is captured by the participant p_i is calculated. Here, we adopt the Euclidean distance as a simple and common distance measure (Equation 8).

$$\beta_{p_i} = \sqrt{(TSA_x - SL_x(p_i))^2 + (TSA_y - SL_y(p_i))^2} \tag{8}$$

where (TSA_x, TSA_y) and $(SL_x(p_i), SL_y(p_i))$ are the coordinates of the TSA and $SL(p_i)$ respectively.

The proximity score depends on the considered phenomenon and its dispersion rate. Some phenomena are location *sensitive* such as noise, pollution, traffic, etc. Other phenomena are more *stable* in the sensing area such as temperature and precipitation. Thus, for this measure, the administrator of the application server have to classify the application according to its sensitivity to the sensing location (i.e. sensitive or stable). The class of the considered phenomena defines the way in which the proximity factor is calculated.

The calculated distance is used to assign a proximity score according to the class of the application. Firstly, we consider a stable phenomenon which is stable in different locations in the sensing area. The same weight is assigned to all contributions which are captured inside the sensing area. Subsequently, a participant is assigned a proximity score δ_{p_i} which is either 1 or 0 to indicate his existence either inside or outside the sensing area respectively, as depicted in Equation 9 where r is the radius length of the sensing area.

$$\delta_{p_i} = \begin{cases} 0 & \text{where} \beta_{p_i} > r \\ 1 & \beta_{p_i} <= r \end{cases} \tag{9}$$

Alternatively, for location sensitive applications, we use the calculated distance β_{p_i} as an input to the inverse Gompertz function to calculate the proximity factor δp_i, as depicted in Equation 10. The output of this function is depicted in Figure 3. Through this function, participants are assigned maximum proximity scores when they are at the center of the sensing area. These scores decrease gradually as they go away from the center. For instance, a participant of distance zero ($\beta_{p_i} = 0$) is exactly in the center of the sensing area. Such participant is assigned the upper proximity score which is 1. Thus, this score allows the application server to trust more the contributions which originate nearby the center of the sensing area.

$$\delta_{p_i} = 1 - a \times e^{-be^{-c\beta}} \tag{10}$$

where a is the upper asymptote, b controls the displacement of the output along the x axis and c adjusts the growth rate of the function. a,b, and c is selected such that the function output matches the radius of the sensing area.

4) Trust Mapping: The calculated parameters are aggregated to calculate the $Trust$ of the considered contribution according to the definition of trust presented earlier. These parameters include the current contribution evaluation θ_{p_i} evaluated by the contribution evaluation module, the aggregated feedback α_{p_i}, the proximity factor δ_{p_i}, and the reputation score $\hat{R}_{p_i}$ assigned to the participant through the previous campaign, see Equation 11,

$$Trust = W_1 \times \theta_{p_i} + W_2 \times \alpha_{p_i} + W_3 \times \delta_{p_i} + W_4 \times \hat{R}_{p_i} \tag{11}$$

where $\sum_{i=1}^{4} W_i = 1$. $\hat{R}_{p_i}$ of a new participant is set to 0 in order not to give a new participant the ability to inject bad data to the system unless he behaves correctly for a period of time.

5) Reputation: The value of the reputation score $\hat{R}_{p_i}$ of the participant p_i is update to a new value R_{p_i}. The reputation update process depends on the quality score assigned to the participant contribution θ_{p_i}. If this score is greater than a predefined threshold τ, the participant is rewarded by increasing his reputation score with ϵ_r such that the output reputation does not exceed 1. Oppositely, if the contribution quality score is below this threshold, the participant is penalized by decreasing his reputation score with ϵ_p such that the reputation score is not less than 0. We set $\epsilon_p > \epsilon_r$, this makes adversaries aggressively penalized while reputation is built gradually. The reputation update process is formulated in the following equation.

$$R_{p_i} = \begin{cases} min\left\{\hat{R}_{p_i} + \epsilon_r, 1\right\} & \text{if } \theta_{p_i} >= \tau \\ max\left\{\hat{R}_{p_i} - \epsilon_p, 0\right\} & \text{if } \theta_{p_i} < \tau \end{cases} \tag{12}$$

The reason behind the exploitation of the contribution score to calculate the reputation of participants and not the trust score of his contribution is apparent for different reasons. Firstly, the trust score incorporates the proximity factor. While the calculation of this factor depends on the location of the participant which is constrained by the participant's habits and his daily activities. Thus, this factor only affects the reliability of the contribution but not the honesty of the participant. Thus,

it should not affect the participant's reputation score. Secondly, using the score of the current contribution allows us to update the reputation score of the participant such that it reflects the most recent behavior of the participant.

V. Experimental Evaluation

A. Experimental Setup

We implemented our scheme with a MATLAB simulation to measure the accuracy of our reputation and trust assessment method. Since the communication is not our concern, we implemented both the server and participants on the same machine.

In this simulation, we consider a noise monitoring application. Thus, we generated the data in accordance with a real noise levels described in [23]. We consider a sensing area where the mean μ of the noise data at the center of this area is 60 db. The noise waves are considered to attenuate due to scattering and absorption. The amplitude of the attenuated wave is calculated according to Equation 13.

$$A = A_0.e^{-\sigma Z} \tag{13}$$

where A_0 is the unattenuated noise wave at the center of the sensing area, A is the reduced amplitude after the wave has traveled a distance Z, while σ is the attenuation coefficient of the signal traveling in the Z direction. We consider $\sigma = 0.0023$. The term e is the exponential (or Napier's constant). The units of the attenuation value in Napier per meter can be converted to decibel/meter by dividing by 0.1151. We also consider a sensing area of radius 300m.

Good participant always send correct sensing data which commensurate with their location. However, we assume that, adversaries launch on-off attack. They send correct data to gain a high reputation scores, then they randomly send false sensing data. The probability by which an adversary sends correct data is referred to as its *nature*. We set the mean of false data to deviate from the correct data such that this mean corresponds to a different level of noise $(\mu + \mu/3)$ (i.e. 80 db). This means that an adversary contributes data which correspond to a completely different level of noise. Furthermore, we assume that, all false reports support each other. Hence, we consider the worst case when all adversaries collude to cause the biggest possible disturbance to the system. However, this case can be hardly met in realistic systems, but it enables us to evaluate our system under the most difficult circumstances. We generated a random sensing location for each participant such that they are uniformly distributed along the sensing area. Table I lists our default parameter settings.

B. On-off Attack

In this experiment, we test the system robustness against on-off attack. We measure how reputation and trust scores of an on-off attacker are affected by his nature. We study the behavior of five adversaries with different values of nature 0, 0.2, 0.5, 0.8. The nature represents the probability by which an on-off attacker sends correct data. To test the worst case, we assume that the five adversaries behaved in good manner

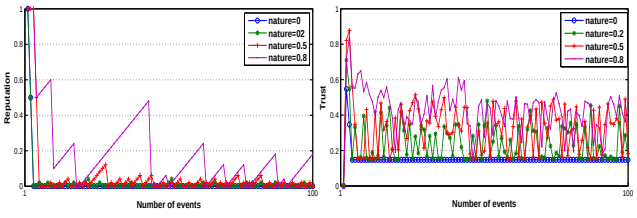
Parameter	Value
Number of participant for each task *NP*	100
The correct noise amplitude at the center μ	60db
The value of adversary noise amplitude $(\mu + \mu/3)$	80db
W_1	0.4
W_2	0.0
W_3	0.2
W_4	0.4
τ	0.4
r	300m
a, b, c	1, 10, 0.3
σ	0.0023
ϵ_r	0.02
ϵ_p	0.5

TABLE I: Default Parameter Settings

until their reputation scores have reached 1 before the test. We then run this experiment for 100 tasks.

In Figure 4 (a), we can see that the reputation score of an adversary degrades. The reputation score of adversaries with nature 0, 0.2, and 0.5, drops down very quickly until it reaches 0. While the reputation score of adversaries with higher nature (e.g. 0.8) still drop down more slowly. It drops to a very low level even if the adversary sends correct data with a very high probability (i.e. 0.8 in this case). An adversary is severely punished for each bad transaction but rewarded gradually for good ones. Thus, bad transactions have larger influence on the reputation score.

We examine the computed trust scores assigned to reports sent by those adversaries. Figure 4 (b) depicts these results. It is obvious that, the trust scores of reports received from adversary with nature 0 are usually assigned a score around 0.2 the minimum possible value of trust. This value results when the reputation score is 0 and contribution quality is very bad (i.e $\theta_{p_i} \approx 0.37$). Thus, $Trust \Longrightarrow (0.4 \times 0.37 + 0.4 \times 0 + 0.2 \times \delta_{p_i})$, where $\delta_{p_i} \in [0, 1]$, $Trust \Longrightarrow [0.15, 0.25]$. While the trust of reports from adversaries with nature 0.2, 0.5, and 0.8 fluctuates much more since they sometimes send correct data. However, their trust scores do not usually exceed 0.5. That is the system does not trust an adversary even if he sends correct reports with high probability (e.g. 0.5 and 0.8).



(a) (b)

Fig. 4: Impact of the nature on the reputation and trust of adversaries in the proposed DTSRS system

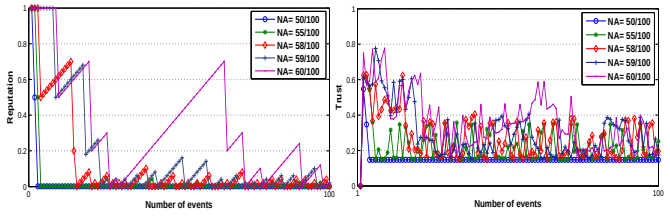
C. Collusion attack

In the following experiment, we measure the resistance of our system under collusion attack. So, we need to define the number of adversaries NA with which the system fails to detect the adversaries under the predefined test setup. In this test, we set the nature of all adversaries to be 0.

Intuitively, the system perfectly detects adversaries as long as the total number of adversaries is less than the size of the trusted set. In this case, there is no intersection between the trusted set and adversaries. We need to test to which extent the trusted set can involve some adversaries and still properly detect adversaries. Thus, we vary the number of adversaries NA in the campaign as 50, 55, 58, 59 and 60 where the trusted set size is 60. That is the trusted set involves 10, 15, 18, 19, and 20 adversary respectively.

Figure 5 (a) depicts the results of this test. It is evident that, the reputation scores of adversaries rapidly drop down to reach zero. However, this drop becomes slow with the increase in the number of adversaries from 58, 59, and 60. However, the system fails to identify adversaries while the number of adversaries NA surpasses 60 adversaries.

Figure 5 (b) shows the trust of the contributions of those participants. It is clear that, adversaries' contributions are usually assigned low trust scores (i.e [0.2,0.4]) even if the number of adversaries reaches 60% of the total number of participants. However, when the number of adversaries reaches 60% of the total number of participants, the trust of adversary's contribution fluctuates. This means that the DTSRS proposed system does not trust adversaries' contributions (i.e. $Trust < 0.5$) until the number of adversaries reaches 60% of the total number of the participants under the current test setup. This reflect that our system is resistant under collusion attack.



(a) (b)

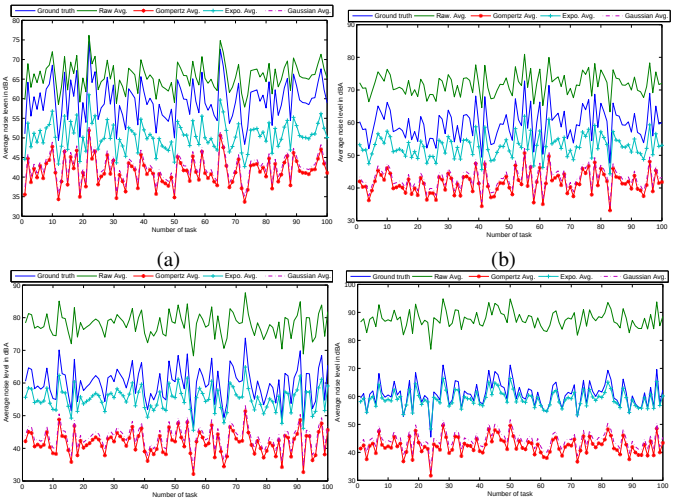
Fig. 5: Impact of the number of adversaries on reputation and trust of an adversary in the proposed DTSRS system

D. Comparison

We measure the accuracy of the aggregated data and how much it agrees with the ground truth. In this experiment, we evaluate the usage of the exponential distribution as a mapping function compared with the functions used in the state-of-the-art such as Gompertz in [8], and Gaussian function in [10]. We compute the scores $Trust$ assigned to each contribution C_{p_i} according to each function f. These scores are then used to calculate the average of the collected contribution of each task as shown in Equation 14. We run the experiment for 100 tasks.

$$v_{f,t} = \frac{\sum_{i=1}^{NC} (Trust_f(C_{p_i}) \times C_{p_i})}{NC} \quad (14)$$

where f is the reputation function (e.g. f can be exponential, Gompertz, or Gaussian), i is the contribution number and NC is the total number of contributions available for the task t. We consider there are 30 adversaries with nature 0 out of 100 participants within the campaign.



(a) (b) (c) (d)

Fig. 6: Average aggregated noise level where the mean of a correct data is 60 dBA and the mean of a false data is 80, 100, 120, 150 depicted in sub-figures a, b,c, and d respectively

We also include the raw average, it is calculated by averaging the collected contributions for the same task without incorporating any additional scores. We run the experiment while the number of adversaries is 30/100 with nature 0. We consider false data with four different mean values 80, 100, 120, and 150 dBA, each one is considered in a separate run. The large values of false data cause more disruption for the aggregated data. We run the experiment four times while each run contains 100 task and there are 100 contribution for each task. We measure the deviation of the average calculated according to Equation 14 in each task from the correct ground truth which has a mean of 60 dBA.

The results of this experiment are shown in Figure 6. The closeness of the calculated average to the ground truth' plot indicates the accuracy of the mapping function for assigning appropriate trust score to each contribution. As it can be observed, the raw average is significantly different from the ground truth since all contributions are equally considered. By looking at the raw average data in the different sub-figure, it becomes worse with the increase of the mean value of false data in sub-figures a, b, c, and d. Both Gompertz and Gaussian averages also deviate significantly from the ground truth. By looking at the sub-figure a, b, c, and d, the Gompertz and Gaussian based averages nearly achieve the same deviation from the ground truth whatever the mean of false data. On the other hand, the average calculated based on exponential distribution not only approximates the ground truth more closely in all sub-figures a, b, c, and d. Additionally, the performance of exponential distribution based average enhances from a to b and c, and it has the best performance in d. That is the exponential based average system can perform better when the mean value of false data significantly deviate from the correct ones. This means that our system has better capabilities to reflect the nature of the ground truth data even

if the system faces a massive disruption.

This is because our proposed exponential distribution has much sharper degradation which allows it to highly consider contributions which has a slight deviation from the correct ones (i.e. the ones that have a deviation more close to 0). In addition, it allows to aggressively assign bad scores to the ones which have much more deviation. Thus, they are less considered. Therefore, the calculated data average has better capabilities to reflect the ground truth data than the Gompertz and Gaussian.

VI. Conclusion

In this paper, we propose the DTSRS reputation system for participatory sensing applications. The system depends on a dynamic trusted set of participants to identify the good data in each campaign. We experimentally evaluated the system by incorporating it within a simulated system for noise monitoring participatory sensing application. The results indicate that DTSRS system accurately assesses the quality of participants' contributions. It exposes the average of the aggregated data to the minimum possible noise. In addition, the system clearly identifies adversaries even if the number of colluding adversaries reaches 60% of the total number of participants in the campaign. Furthermore, adversaries who launch on-off attack are clearly identified even if they contribute good data with high probability (e.g. 0.8). Therefore, the proposed DTSRS reputation system can defend against corruption, On-Off, and collusion attacks which are not considered in literature. In a future work, we target to manage both the conflicting objectives of trust assessment and privacy preservation of participants in participatory sensing environment. Thus, we are going to incorporate the DTSRS reputation system within a privacy preserving framework.

References

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

[2] 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, Sep. 2010.

[3] E. Kanjo, J. Bacon, D. Roberts, and P. Landshoff, "Mobsens: Making smart phones smarter," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 50–57, Oct 2009.

[4] L. Nachman, A. Baxi, S. Bhattacharya, V. Darera, N. Kodalapura, V. Mageshkumar, S. Rath, and R. Acharya, "Jog falls: A pervasive healthcare platform for diabetes management," in *Pervasive Computing*, vol. 6030, May 2010, pp. 94–111.

[5] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: A participatory sensing fuel-efficient maps application," *MobiSys*, 2010, pp. 151–164.

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

[7] H. Mousa, S. B. Mokhtar, O. Hasan, O. Younes, M. Hadhoud, and L. Brunie, "Trust management and reputation systems in mobile participatory sensing applications," *Computer Networks*, vol. 90, no. C, pp. 49–73, Oct. 2015.

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

[9] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Enabling reputation and trust in privacy-preserving mobile sensing," *IEEE Transactions on Mobile Computing*, vol. 99, p. 1, 2014.

[10] A. Manzoor, M. Asplund, M. Bouroche, S. Clarke, and V. Cahill, "Trust evaluation for participatory sensing," in *MobiQuitous*, 2012, pp. 176–187.

[11] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos, "Loci: fast outlier detection using the local correlation integral," in *International Conference on Data Engineering, 2003*, 2003, pp. 315–326.

[12] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, May 2000.

[13] C. Beecks, M. S. Uysal, and T. Seidl, "A comparative study of similarity measures for content-based multimedia retrieval," in *ICME*, 2010, pp. 1552–1557.

[14] H. Amintoosi and S. S. Kanhere, "A reputation framework for social participatory sensing systems," *MONET*, vol. 19, no. 1, pp. 88–100, 2014.

[15] ——, "A trust framework for social participatory sensing systems," in *MobiQuitous*, 2012, pp. 237–249.

[16] R. R. Kalidindi, K. Raju, V. V. Kumari, and C. S. Reddy, "Trust based participant driven privacy control in participatory sensing," *CoRR*, vol. abs/1103.4727, 2011.

[17] 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*, 2009.

[18] A. Dua, W. Hu, and N. Bulusu, "Demo abstract: A trusted platform based framework for participatory sensing," in *IPSN*, 2009

[19] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall, "Toward trustworthy mobile sensing," in *HotMobile '10*, 2010, pp. 31–36.

[20] 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," *Department of Computer Science*, 2011.

[21] 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.

[22] A. Jøsang and J. Golbeck, "Challenges for robust of trust and reputation systems," in (STM), 2009.

[23] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, "Fundamentals of acoustics," *4th Edition, by Lawrence E. Kinsler, Austin R. Frey, Alan B. Coppens, James V. Sanders, pp. 560. ISBN 0-471-84789-5. Wiley-VCH, December 1999.*, vol. 1, 1999.