Towards Data Poisoning Attacks in Crowd Sensing Systems

Chenglin Miao
State University of New York at Buffalo, NY, USA
cmiao@buffalo.edu

Qi Li
University of Illinois at Urbana-Champaign, IL, USA
qili5@illinois.edu

Houping Xiao
State University of New York at Buffalo, NY, USA
houpingx@buffalo.edu

Wenjun Jiang
State University of New York at Buffalo, NY, USA
wenjunji@buffalo.edu

Mengdi Huai
State University of New York at Buffalo, NY, USA
mengdihu@buffalo.edu

Lu Su*
State University of New York at Buffalo, NY, USA
lusu@buffalo.edu

ABSTRACT

With the proliferation of sensor-rich mobile devices, crowd sensing has emerged as a new paradigm of collecting information from the physical world. However, the sensory data provided by the participating workers are usually not reliable. In order to identify truthful values from the crowd sensing data, the topic of truth discovery, whose goal is to estimate each worker's reliability and infer the underlying truths through weighted data aggregation, is widely studied. Since truth discovery incorporates workers' reliability into the aggregation procedure, it shows robustness to the data poisoning attacks, which are usually conducted by the malicious workers who aim to degrade the effectiveness of the crowd sensing systems through providing malicious sensory data. However, truth discovery is not perfect in all cases. In this paper, we study how to effectively conduct two types of data poisoning attacks, i.e., the availability attack and the target attack, against a crowd sensing system empowered with the truth discovery mechanism. We develop an optimal attack framework in which the attacker can not only maximize his attack utility but also disguise the introduced malicious workers as normal ones such that they cannot be detected easily. The desirable performance of the proposed framework is verified through extensive experiments conducted on a real-world crowd sensing system.

CCS CONCEPTS

• **Security and privacy** → **Systems security**; • **Information systems** → *Information systems applications*;

KEYWORDS

Data poisoning, crowd sensing, truth discovery

ACM Reference Format:
Chenglin Miao, Qi Li, Houping Xiao, Wenjun Jiang, Mengdi Huai, and Lu Su. 2018. Towards Data Poisoning Attacks in Crowd Sensing Systems. In *Mobihoc '18: The Nineteenth International Symposium on Mobile Ad Hoc Networking and Computing, June 26–29, 2018, Los Angeles, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209582.3209594

*L. Su is the corresponding author.

Mobihoc '18, June 26–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5770-8/18/06…$15.00
https://doi.org/10.1145/3209582.3209594

1 INTRODUCTION

With the proliferation of increasingly capable human-carried mobile devices (e.g., smartphones, smartwatches, and smartglasses) equipped with a plethora of sensors (e.g., accelerometer, GPS, camera, compass), crowd sensing has emerged as a new way of collecting information from the physical world. In crowd sensing applications, humans work as sensor carriers or even the sensors, and the collection of sensory data is outsourced to a large crowd of participating users (also called *workers*) carrying sensing devices. Recently, a large variety of crowd sensing systems [11, 12, 17, 25, 27, 29, 40, 41, 44] have been developed and they serve a wide spectrum of applications with significant societal and economic impacts, including healthcare, urban and environment monitoring, smart transportation, crowd wisdom, etc.

Although crowd sensing provides an effective way to obtain useful information from the physical world, the sensory data collected from participating workers are not always trustworthy. Due to the openness of the crowd sensing systems, the malicious parties can easily conduct malicious attacks. One important form of attacks is called *data poisoning*, where an attacker tries to degrade the effectiveness of the crowd sensing systems through creating or recruiting a group of malicious workers and letting them submit malicious data. In this paper, we focus on two types of data poisoning attacks: the *availability attack* and the *target attack*. In the availability attack, the attacker tries to disturb the final results as much as possible through manipulating the malicious workers' sensory data. In the target attack, the attacker aims to skew the final results to predetermined target values. Traditionally, when multiple workers provide conflicting observations on the same object, the final results are usually obtained using aggregation methods such as majority voting. However, such traditional aggregation methods are much vulnerable to these two types of attacks, *as they treat all the participating workers equally, including the malicious ones.* Recently, an advanced aggregation approach, truth discovery [21–24, 33, 36, 37, 42], has been widely studied, as it can distinguish workers with varying reliability degrees. The principle of truth discovery is established by the following intuition: A worker will be assigned a high weight if his data are close to the aggregated results, and the data of a worker will be counted more in the aggregation procedure if he has a high weight.

If the attacker can create or recruit many malicious workers, the attack goal is relatively easy to achieve, especially if the malicious workers outnumber the normal workers. However, when the attacker has limited resources, which is more often in real life, the attack strategy becomes very important. Suppose the observations from workers are categorical (e.g., the model and make of the car that hit the old lady and ran), then one intuitive attack strategy is to let all the malicious workers report the answer with the second highest vote count (for the availability attack), or provide votes to the target answer (for the target attack). This strategy may be the optimal choice if the aggregation results are derived by majority voting. However, the story is much more complicated if the truth discovery approach is used for aggregation. Thanks to the ability of distinguishing workers with different reliability degrees, the truth discovery approach can easily detect the malicious workers, since they always disagree with the majority even when there is no chance to win, and therefore, assign low weights to the malicious workers. Consequently, the impact of the malicious workers will be greatly reduced, and the attack goal cannot be achieved.

Although truth discovery methods can tolerate the malicious behaviors of the workers to some degree and effectively improve the aggregation results, it is not perfect in all cases. In this paper, we propose an optimal attack framework that can take down a sensing system even with truth discovery empowered. Compared with the aforementioned native attack strategy, the strategy derived from the proposed optimal attack framework makes the malicious workers behave "smarter". They can successfully disguise themselves as normal workers. If there is little hope to achieve the attack goal on some objects, they will tend to agree with the normal workers on those objects to gain higher weights, and in turn, can exert stronger impact on other objects.

In our design, the optimal attack strategy is found by solving a bi-level optimization problem where the objective is to maximize the attack utility, in other words, the total number of the objects whose true values are skewed. As the attack goal is either achieved or not on one object, the attack utilities are discrete values, making it hard to solve the optimization problem. To handle this challenge, we use a continuous and differentiable sigmoid function to approximate the discrete attack utilities. Then we derive the optimal attack strategy by iteratively solving the upper-level and lower-level optimization problems, where the former adopts the gradient ascent method and the latter is solved by block coordinate decent method.

In summary, we make the following contributions in this paper:

- We propose an optimal data poisoning attack framework, based on which the attacker can not only maximize his attack utility but also successfully disguise the attack behaviors when attacking a crowd sensing system employing the truth discovery mechanism.
- Two types of data poisoning attack goals, i.e., the availability attack and the target attack, are taken into account in the proposed optimal attack framework.
- We build a real-world crowd sensing system to evaluate the proposed optimal attack framework. The results clearly demonstrate the advantages of the proposed framework.

2 PROBLEM SETTING

In this section, we first introduce some concepts adopted in this paper. Then, the problem setting is described.

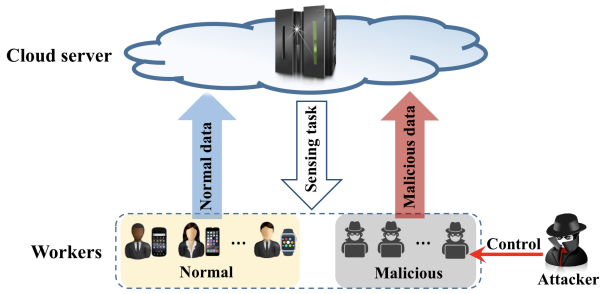

Figure 1: The crowd sensing system under attack

Definition 2.1. An *object* is an item or a phenomenon of interest, such as a question and the congestion level of a road segment. Its true information is defined as the *ground truth*.

Definition 2.2. An *observation* (or *sensory data*) is the data or information describing a particular object collected by a participating worker. In this paper, we focus on categorical observation data, and will leave the study on the data of other types as future work.

The crowd sensing system considered in this paper consists of a *cloud server* and some *participating workers*. The cloud server is a platform which holds some sensing tasks. In each sensing task, usually there are multiple objects needed to be observed. The participating workers are the mobile device users who carry out the sensing tasks and provide their observations to the cloud server. After collecting the observations from all the workers, the cloud server needs to estimate the true information (i.e., truth) of each object by conducting the truth discovery algorithm.

Suppose there is an *attacker* who aims to attack the crowd sensing system empowered with the truth discovery algorithm. As shown in Figure 1, the attacker cannot manipulate the observations of the *normal workers* who carry out the sensing tasks without any malicious behavior, but he can create or recruit a group of *malicious workers* and conduct attacks by carefully designing their observations. When conducting the availability attack, the attacker wants to maximize the error of the truth discovery algorithm running on the crowd sensing system, and eventually render the discovered truths useless. When conducting the target attack, the attacker aims to skew the final estimated object truths calculated by the cloud server to certain target values. In this paper, we assume that the attacker has complete knowledge of the truth discovery algorithm and the sensory data from normal workers. This assumption enables a robust assessment of the vulnerability of the crowd sensing system. Additionally, it is entirely possible that the attacker can get the normal workers' data through eavesdropping the communications between normal workers and the cloud server.

We formally define the problem addressed in this paper as: Suppose the cloud server outsources a sensing task to a group of participating workers. In this sensing task, there are M objects which are observed by K normal workers. We use $W = \{w_k\}_{k=1}^{K}$ to denote the weights (i.e., reliability degrees) of the normal workers. The sensory data of normal workers are denoted as $X = \{x_m^k\}_{m,k=1}^{M,K}$, in which x_m^k denotes the observation of the k-th normal worker for the m-th object. The ground truth of each object is unknown by any party in the crowd sensing system, and the cloud server needs to

calculate the estimated values $X^* = \{x_m^*\}_{m=1}^M$ of the ground truths for all the objects. Assume that there are K' malicious workers that are created or recruited by the attacker in the crowd sensing system, and their weights are denoted as $\widetilde{W} = \{\widetilde{w}_{k'}\}_{k'=1}^{K'}$. We use $\widetilde{X} = \{\widetilde{x}_m^{k'}\}_{k',m=1}^{K',M}$ to denote the observations of all the malicious workers and $\widetilde{x}_m^{k'}$ is the observation of the k'-th malicious worker for the m-th object. Our goal in this paper is to *find an optimal attack strategy (i.e., an optimal $\widetilde{X}$) such that the attack goal of the attacker can be achieved as much as possible.*

3 PRELIMINARY

In order to get an accurate final observation for each object in the sensing task, it is important for the crowd sensing systems to properly aggregate the collected observations. The traditional aggregation method is majority voting, which treats the observations made by the majority of workers as the final object truth. The drawback of majority voting is that it treats all the participating workers equally. In practice, however, the information quality usually varies among different participating workers. Ideally, a sophisticated aggregation approach should be able to identify high-quality workers and use them to improve the aggregated results. However, a crucial challenge is that the workers' quality is usually unknown *a priori* in practice. To tackle this challenge, truth discovery [23] emerges as a hot topic since it can automatically estimate worker quality from the data in the form of worker weights.

Although different truth discovery approaches have been designed for different scenarios, they share the same basic idea: A worker ought to be assigned a high weight if his observations are close to the aggregated results, and the observations of a worker ought to be counted more in the aggregation procedure if he has a high weight. In this paper, we consider the widely adopted truth discovery method CRH [22, 24], in which an optimization framework is proposed to minimize the weighted deviation from the workers' observations to the estimated truths:

$$\begin{aligned} \min_{X^*, W} \quad & f(X^*, W) = \sum_{k=1}^{K} w_k \sum_{m \in O_k} d(x_m^k, x_m^*) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \exp(-w_k) = 1, \end{aligned} \tag{1}$$

where O_k is the set of objects observed by the k-th normal worker (in this section, we assume that there are no malicious workers in the crowd sensing system); $d(\cdot)$ is the loss function to measure the distance between workers' observations and the estimated truths.

In this paper, we consider the scenario where the sensory data are categorical. That is, for each object, the worker would choose an answer from one of the C candidate answers. We use $x_m^k = (0, ..., \underset{q}{1}, ..., 0)^T$ to denote that the k-th worker selects the q-th candidate answer for the m-th object. Then the distance between the observation vector x_m^k and the estimated truth vector x_m^* is defined as:

$$d(x_m^k, x_m^*) = (x_m^k - x_m^*)^T (x_m^k - x_m^*) = \sum_{c=1}^{C} (x_{mc}^k - x_{mc}^*)^2, \tag{2}$$

where x_{mc}^k and x_{mc}^* represent the c-th value in vector x_m^k and vector x_m^*, respectively.

CRH aims to learn the the estimated values X^* of the truths and worker weights W together by optimizing the objective function in Eqn. (1). In order to achieve the goal, block coordinate descent approach [4] is adopted and the following two steps are iteratively conducted until the convergence criterion is satisfied.

Step I: Truths Update. In this step, the workers' weights W are fixed, and the estimated object truths X^* are updated according to:

$$x_m^* = \frac{\sum_{k \in U_m} w_k x_m^k}{\sum_{k \in U_m} w_k}, \tag{3}$$

where U_m is the set of normal workers who observe the m-th object.

Clearly, the estimated object truth x_m^* is a vector of continuous values. It can be viewed as a probability vector in which each element represents the probability of the corresponding candidate answer being true. For example, suppose x_m^* equals to $(0.1, 0.7, 0.1, 0.1)$, then it implies that with 70% probability the second candidate answer is the true answer for the m-th object, and the probabilities of the others are all 10%. In this case, we assign the final estimated truth as the candidate answer with the largest value in vector x_m^*.

Step II: Worker Weights Update. In this step, the estimated object truths X^* are fixed, and the participating workers' weights W are updated according to:

$$w_k = \log\left(\frac{\sum_{l=1}^{K} \sum_{m \in O_l} d(x_m^l, x_m^*)}{\sum_{m \in O_k} d(x_m^k, x_m^*)}\right), \tag{4}$$

where O_l is the set of objects observed by the l-th normal worker.

The pseudo code of the truth discovery procedure is summarized as Algorithm 1.

Algorithm 1: Truth Discovery framework

Input: Observations from K workers: $\{x_m^k\}_{m,k=1}^{M,K}$
Output: Estimated truths for M objects: $\{x_m^*\}_{m=1}^M$

```
Initialize the workers' weights uniformly;
repeat
    for each object m do
        Update the estimated object truth x*_m based on Eqn. (3);
    end
    for each worker k do
        Update the k-th worker's weight w_k based on Eqn. (4);
    end
until Convergence criterion is satisfied;
return The estimated object truths {x*_m}_{m=1}^M;
```

4 OPTIMAL ATTACK FRAMEWORK

In this section, we present our optimal attack framework against the crowd sensing systems with truth discovery empowered. We first analyze the effect of malicious workers on the truth discovery framework in section 4.1. Then two types of data poisoning attacks (i.e., the availability attack and the target attack) are discussed in section 4.2 and 4.3, respectively.

4.1 Truth Discovery with Malicious Workers

We use $\widehat{X}^* = \{\widehat{x}_m^*\}_{m=1}^M$ to denote the estimated object truths after the data poisoning attack. After taking the malicious workers into account, the truth discovery framework in Eqn.(1) becomes:

$$\min_{\widehat{X}^*, W, \widetilde{W}} f(\widehat{X}^*, W, \widetilde{W}) = \sum_{k=1}^{K} w_k \sum_{m \in O_k} d(x_m^k, \widehat{x}_m^*) + \sum_{k'=1}^{K'} \widetilde{w}_{k'} \sum_{m \in \widetilde{O}_{k'}} d(\widetilde{x}_m^{k'}, \widehat{x}_m^*) \quad (5)$$
$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) + \sum_{k'=1}^{K'} \exp(-\widetilde{w}_{k'}) = 1,$$

where $\widetilde{O}_{k'}$ is the set of the objects observed by the k'-th malicious worker. Here we decompose the participating workers into normal and malicious ones for the purpose of analyzing the effect of malicious workers on the estimated object truths. However, note that from the cloud server's perspective, it is not aware of the attack and cannot differentiate the two types of workers when conducting truth discovery. Based on the block coordinate descent method which is adopted in the original truth discovery framework, the optimal solution $\{\widehat{X}^*, W, \widetilde{W}\}$ can be calculated by iteratively conducting the following two steps until the convergence criterion is satisfied.

Step I: Truths Update. In this step, we first fix the weights of normal and malicious workers (i.e., W and $\widetilde{W}$), then update the estimated object truths $\widehat{X}^*$ according to

$$\widehat{x}_m^* = \frac{\sum_{k \in U_m} w_k x_m^k + \sum_{k' \in \widetilde{U}_m} \widetilde{w}_{k'} \widetilde{x}_m^{k'}}{\sum_{k \in U_m} w_k + \sum_{k' \in \widetilde{U}_m} \widetilde{w}_{k'}}, \quad (6)$$

where $\widetilde{U}_m$ is the set of malicious workers who observe the m-th object. As described in section 3, $\widehat{x}_m^*$ is a vector in which each element represents the probability of the corresponding candidate answer being true after the attack. The c-th element in this vector is updated as

$$\widehat{x}_{mc}^* = \frac{\sum_{k \in U_m} w_k x_{mc}^k + \sum_{k' \in \widetilde{U}_m} \widetilde{w}_{k'} \widetilde{x}_{mc}^{k'}}{\sum_{k \in U_m} w_k + \sum_{k' \in \widetilde{U}_m} \widetilde{w}_{k'}}, \quad (7)$$

where $\widetilde{x}_{mc}^{k'}$ is the c-th value in vector $\widetilde{x}_m^{k'}$.

Step II: Worker Weights Update. In this step, the estimated object truths $\widehat{X}^* = \{\widehat{x}_m^*\}_{m=1}^M$ are fixed. Then we update the weights of normal and malicious workers (i.e., W and $\widetilde{W}$) as

$$w_k = \log\left(\frac{\sum_{l=1}^{K} \sum_{m \in O_l} d(x_m^l, \widehat{x}_m^*) + \sum_{l'=1}^{K'} \sum_{m \in \widetilde{O}_{l'}} d(\widetilde{x}_m^{l'}, \widehat{x}_m^*)}{\sum_{m \in O_k} d(x_m^k, \widehat{x}_m^*)}\right) \quad (8)$$

$$\widetilde{w}_{k'} = \log\left(\frac{\sum_{l=1}^{K} \sum_{m \in O_l} d(x_m^l, \widehat{x}_m^*) + \sum_{l'=1}^{K'} \sum_{m \in \widetilde{O}_{l'}} d(\widetilde{x}_m^{l'}, \widehat{x}_m^*)}{\sum_{m \in \widetilde{O}_{k'}} d(\widetilde{x}_m^{k'}, \widehat{x}_m^*)}\right), \quad (9)$$

where $\widetilde{O}_{l'}$ is the set of objects observed by the l'-th malicious worker.

From the above equations, we can see the estimated object truths $\widehat{X}^* = \{\widehat{x}_m^*\}_{m=1}^M$ are only dependent on the observations of malicious workers (i.e., $\widetilde{X} = \{\widetilde{x}_m^{k'}\}_{m,k'=1}^{M,K'}$) once the data of normal workers are given. In this way, the attacker can attack truth discovery algorithm by elaborately designing the observations of malicious workers.

4.2 Availability Attack

In the availability attack, the attacker aims to maximize the error of the crowd sensing systems where the observations from multiple workers are aggregated by the truth discovery algorithm, and eventually render them useless. In other words, the attacker tries to make the deviation between the outputs of truth discovery (Algorithm 1) before and after the availability attack as much as possible. More specifically, if the final truth discovery result on an object is changed after the attack, it means that the attack on this object succeeds. Otherwise, the attack on this object fails. In this section we discuss how to find the optimal attack strategy from the perspective of the attacker so that the attack can succeed on as many objects as possible.

Given the number of malicious workers created or recruited by the attacker and the objects they can observe, the attacker needs to find the optimal assignments for each malicious worker's observations to conduct the availability attack. Let's denote the final estimated answers for the m-th object before and after the attack as x_m^{*f} and $\widehat{x}_m^{*f}$ respectively. We can formulate the goal of the availability attack into an optimization problem as follows:

$$\max_{\widetilde{X}} \sum_{m=1}^{M} \mathbb{1}(\widehat{x}_m^{*f} \neq x_m^{*f}) \quad (10)$$
$$\text{s.t.} \quad \{\widehat{X}^{*f}, W, \widetilde{W}\} = \operatorname*{argmin}_{\widehat{X}^{*f}, W, \widetilde{W}} f(\widehat{X}^{*f}, W, \widetilde{W})$$
$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) + \sum_{k'=1}^{K'} \exp(-\widetilde{w}_{k'}) = 1,$$

where $\widehat{X}^{*f} = \{\widehat{x}_m^{*f}\}_{m=1}^M$ are the final estimated answers after the attack and $\mathbb{1}(\cdot)$ is the indicator function. In this optimization problem, the truth discovery framework Eqn.(5) becomes a constraint. This is a bi-level optimization problem [2]. The optimization over malicious observations $\widetilde{X}$ is the upper-level problem, and the optimization over $\{\widehat{X}^{*f}, W, \widetilde{W}\}$ given $\widetilde{X}$ is the lower-level problem. x_m^{*f} is the final aggregation result (calculated based on normal workers' data) before the attack, and it is a constant once the normal workers' sensory data are given. $\widehat{x}_m^{*f}$ depends on the attack strategy (i.e., $\widetilde{X}$) and can be different as the attack strategy varies.

For each object, the malicious workers need to pick one candidate answer. An intuitive attack strategy is to choose the answer with the second highest probability to be true. The reason is simple, this answer has the most chance to win over the answer with the highest probability. However, this attack strategy may not be the optimal choice under truth discovery mechanism. Let's consider the following example: If the margin between the answers with the highest and the second highest probability is too large for an object (e.g., 100 votes V.S. 2 votes), it is impossible for the limited number of malicious workers to change the aggregation result on this object. Moreover, since the malicious workers always disagree with the majority, the truth discovery algorithm can detect them easily and assign them with low weights. Consequently, the impact of the malicious workers on other objects also decreases, and thus may fail on all objects. To address this challenge, we take the truth discovery framework as a constraint in our designed optimization problem (10). Then the weights of malicious workers will be taken into account during the procedure of finding the optimal attack strategy. As a result, we may find a better attack strategy compared with the intuitive one. The optimal attack may sacrifice on some of the objects where there is little chance to succeed, and agree with the majority workers on those objects. The benefit of doing so is

that the truth discovery algorithm may consider them as normal workers or even good workers and increase their weights, and eventually increase their impact on other objects.

Since the answer with the second highest probability has the most chance to win over the answer with the highest probability, we only consider the change on these two answers. We reformulate problem (10) as:

$$\max_{\widetilde{X}} \sum_{m=1}^{M} \frac{1}{2}\{1 - \mathrm{sgn}[(x^*_{mc_2} - x^*_{mc_1}) \cdot (\widehat{x}^*_{mc_2} - \widehat{x}^*_{mc_1})]\} \tag{11}$$

$$\text{s.t.} \quad \{\widehat{X}^*, W, \widetilde{W}\} = \underset{\widehat{X}^*, W, \widetilde{W}}{\mathrm{argmin}} \ f(\widehat{X}^*, W, \widetilde{W})$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) + \sum_{k'=1}^{K'} \exp(-\widetilde{w}_{k'}) = 1$$

$$\{\widetilde{x}^{k'}_{mc}\}^{K',M,C}_{k',m,c=1} \in \{0, 1\} \tag{12}$$

where c_1 and c_2 indicate the answers with the highest and the second highest element in the probability vector (i.e., x^*_m) output by the truth discovery algorithm before the attack. Constraint (12) is used to limit each element $\widetilde{x}^{k'}_{mc}$ to 0 or 1. This optimization problem reflects the following idea: After the attack, if the answer with the second highest probability does not win over the answer with the highest probability, the attack fails on this object. Consequently, there is no gain in the objective value.

In the objective function (11), $(x^*_{mc_2} - x^*_{mc_1}) < 0$ when the normal workers' data are given. Then we can know:

$$\mathrm{sgn}[(x^*_{mc_2} - x^*_{mc_1}) \cdot (\widehat{x}^*_{mc_2} - \widehat{x}^*_{mc_1})] = \begin{cases} 1 & if\ \widehat{x}^*_{mc_2} < \widehat{x}^*_{mc_1} \\ 0 & if\ \widehat{x}^*_{mc_2} = \widehat{x}^*_{mc_1} \\ -1 & if\ \widehat{x}^*_{mc_2} > \widehat{x}^*_{mc_1}. \end{cases} \tag{13}$$

However, Eqn. (13) is not continuous, and this makes it difficult to solve the above optimization problem. A potential way to address this challenge is to approximate the objective function (11) by a continuous and differentiable function. Considering that function $u_1(x) = \frac{1}{2}(1 - \mathrm{sgn}\, x)$ can be well approximated by function $u_2(x) = 1 - \frac{1}{1+\exp(-\theta x)}$ when θ (i.e., the steepness of the curve) is set to an appropriate value, we approximate the objective function (11) by the following objective function:

$$\max_{\widetilde{X}} \sum_{m=1}^{M} \{1 - \frac{1}{1 + \exp[-\theta(x^*_{mc_2} - x^*_{mc_1})(\widehat{x}^*_{mc_2} - \widehat{x}^*_{mc_1})]}\}. \tag{14}$$

From the perspective of the attacker, $\sum_{m=1}^{M}\{1 - \frac{1}{1+\exp[-\theta(x^*_{mc_2} - x^*_{mc_1})(\widehat{x}^*_{mc_2} - \widehat{x}^*_{mc_1})]}\}$ in the objective function (14) can be treated as his utility, and he needs to find an appropriate attack strategy such that the utility can be maximized.

When solving the above optimization problem, we still have another challenge, i.e., the value of each element (i.e., $\widetilde{x}^{k'}_{mc}$) in $\widetilde{X}$ is categorical, which makes it difficult to solve the upper-level problem. Here we treat each observation of malicious workers (i.e., $\widetilde{x}^{k'}_{m}$) as a probability vector and relax the value of $\widetilde{x}^{k'}_{mc}$ to the range $(0, 1)$. In this way, we can solve the optimization problem according to the gradient-based methods. Please note that the summation of all the elements in vector $\widetilde{x}^{k'}_{m}$ should be 1, and the candidate answer with the largest value in this vector will be submitted to the cloud server. Then the following optimization problem needs to be solved in order to maximize the attacker's utility.

$$\max_{\widetilde{X}} \ g(\widetilde{X}) = \sum_{m=1}^{M} \{1 - \frac{1}{1 + \exp[-\theta(x^*_{mc_2} - x^*_{mc_1})(\widehat{x}^*_{mc_2} - \widehat{x}^*_{mc_1})]}\} + \delta_1 \sum_{k'=1}^{K'} \sum_{m \in \widetilde{O}_{k'}} \sum_{c=1}^{C} \log \widetilde{x}^{k'}_{mc} + \delta_2 \sum_{k'=1}^{K'} \sum_{m \in \widetilde{O}_{k'}} \sum_{c=1}^{C} \log(1 - \widetilde{x}^{k'}_{mc}) \tag{15}$$

$$\text{s.t.} \quad \{\widehat{X}^*, W, \widetilde{W}\} = \underset{\widehat{X}^*, W, \widetilde{W}}{\mathrm{argmin}} \ f(\widehat{X}^*, W, \widetilde{W}) \tag{16}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) + \sum_{k'=1}^{K'} \exp(-\widetilde{w}_{k'}) = 1$$

$$\sum_{c=1}^{C} \widetilde{x}^{k'}_{mc} = 1, \ k' = 1, \ldots, K' \text{ and } m = 1, \ldots, M. \tag{17}$$

The objective function $g(\widetilde{X})$ contains three terms: The first term is the utility of the attacker. The second and the third terms work as the barriers to limit the sensory data of malicious workers to the range $(0, 1)$. Parameters δ_1 and δ_2 are used to adjust the trade-off between these three terms. Here we use the barriers instead of constraint $\{\widetilde{x}^{k'}_{mc}\}^{K',M,C}_{k',m,c=1} \in (0, 1)$ in the optimization problem to reduce the computation complexity. The optimal solution of the above problem is very close to that of the original optimization problem when parameters δ_1 and δ_2 are small, and θ is large. Constraint (17) can guarantee the summation of the elements in the probability vector (i.e., $\widetilde{x}^{k'}_{m}$) equals to 1.

Next, we discuss how to solve this optimization problem. Inspired by the dual ascent method [6], we first get the Lagrangian form of the upper-level problem:

$$L_1(\widetilde{X}, \Psi) = \sum_{m=1}^{M} \{1 - \frac{1}{1 + \exp[-\theta(x^*_{mc_2} - x^*_{mc_1})(\widehat{x}^*_{mc_2} - \widehat{x}^*_{mc_1})]}\} + \delta_1 \sum_{k'=1}^{K'} \sum_{m \in \widetilde{O}_{k'}} \sum_{c=1}^{C} \log \widetilde{x}^{k'}_{mc} + \delta_2 \sum_{k'=1}^{K'} \sum_{m \in \widetilde{O}_{k'}} \sum_{c=1}^{C} \log(1 - \widetilde{x}^{k'}_{mc}) + \sum_{k'=1}^{K'} \sum_{m \in \widetilde{O}_{k'}} \psi^{k'}_{m} (\sum_{c=1}^{C} \widetilde{x}^{k'}_{mc} - 1), \tag{18}$$

where $\Psi = \{\psi^{k'}_{m}\}^{M,K'}_{m,k'=1}$ are the Lagrangian multipliers. The solution we adopted here is a two-phase iterative procedure:

Phase I: In this phase, we first fix the Lagrange multipliers $\Psi = \{\psi^{k'}_{m}\}^{M,K'}_{m,k'=1}$, which are calculated in the previous iteration. Then we solve the following optimization problem:

$$\max_{\widetilde{X}} \ L_1(\widetilde{X}, \Psi)$$

$$\text{s.t.} \quad \{\widehat{X}^*, W, \widetilde{W}\} = \underset{\widehat{X}^*, W, \widetilde{W}}{\mathrm{argmin}} \ f(\widehat{X}^*, W, \widetilde{W})$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) + \sum_{k'=1}^{K'} \exp(-\widetilde{w}_{k'}) = 1. \tag{19}$$

The method used to solve optimization problem (19) is also a two-step iterative procedure. Here we call this procedure the *inner iterative procedure* in order to differentiate it from the two-phase iterative procedure mentioned above. The two steps of the inner iterative procedure are summarized as follows:

Step ①: We fix the malicious workers' observations $\widetilde{X}$, which are calculated in the previous iteration of the inner iterative procedure. Then we solve the lower-level problem to get the optimal solution $\{\widehat{X}^*, W, \widetilde{W}\}$, which is the truth discovery problem discussed in section 4.1.

Step ②: We adopt the gradient ascent method to solve the upper-level problem. More specifically, in iteration r of the inner iterative procedure, $\widetilde{x}_{mc_1}^{k'}$ and $\widetilde{x}_{mc_2}^{k'}$ in vector $\widetilde{x}_m^{k'}$ are updated as

$$\widetilde{x}_{mc}^{k'(r+1)} \leftarrow \widetilde{x}_{mc}^{k'(r)} + \alpha_r^1 \cdot \nabla_{\widetilde{x}_{mc}^{k'}} L_1(\widetilde{X}, \Psi), \quad c \in \{c_1, c_2\} \tag{20}$$

where α_r^1 is the step size in iteration r of the inner iterative procedure. For gradient $\nabla_{\widetilde{x}_{mc}^{k'}} L_1(\widetilde{X}, \Psi)$, it is calculated as

$$\begin{aligned}\nabla_{\widetilde{x}_{mc}^{k'}} L_1(\widetilde{X}, \Psi) = &\sum_{m'=1}^{M} \{\frac{\exp(\theta d_1 d_2)\theta d_1}{[1+\exp(\theta d_1 d_2)]^2} \cdot \frac{\partial \widehat{x}_{m'c_1}^*}{\partial \widetilde{x}_{mc}^{k'}}\} \\ &- \sum_{m'=1}^{M} \{\frac{\exp(\theta d_1 d_2)\theta d_1}{[1+\exp(\theta d_1 d_2)]^2} \cdot \frac{\partial \widehat{x}_{m'c_2}^*}{\partial \widetilde{x}_{mc}^{k'}}\} \\ &+ \frac{\delta_1}{\widetilde{x}_{mc}^{k'}} - \frac{\delta_2}{1-\widetilde{x}_{mc}^{k'}} + \psi_m^{k'},\end{aligned} \tag{21}$$

where $d_1 = x_{m'c_2}^* - x_{m'c_1}^*$ and $d_2 = \widehat{x}_{m'c_2}^* - \widehat{x}_{m'c_1}^*$. Here $\frac{\partial \widehat{x}_{m'c_1}^*}{\partial \widetilde{x}_{mc}^{k'}}$ and $\frac{\partial \widehat{x}_{m'c_2}^*}{\partial \widetilde{x}_{mc}^{k'}}$ are calculated based on Eqn. (7):

$$\frac{\partial \widehat{x}_{m'c_1}^*}{\partial \widetilde{x}_{mc}^{k'}} = \begin{cases} \frac{\widetilde{w}_{k'}}{\sum_{k\in U_m} w_k + \sum_{k'\in \widetilde{U}_m} \widetilde{w}_{k'}} & m = m' \text{ and } c = c_1 \\ 0 & \text{others.} \end{cases} \tag{22}$$

$$\frac{\partial \widehat{x}_{m'c_2}^*}{\partial \widetilde{x}_{mc}^{k'}} = \begin{cases} \frac{\widetilde{w}_{k'}}{\sum_{k\in U_m} w_k + \sum_{k'\in \widetilde{U}_m} \widetilde{w}_{k'}} & m = m' \text{ and } c = c_2 \\ 0 & \text{others.} \end{cases} \tag{23}$$

The reason why we only update $\widetilde{x}_{mc_1}^{k'}$ and $\widetilde{x}_{mc_2}^{k'}$ is that only the two answers (i.e., c_1 and c_2) with the highest and second highest probability in vector x_m^* are considered when we assign the observations for the malicious workers. The malicious workers who observe the m-th object should select one of the answers (i.e., c_1 or c_2) as his observation for this object in order to achieve the attack goal.

Step ① and step ② in the inner iterative procedure will be conducted until the convergence criterion is satisfied. Here the convergence criterion is that all the gradients $\{\nabla_{\widetilde{x}_{mc}^{k'}} L_1(\widetilde{X}, \Psi)\}_{m,k'}^{M,K'}$ are less than a threshold.

Phase II: We adopt the gradient descent method to update the Lagrangian multipliers $\Psi = \{\psi_m^{k'}\}_{m,k'=1}^{M,K'}$ based on $\widetilde{X}$ calculated in phase I. More specifically, in iteration t, $\psi_m^{k'}$ is updated as

$$\psi_m^{k'(t+1)} \leftarrow \psi_m^{k'(t)} - \alpha_t^2 \cdot (\sum_{c=1}^{C} \widetilde{x}_{mc}^{k'} - 1), \tag{24}$$

where α_t^2 is the step size in iteration t.

The above two phases will be iteratively conducted until the Lagrangian multipliers $\{\psi_m^{k'}\}_{m,k'=1}^{M,K'}$ converge. We can get malicious workers' observation vectors $\widetilde{X} = \{\widetilde{x}_m^{k'}\}_{m,k'=1}^{M,K'}$. Then the k'-th malicious worker selects the candidate answer with the largest value in vector $\widetilde{x}_m^{k'}$ as the final observation of the m-th object, and submit it to the cloud server. The procedure is summarized as Algorithm 2.

Algorithm 2: Optimizing $\widetilde{X}$ for the availability attack

Input: The number of objects: M; the number of normal workers: K; the normal workers' observations: X; the number of malicious workers: K'; the objects observed by the malicious workers: $\{\widetilde{O}_{k'}\}_{k'=1}^{K'}$.

Output: The optimal attack strategy $\widetilde{X}_{opt}$

```
Initialize the malicious workers' observations X̃ and the Lagrange multipliers Ψ;
X̃_opt ← ∅;
while Ψ = {ψ_m^{k'}}_{m,k'=1}^{M,K'} does not converge do
    while the gradients do not satisfy the convergence criterion do
        Calculate the optimal solution {X̂*, W, W̃} based on Eqn.(7), Eqn.(8) and Eqn.(9);
        Update X̃ based on Eqn.(20);
    end
    Update Ψ based on Eqn.(24);
end
for each x̃_m^{k'} ∈ X̃ do
    x̃_m^{k'(opt)} ← the candidate answer with the largest value in vector x̃_m^{k'};
    X̃_opt ← X̃_opt ∪ {x̃_m^{k'(opt)}};
end
return The optimal attack strategy X̃_opt;
```

4.3 Target Attack

In the target attack, the attacker tries to skew the estimated truths of some objects (called the *target objects*) to certain target answers through poisoning the sensory data. The target answers are usually predetermined by the attacker. When conducting the target attack, if the final truth discovery result on a target object is changed to the target answer after the attack, it means that the attack on this object succeeds. Otherwise, the attack on this object fails.

Given the limited capability of the attacker, in this section we discuss how to find the optimal attack strategy so that the attack can succeed on as many target objects as possible. Suppose that the attacker wants to attack $\overline{M}$ ($\overline{M} \le M$) objects among all the objects observed by the normal workers. The target answer and the final estimated answer for the $\overline{m}$-th target object after the attack are denoted as $\overline{x}_{\overline{m}}^{*f}$ and $\widehat{x}_{\overline{m}}^{*f}$ respectively. We can formulate the goal of the target attack into an optimization problem as follows:

$$\begin{aligned}\max_{\widetilde{X}} \quad & \sum_{\overline{m}=1}^{\overline{M}} \mathbb{1}(\widehat{x}_{\overline{m}}^{*f} = \overline{x}_{\overline{m}}^{*f}) \\ \text{s.t.} \quad & \{\widehat{X}^{*f}, W, \widetilde{W}\} = \underset{\widehat{X}^{*f}, W, \widetilde{W}}{\operatorname{argmin}} f(\widehat{X}^{*f}, W, \widetilde{W}) \\ & \text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) + \sum_{k'=1}^{K'} \exp(-\widetilde{w}_{k'}) = 1,\end{aligned} \tag{25}$$

where $\widehat{X}^{*f} = \{\widehat{x}_{\overline{m}}^{*f}\}_{\overline{m}=1}^{\overline{M}}$ are the final estimated object answers after the attack and $\{\widehat{x}_{\overline{m}}^{*f}\}_{\overline{m}=1}^{\overline{M}} \in \widehat{X}^{*f}$.

When conducting the target attack, the malicious workers need to pick one candidate answer for each target object. An intuitive attack strategy is to choose the target answer. However, this strategy

may not be the optimal choice under truth discovery framework. The reason is similar to that in the availability attack. The weights of malicious workers can be greatly decreased since they would disagree with the majority of the normal workers. A better strategy may sacrifice on some of the target objects where it is unlikely to skew the estimated truths, so that the weights of malicious workers can be increased, and eventually their impact on other objects can be improved. Here we assume that each malicious worker only observes the target objects. Since we only consider the change between the target answers and the answers with the highest probability values in the estimated truth vectors before the attack, we reformulate problem (25) as

$$\max_{\widetilde{X}} \sum_{\overline{m}=1}^{\overline{M}} \frac{1}{2}\{1 - \text{sgn}[(x^*_{\overline{m}c_1} - x^*_{\overline{m}c_T}) \cdot (\widehat{x}^*_{\overline{m}c_1} - \widehat{x}^*_{\overline{m}c_T})]\} \tag{26}$$

$$\text{s.t.} \quad \{\widehat{X}^*, W, \widetilde{W}\} = \underset{\widehat{X}^*, W, \widetilde{W}}{\text{argmin}} \; f(\widehat{X}^*, W, \widetilde{W})$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) + \sum_{k'=1}^{K'} \exp(-\widetilde{w}_{k'}) = 1$$

$$\{\widetilde{x}^{k'}_{\overline{m}c}\}^{K', \overline{M}, C}_{k', \overline{m}, c=1} \in \{0, 1\},$$

where $\widehat{X}^*$ is the set of probability vectors output by truth discovery algorithm. In the objective function (26), c_T and c_1 represent the target answer and the answer with the highest value in the probability vector calculated before the attack. This optimization problem reflects the idea that if the object truth does not switch from the answer with the highest probability value to the target answer after the attack, the attack fails on this target object.

Similar to the availability attack, in order to solve this optimization problem, we approximate the objective function (26) by:

$$\max_{\widetilde{X}} \sum_{\overline{m}=1}^{\overline{M}} \{1 - \frac{1}{1 + \exp[-\theta(x^*_{\overline{m}c_1} - x^*_{\overline{m}c_T})(\widehat{x}^*_{\overline{m}c_1} - \widehat{x}^*_{\overline{m}c_T})]}\}, \tag{27}$$

where θ denotes the steepness of the curve. Then we can formulate the following optimization problem to achieve the attacker's goal.

$$\max_{\widetilde{X}} \; h(\widetilde{X}) = \sum_{\overline{m}=1}^{\overline{M}} \{1 - \frac{1}{1 + \exp[-\theta(x^*_{\overline{m}c_1} - x^*_{\overline{m}c_T})(\widehat{x}^*_{\overline{m}c_1} - \widehat{x}^*_{\overline{m}c_T})]}\}$$

$$+ \delta_1 \sum_{k'=1}^{K'} \sum_{\overline{m}=1}^{\overline{M}} \sum_{c=1}^{C} \log \widetilde{x}^{k'}_{\overline{m}c} + \delta_2 \sum_{k'=1}^{K'} \sum_{\overline{m}=1}^{\overline{M}} \sum_{c=1}^{C} \log(1 - \widetilde{x}^{k'}_{\overline{m}c}) \tag{28}$$

$$\text{s.t.} \quad \{\widehat{X}^*, W, \widetilde{W}\} = \underset{\widehat{X}^*, W, \widetilde{W}}{\text{argmin}} \; f(\widehat{X}^*, W, \widetilde{W})$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) + \sum_{k'=1}^{K'} \exp(-\widetilde{w}_{k'}) = 1$$

$$\sum_{c=1}^{C} \widetilde{x}^{k'}_{\overline{m}c} = 1, \text{ where } k' = 1, \ldots, K' \text{ and } \overline{m} = 1, \ldots, \overline{M}.$$

Similar to the optimization problem formulated in the availability attack, this problem is a bi-level optimization problem and the objective function $h(\widetilde{X})$ contains three terms: The first term represents the utility of the attacker. The second and the third terms are the barriers used to limit each element in the malicious workers' observation vectors to the range $(0, 1)$. The solution for this optimization problem is also a two-phase iterative procedure which is similar to that for the availability attack.

5 EXPERIMENTS ON THE CROWD WISDOM SYSTEM

We build a crowd wisdom system to evaluate the performance of the proposed attack framework. In this system, the cloud server publishes some multi-choice trivia questions using the Android App we developed, and the workers can view and submit their answers using the App. After receiving the answers from the workers, the cloud server applies the truth discovery approach (i.e., the CRH framework) to infer the true answer for each question. The attack occurs after the data of the normal workers are submitted to the cloud server but before the truth discovery procedure starts. In our experiment, 30 smartphone users are employed as the normal workers and 19 questions are used as the objects. Each question has 4 candidate answers and the users can only choose one answer for each question. The participants are not required to answer all questions. Instead, they can choose any questions as they will.

5.1 Availability Attack

In the availability attack, the attacker tries to maximally disturb the truth discovery results. In this experiment, a fixed number of malicious workers can be created, and each of them can observe a randomly selected subset of objects. We compare the proposed availability attack framework with the following attack strategy.

Baseline. The attacker first runs the truth discovery algorithm (CRH) on the observations provided by the normal workers. Then the attacker sets each malicious worker's observation on a given object as the candidate answer which has the second highest probability value based on the truth discovery result on this object. For example, for one question, CRH outputs the aggregation result as $(0.6, 0.1, 0.2, 0.1)$. Then the malicious workers who are assigned to this question will provide observations as $(0, 0, 1, 0)$. This baseline method is intuitive since this candidate answer is more likely to win over the estimated object truth before the attack than other candidate answers. In fact, it is the optimal attack strategy if the aggregation method is voting (that is, for each object, the candidate answer which has the highest vote counts is the aggregation result).

For the proposed attack framework, the optimal observations for each malicious worker are calculated according to Algorithm 2. We set $\theta = 100$ and initialize the observations of malicious workers on an object as the truth discovery results from the normal workers' observations on that object. In order to evaluate the performance of the availability attack strategies, we adopt two metrics: the *utility* defined in Eqn. (14), and the *change rate*. For the latter, it is defined as the percentage of the objects which has different final aggregation results before and after the attack. It is equivalent to the utility defined in Eqn. (10). All the experiments are conducted 20 times and we report the average results.

5.1.1 The Effect of the Percentage of Malicious Workers. Here we assume that each malicious worker can observe 10 randomly selected objects. Then we vary the percentage of malicious workers from 0.03 to 0.3 and calculate the attacker's utility and the change rate. The results are shown in Figure 2, from which we can see that the proposed optimal attack framework outperforms the baseline method in all cases. This figure also shows that the advantage of the proposed attack framework is marginal when the percentage of malicious workers is 0.03. This is because the number of malicious

workers in this case is too small, and it is hard to change the aggregation results much. However, the advantage of the proposed attack framework becomes bigger when the percentage of malicious workers gradually increases. To change the aggregation results on 20% of the objects, the proposed attack framework only needs less than 12% of malicious workers whereas the baseline method needs about 21% of malicious workers. For the proposed attack framework, the increment of change rate slows down after the malicious workers occupy 15% of the total workers, but the utility keeps increasing steadily. The reason is that the change of an estimated object truth is either 0 (not changed) or 1 (changed), while the utility is a continuous value. For example, the estimated object truth vector that changes from $(0.7, 0.2, 0.1, 0)$ to $(0.5, 0.4, 0.1, 0)$ does not increase the change rate, but increases the utility.

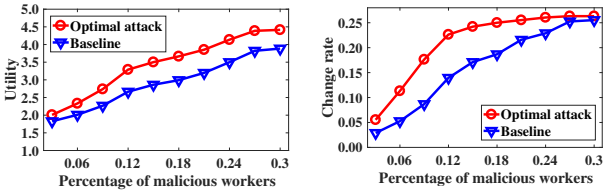

Figure 2: Utility and Change rate w.r.t. the percentage of malicious workers for availability attack.

5.1.2 The Effect of the Number of the Observed Objects. With the fixed number of malicious workers, if one malicious worker can observe more objects, he can make impact to more objects, and thus achieve higher impact to the overall sensing system. In this experiment, we examine the effect of the number of the objects a malicious worker can observe. Here we fix the percentage of malicious workers to be 10%. Then we vary the number of objects that each malicious worker can observe from 2 to 18. The results are reported in Figure 3. The results clearly demonstrate the advantage of the proposed attack framework over the baseline method. With the increment of the observed objects, the malicious workers exert more and more impact on the sensing system, and the advantage of the proposed attack framework over the baseline method also increases. Figure 3 shows that to achieve 20% change rate, the malicious workers of the proposed attack framework only need to observe on 10 objects, while the malicious workers of the baseline method needs to observe on 18 objects.

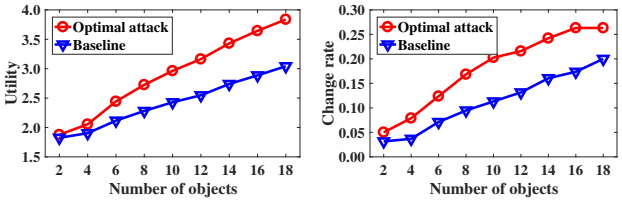

Figure 3: Utility and Change rate w.r.t. the number of the objects which are observed by each malicious worker

5.1.3 Comparison on Weights of the Malicious Workers. The reason that the proposed attack framework outperforms the baseline method lies in the fact that the effect of worker reliability estimation in the truth discovery algorithm is considered. The proposed attack framework will let the malicious worker "fake" like a normal worker or even a good worker on some objects to enhance its weight. Whereas for the baseline method, the malicious workers always disagree with the majority of the normal workers, and thus the suspicious behavior may be detected by the truth discovery algorithm and cause the decrease in the weights.

In this experiment, we examine the weight distributions for both the normal workers and the malicious workers. We choose the following two settings: the percentage of malicious workers is set as 0.15 (i.e., 6 malicious workers), and we let them observe 5 objects and 15 objects respectively. In Figure 4, we plot the weights for all workers after the proposed attack framework attacks the sensing system, and the weights for all workers after the baseline method attacks the sensing system for the aforementioned two scenarios. From Figures 4a and 4c, we can see that the malicious workers from the proposed attack framework all have high weights comparing with the normal workers. This means that the malicious workers successfully blend into the normal workers. Therefore, it is hard for the truth discovery algorithm to detect the attack. In contrast, the malicious workers from the baseline method all have very low weights comparing with the normal workers, as shown in Figures 4b and 4d. The two figures confirm our expectations that the truth discovery algorithm finds these malicious workers since they behave differently from the normal workers. The low worker weights not only limit the impact of the malicious workers, but also make them vulnerable to straightforward defense mechanism.

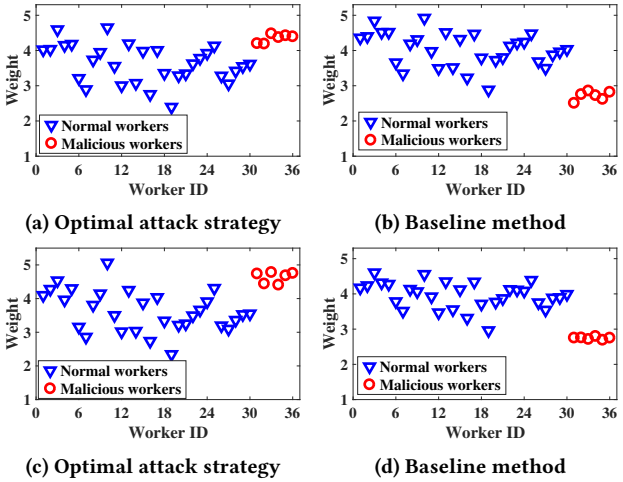

Figure 4: The weight of each worker for availability attack. (a) and (b) show the worker weights when each malicious worker observe 5 objects. (c) and (d) show the worker weights when each malicious worker observe 15 objects.

5.2 Target Attack

In the target attack, the attacker tries to skew the truth discovery results to the target values on certain objects. To make the problem more interesting, we assume that the target values are not the same as the values that have the highest probabilities derived by the truth discovery algorithm before attack; and to make the target attack different from the availability attack, we further assume that not all the target values are the same as the values with the second highest probabilities. We compare the proposed target attack framework with the following attack strategy.

Baseline. For the target objects, the attacker sets the malicious workers' observations as the target choices.

For the proposed attack framework, we set $\theta = 100$ and initialize the observations of malicious workers on an object as the truth discovery results from the normal workers on that object. In order to evaluate the performance of the target attack strategies, we adopt two metrics: the *utility* defined in Eqn. (27), and the *change rate*. For the latter, it is defined as the number of objects that are successfully changed to the target value divided by the total number of target objects. It is equivalent to the utility defined in Eqn. (25).

5.2.1 The Effect of the Percentage of Malicious Workers and the Number of the Target Objects. In this experiment, we examine how the percentage of malicious workers and the number of the target objects affect the attack results. We vary the percentage of malicious workers from 0.03 to 0.27 with 10 and 15 target objects. The results are plotted in Figure 5. We can still observe that the proposed attack framework outperforms the baseline method in all cases. The proposed attack framework can usually use one or two fewer malicious workers to achieve the same change rate comparing with the baseline method. The effect of the percentage of malicious workers in the target attack is similar to that of in the availability attack: the more malicious workers, the higher the utility and the change rate. Increasing the number of the target objects, however, makes the attack goal harder to achieve. This is because that under our problem settings, the target attack is significantly more difficult than the availability attack, since the target values may be supported by much fewer normal workers. Therefore, when there are more target objects, the attacker needs to add more malicious workers to achieve the same change rate.

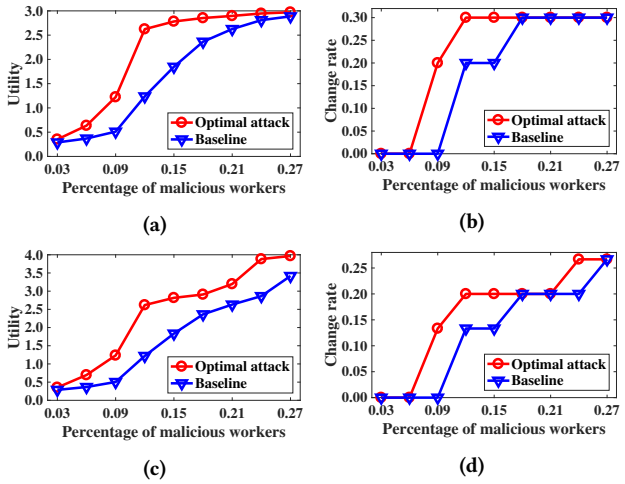

Figure 5: Utility and Change rate w.r.t. the percentage of malicious workers for target attack. (a) and (b) show the results when 10 objects are attacked. (c) and (d) show the results when 15 objects are attacked.

5.2.2 Comparison on Weights of the Malicious Workers. Next, we explore the worker weight distributions in the target attack. The following two settings are compared: the percentage of malicious workers is set as 0.15 (i.e., 6 malicious workers), and we let the number of target objects to be 10 and 15 respectively. In Figure 6, we plot the weights for all workers after the proposed attack framework attacks the sensing system, and the weights for all workers after the baseline method attacks the sensing system. From Figures 6a and 6c, we can see that the weights of the malicious workers from the proposed attack framework are similar to the weights of the normal workers, so they again successfully blend into the normal workers. The reason is that for some target objects where the target values are too hard to achieve, the malicious workers may disguise their purpose by agreeing with the normal workers. In contrast, the malicious workers from the baseline method all have very low weights (Figures 6b and 6d), as they always choose the choices that are not supported by the normal workers. The results suggest that the malicious workers from the baseline method are more vulnerable to straightforward defense mechanisms.

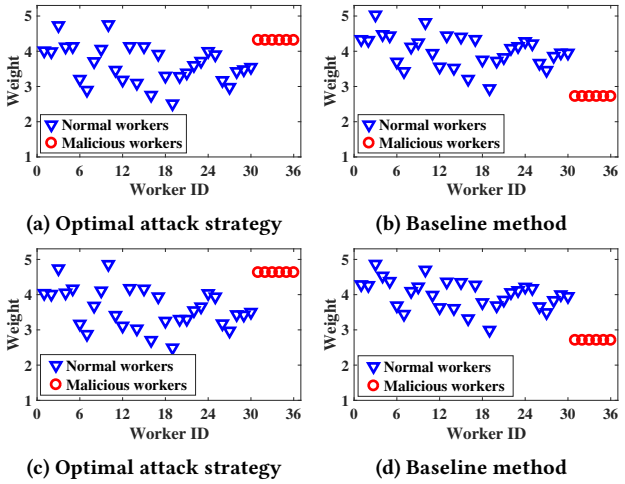

Figure 6: The weight of each worker for target attack. (a) and (b) show the results when 10 objects are attacked. (c) and (d) show the results when 15 objects are attacked.

6 RELATED WORK

As an advanced approach to discover truthful information from unreliable data, truth discovery has recently drawn much more attention [21–24, 33, 36, 37, 42]. Compared with the traditional data aggregation methods (e.g., majority voting) in crowd sensing systems, truth discovery can provide more reliable aggregation results by inferring the workers' reliability based on the data from all workers. Although different truth discovery schemes have been developed to tackle different scenarios, these schemes do not take data poisoning attacks into consideration.

The data poisoning attacks, also known as false data injection attacks, have recently been widely studied in crowd sensing and crowd sourcing applications [7–10, 15, 16, 20, 30, 31, 34, 35, 38, 45]. The data poisoning attacks and related defense schemes are also studied in the applications other than crowd sensing and crowd sourcing, such as Internet of Things [14, 32, 43], electric power grids [26] and network coding [18]. Besides, there also has been prior research exploring the data poisoning attacks on machine learning algorithms [1, 3, 5, 13, 19, 39]. However, these previous works do not investigate how to effectively attack the crowd sensing systems empowered with truth discovery mechanism, which could tolerate the malicious workers to some degree and is hard to be attacked. Although a similar attack framework is proposed for crowd sourcing scenarios in [28], the attacked algorithm discussed in this work is different from the truth discovery algorithm we considered

in this paper. Additionally, the work in [28] mainly focuses on the availability attack and assume that the crowdsourcing answers are binary. However, in this paper we investigate a more general case where the attacker can conduct both the availability attack and the target attack, and the sensory data can be multi-class.

7 CONCLUSIONS

In this paper, we study two types of data poisoning attacks, i.e., the availability attack and the target attack, against a crowd sensing system empowered with the truth discovery mechanism. We first analyze the pitfalls when attacking such a crowd sensing system and then design an optimal attack framework to derive the (approximately) optimal attack strategy. Through manipulating the malicious workers' sensory data based on the derived attack strategy, the attacker can not only maximize his attack utility but also successfully disguise the attack behaviors. The proposed optimal attack framework is tested on a real-world crowd sensing system. The experimental results demonstrate that compared with the native baseline schemes, the proposed attack framework can achieve higher attack utility and at the same time, let the malicious workers gain higher reliability degrees such that they cannot be detected easily.

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under grants CNS-1652503 and CNS-1737590.

REFERENCES

[1] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data Poisoning Attacks against Autoregressive Models.. In *AAAI, 2016.*
[2] Jonathan F Bard. *Practical bilevel optimization: algorithms and applications.* Kluwer Academic Publishers, 1998.
[3] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure?. In *ASIACCS, 2006.*
[4] Dimitri P Bertsekas. *Nonlinear programming.* Athena scientific Belmont, 1999.
[5] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *ICML, 2012.*
[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. In *Foundations and Trends® in Machine Learning, 2011.*
[7] Shih-Hao Chang and Zhi-Rong Chen. Protecting Mobile Crowd Sensing against Sybil Attacks Using Cloud Based Trust Management System. In *Mobile Information Systems, 2016.*
[8] Akshay Dua, Nirupama Bulusu, Wu-Chang Feng, and Wen Hu. Combating software and Sybil attacks to data integrity in crowd-sourced embedded systems. In *TECS, 2014.*
[9] Lingjie Duan, Alexander W Min, Jianwei Huang, and Kang G Shin. Attack prevention for collaborative spectrum sensing in cognitive radio networks. In *JSAC, 2012.*
[10] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *CHI, 2015.*
[11] Lin Gao, Fen Hou, and Jianwei Huang. Providing long-term participation incentive in participatory sensing. In *INFOCOM, 2015.*
[12] Shaohan Hu, Lu Su, Hengchang Liu, Hongyan Wang, and Tarek F Abdelzaher. Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification. In *TOSN, 2015.*
[13] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. Adversarial machine learning. In *AISec, 2011.*
[14] Vittorio P Illiano and Emil C Lupu. Detecting malicious data injections in wireless sensor networks: A survey. In *ACM Computing Surveys, 2015.*
[15] Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Identifying Unreliable and Adversarial Workers in Crowdsourced Labeling Tasks. In *Journal of Machine Learning Research, 2016.*
[16] Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Reputation-based worker filtering in crowdsourcing. In *NIPS, 2014.*
[17] Changkun Jiang, Lin Gao, Lingjie Duan, and Jianwei Huang. Scalable Mobile Crowdsensing via Peer-to-Peer Data Sharing. In *arXiv preprint arXiv:1705.05343, 2017.*
[18] Anh Le and Athina Markopoulou. Cooperative defense against pollution attacks in network coding using spacemac. In *JSAC, 2012.*
[19] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data Poisoning Attacks on Factorization-Based Collaborative Filtering. In *NIPS, 2016.*
[20] Hongjuan Li, Xiuzhen Cheng, Keqiu Li, Chunqiang Hu, Nan Zhang, and Weilian Xue. Robust collaborative spectrum sensing schemes for cognitive radio networks. In *TPDS, 2014.*
[21] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. A confidence-aware approach for truth discovery on long-tail data. In *VLDB, 2014.*
[22] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *SIGMOD, 2014.*
[23] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. A survey on truth discovery. In *Sigkdd Explorations Newsletter, 2016.*
[24] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. Conflicts to Harmony: A Framework for Resolving Conflicts in Heterogeneous Data by Truth Discovery. In *TKDE, 2016.*
[25] Shengzhong Liu, Zhenzhe Zheng, Fan Wu, Shaojie Tang, and Guihai Chen. Context-aware data quality estimation in mobile crowdsensing. In *INFOCOM, 2017.*
[26] Yao Liu, Peng Ning, and Michael K Reiter. False data injection attacks against state estimation in electric power grids. In *TISSEC, 2011.*
[27] Chenglin Miao, Wenjun Jiang, Lu Su, Yaliang Li, Suxin Guo, Zhan Qin, Houping Xiao, Jing Gao, and Kui Ren. Cloud-enabled privacy-preserving truth discovery in crowd sensing systems. In *SenSys, 2015.*
[28] Chenglin Miao, Qi Li, Lu Su, Mengdi Huai, Wenjun Jiang, and Jing Gao. Attack under Disguise: An Intelligent Data Poisoning Attack Mechanism in Crowdsourcing. In *WWW, 2018.*
[29] Dan Peng, Fan Wu, and Guihai Chen. Data Quality Guided Incentive Mechanism Design for Crowdsensing. In *TMC, 2017.*
[30] Zhengrui Qin, Qun Li, and George Hsieh. Defending against cooperative attacks in cooperative spectrum sensing. In *TWC, 2013.*
[31] Ankit Singh Rawat, Priyank Anand, Hao Chen, and Pramod K Varshney. Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks. In *TSP, 2011.*
[32] Mohsen Rezvani, Aleksandar Ignjatovic, Elisa Bertino, and Sanjay Jha. Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. In *TDSC, 2015.*
[33] Lu Su, Qi Li, Shaohan Hu, Shiguang Wang, Jing Gao, Hengchang Liu, Tarek F Abdelzaher, Jiawei Han, Xue Liu, Yan Gao, et al. Generalized decision aggregation in distributed sensing systems. In *RTSS, 2014.*
[34] Gang Wang, Bolun Wang, Tianyi Wang, Ana Nika, Haitao Zheng, and Ben Y Zhao. Defending against Sybil devices in crowdsourced mapping services. In *MobiSys, 2016.*
[35] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y Zhao. Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers.. In *Usenix Security, 2014.*
[36] Shiguang Wang, Lu Su, Shen Li, Shaohan Hu, Tanvir Amin, Hongwei Wang, Shuochao Yao, Lance Kaplan, and Tarek Abdelzaher. Scalable social sensing of interdependent phenomena. In *IPSN, 2015.*
[37] Shiguang Wang, Dong Wang, Lu Su, Lance Kaplan, and Tarek F Abdelzaher. Towards cyber-physical systems in social spaces: The data reliability challenge. In *RTSS, 2014.*
[38] Wei Wang, Lin Chen, Kang G Shin, and Lingjie Duan. Thwarting intelligent malicious behaviors in cooperative spectrum sensing. In *TMC, 2015.*
[39] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning?. In *ICML, 2015.*
[40] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *MobiCom, 2012.*
[41] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones. In *TON, 2016.*
[42] Shuo Yang, Fan Wu, Shaojie Tang, Xiaofeng Gao, Bo Yang, and Guihai Chen. On Designing Data Quality-Aware Truth Estimation and Surplus Sharing Method for Mobile Crowdsensing. In *JASC, 2017.*
[43] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. Sybil attacks and their defenses in the internet of things. In *IEEE Internet of Things Journal, 2014.*
[44] Mengyuan Zhang, Lei Yang, Xiaowen Gong, and Junshan Zhang. Privacy-Preserving Crowdsensing: Privacy Valuation, Network Effect, and Profit Maximization. In *GLOBECOM, 2016.*
[45] Rui Zhang, Jinxue Zhang, Yanchao Zhang, and Chi Zhang. Secure crowdsourcing-based cooperative pectrum sensing. In *INFOCOM, 2013.*