Crowdsourcing under Data Poisoning Attacks: A Comparative Study

Farnaz Tahmasebian, Li Xiong, Mani Sotoodeh, and Vaidy Sunderam

Emory University, Atlanta GA, USA
{ftahmas, lxiong, msotood, vss}@emory.edu

Abstract. Crowdsourcing is a paradigm that provides a cost-effective solution for obtaining services or data from a large group of users. It is increasingly being used in modern society for data collection in domains such as image annotation or real-time traffic reports. A key component of these crowdsourcing applications is *truth inference* which aims to derive the true answer for a given task from the user-contributed data, e.g. the existence of objects in an image, or true traffic condition of a road. In addition to the variable quality of the contributed data, a potential challenge presented to crowdsourcing applications is *data poisoning attacks* where malicious users may intentionally and strategically report incorrect information in order to mislead the system to infer the wrong truth for all or a targeted set of tasks. In this paper, we propose a comprehensive data poisoning attack taxonomy for truth inference in crowdsourcing and systematically evaluate the state-of-the-art truth inference methods under various data poisoning attacks. We use several evaluation metrics to analyze the robustness or susceptibility of different methods against various attacks, which sheds light on the resilience of existing methods and ultimately helps in building more robust truth inference methods in an open setting.

Keywords: Truth Inference · Data Poisoning · Crowdsourcing

1 Introduction

Crowdsourcing is a paradigm in which organizations or individuals obtain data or service from a large or relatively open group of users, or *crowd*. It has been increasingly used in modern society for data collection in various domains such as image annotation or real-time traffic reports. Amazon Mechanical Turk (MTurk) [6, 20] is one of the most pervasive crowdsourcing marketplaces, in which requesters submit various tasks requiring human intelligence, such as labeling objects in an image or flagging inappropriate content. Another example is Waze [46], a navigation and traffic sharing application. Users can report the traffic status at various locations using the app which is then aggregated to update the traffic condition shown on the map.

A key component of crowdsourcing applications is *truth inference* which aims to derive the answers for the tasks, e.g. the objects in the image, or true traffic condition of the road, by aggregating the user-provided data. Truth inference [12,

24,29,36,38] is a challenging task due to the open nature of the crowd. First, the number of available ratings per task varies significantly. Second, the reliability of the workers can vary. For example, in the Waze application, it is common for some users to not care to report at all or to carelessly report the traffic condition. So, estimating the level of trust one has in workers' responses and ultimately inferring the correct label for the tasks by aggregating their responses becomes complicated. Finally, the crowdsourcing applications may be subject to *data poisoning attacks* [23, 40] where malicious users purposely and strategically report incorrect responses to mislead the system to infer the wrong label for all or a targeted set of tasks. In the Waze example, attackers might want to take the road with the least traffic by deceiving Waze application to wrongly indicate there is heavy traffic on that specific road. This can be achieved via Sybil attacks [10, 14, 49, 53] where an attacker creates a large number of Sybil workers to strategically report wrong answers.

Traditional Sybil detection in online social networks [1, 45] typically relies on additional features or metadata (e.g., connectivity graph and IP addresses). Recent works have proposed Sybil detection methods for crowdsourcing through defining golden questions and clustering workers [10, 49]. In this paper, we focus on the truth inference methods that only rely on workers' answers, and their robustness against poisoning attacks, which are orthogonal and complementary to Sybil detection methods using additional metadata.

The simplest method in truth inference is *majority voting*, where the truth will be the one chosen by the majority of the assigned workers. Since the reliability of workers is not considered, majority voting may fail in the presence of unreliable or malicious workers. Considerable research has been done on improving the accuracy of truth inference methods, including optimization-based methods [22, 28], probabilistic graphical model based methods [11, 25, 33, 44, 56], and neural network based methods [15, 17, 50]. These methods construct models that either explicitly or implicitly consider the credibility of workers, which create some form of defense against unreliable or malicious workers. Zheng et al. [56] evaluated truth inference methods with various worker models, types of tasks, and task models. The evaluation is focused on "normal" settings where workers may have varying reliability, but do not intentionally or strategically manipulate the answers. They concluded that truth inference methods that model worker behavior based on confusion matrix and utilize a probabilistic graphical model (PGM) have the best performances in most settings.

One unanswered question is how robust these inference methods are under intentional and strategic data poisoning attacks that are beyond normal worker behaviors. Adversaries may disguise themselves as normal workers by providing reliable answers for certain tasks to escape the worker's reliability model while providing the wrong answer for other targeted tasks. In the worst case, when adversaries know the truth inference method and other workers' answers, they may optimize theirs to maximizes the error of the truth inference method for all tasks or a subset of tasks [30, 31]. Thus, it is important to understand the various types of data poisoning attacks and evaluate how different truth inference

methods behave under such attacks to ultimately build robust truth inference methods.

Contributions. We propose a comprehensive data poisoning attack taxonomy for truth inference in crowdsourcing and systematically evaluate the state-of-the-art truth inference methods under various attacks. In summary:

- We present a comprehensive data poisoning attack taxonomy in crowdsourcing. We analyze the attacks along different dimensions, including attack goal (targeted vs untargeted), adversarial knowledge (black-box vs white-box), and attack strategy (heuristic vs optimization based). We also discuss the similarity and differences between data poisoning attacks in crowdsourcing and those in machine learning and other domains.
- We design heuristic and optimization based attacks that can be used on various truth inference methods as part of our evaluation methodology. The heuristic-based attacks assume black-box or no adversarial knowledge and model the worker behavior using a confusion matrix [11] and an additional disguise parameter to hide their malicious behavior. The optimization based attacks assume white-box or full adversarial knowledge including the truth inference methods being used and other workers' answers, and are adapted from existing optimization based attacks [30] while making them more generic so they are applicable to broader types of truth inference methods.
- We systematically evaluate the state-of-the-art truth inference methods under both heuristics and optimization based data poisoning attacks. The truth inference methods are selected carefully to represent the different types of methods including majority voting based [22], optimization based [28], probabilistic graphical model based [11, 24, 25], and neural network based [50]. They also portray different worker's behavior models including probability based [28], confusion matrix based [11,24,25], and implicit models [22,28]. Our study includes not only the best performing methods from the experimental study [56], but also additional direct computation [22] and optimization based methods [22, 28] and more recent neural network based methods [50].
- We propose several metrics to evaluate the robustness of truth inference methods against data poisoning attacks. We experiment on synthetic and real-world datasets, analyze the results over parameters such as percentage of malicious workers, different attack parameters, and sparsity of the crowdsourcing dataset. We summarize the experiment findings and draw conclusions on the robustness, strengths, and weaknesses of the methods. It is our belief that these results help understand the resilience of existing methods and ultimately build more robust truth inference methods against data poisoning attacks.

Section 2 summarizes the existing attacks in related domains. Section 3 formally defines the truth inference problem and presents a categorization of existing truth inference methods and the selected methods for evaluation. Section 4 presents the attack taxonomy. Section 5 describes our evaluation methodology. Section 6 presents the results with discussions and Section 7 concludes the paper with key findings and future work.

2 Related Work

Truth inference methods have been studied extensively. We provide a brief description of the methods included in our evaluation in Section 3. Here we briefly review the works in data poisoning attacks.

Data poisoning attacks for machine learning (ML) algorithms have been increasingly studied recently [2,5,18,23,37,40,47]. However, data poisoning attacks in ML and crowdsourcing differ in three ways: (1) attacks in ML deal with supervised models and the goal is to degrade the performance of the model on a validation dataset, but crowdsourcing is an unsupervised problem, (2) to carry out the attacks in ML, a certain percentage of records are poisoned, while all the features associated with the poisoned record (e.g. an image) can be altered, but in crowdsourcing, a fraction of workers may be malicious, and (3) ML problems typically have rich features for records while in crowdsourcing for each task only some ratings from workers are available. Hence crowdsourcing systems are more susceptible to data poisoning attacks due to its open and unsupervised nature and lack of rich features for the truth inference problem.

Shilling attack [7, 9, 16, 32, 34, 55] is another type of data poisoning attacks in recommender systems where intruders attempt to simulate the behavior of a subset of users, leading to their dissatisfaction in a recommender system, and disruption in other users' activities [16, 34]. The main difference between recommender systems and truth discovery is the subjectivity of the true labels of tasks in recommender systems, i.e. users' personal opinions, while ground truth for truth discovery is universal.

Other related attacks include spammer [13, 19, 20, 35, 48] and sybil [8, 26, 39, 43, 46, 51–54] attacks. In a spammer attack, workers (bots) randomly submit answers to tasks [13, 19, 20]. In sybil attacks, infiltrators create fake identities to affect the performance of the system [53,54]. The attack and defense methods, for example in social media and IoT, typically utilize metadata such as connectivity of graph and relationship between nodes, and IP address. Sybil and spammer attacks mainly focus on the system infiltration part of the attack. The data poisoning attacks considered here assume adversaries have already successfully created or compromised multiple workers and can inject strategic answers.

3 Truth Inference Methods

Truth Inference Problem Definition. We consider a crowdsourcing system comprised of some tasks and a pool of workers. Each task is assigned to a subset of workers. The goal in truth inference is to determine the true answer based on available answers for each task. Tasks in crowdsourcing can be categorized as 1) *decision-making tasks* where workers select a binary answer (e.g. yes or no), 2) *single label tasks* where workers select a single label among multiple candidate labels, and 3) *numeric tasks* where an answer is a number. Moreover, truth inference methods may consider factors such as type of tasks, tasks' difficulty, and task assignment methods [56]. In this paper, we focus on the robustness of truth inference methods under data poisoning attacks for decision-making tasks, i.e. the binary truth inference problem, and do not consider other variations largely

orthogonal to our evaluation. Examples of decision-making tasks are labeling the sentiment of sentences as positive or negative or reporting accidents on road. We also assume tasks are equally hard for the average worker and tasks are randomly assigned to a subset of workers following the power-law distribution [41].

Definition 1. *(Truth Inference [36]) Consider a set of M workers $\boldsymbol{W} = \{w_1, .., w_M\}$ and a set of N tasks $\boldsymbol{T} = \{t_1, t_2, ..., t_N\}$ where each task t_j has a truth label $z^*_{t_j} \in \boldsymbol{L} = \{0, 1\}$. Given an answer matrix $\boldsymbol{C}$ where each element $c^{w_i}_{t_j}$ indicates the answer from worker w_i for task t_j, the goal of truth inference is to infer the truth label $\hat{z}_{t_j} \in \boldsymbol{L}$ for each task $t_j \in \boldsymbol{T}$.*

Symbol	Description
N	Number of tasks
M	Number of workers
$\mathbf{T}$	Set of tasks
$\mathbf{W}$,$\mathbf{W}'$	Set of normal, malicious workers
w_i	i-th worker
t_j	j-th task
$\mathbf{T}_{w_i}$	Set of tasks assigned to i-th worker
π^{w_i}	Confusion matrix of i-th worker
$\mathbf{C}$	Answer matrix by all workers
$\mathbf{Z}^*$	Ground truth vector
$\hat{\mathbf{Z}}$	Predicted truth vector

Table 1: Summary of Notations

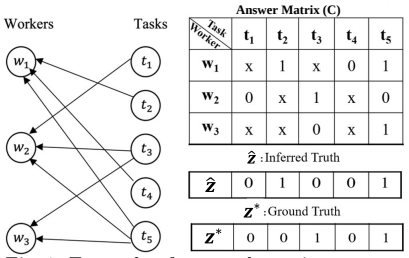


Fig. 1: Example of a crowdsourcing system

Figure 1 illustrates a crowdsourcing system with 3 workers and 5 tasks and a bipartite task-worker assignment graph. The input of a truth inference method is an answer matrix $\mathbf{C}$ provided by the workers. Workers label each task as 0 or 1, while x reflects that the task is not assigned to that worker. The output is $\hat{\mathbf{Z}}$ reflecting the inferred answer for the task. The ground truth vector $\mathbf{Z}^*$ is shown as a reference. Table 1 summarizes notations used throughout the paper.

Truth Inference Methods. There are four main categories of truth inference techniques: 1) direct computation, 2) optimization, 3) probabilistic graphical model (PGM), and 4) neural networks. Direct computation aggregates workers' answers by majority voting while treating workers equally or heuristically assigning weights to them [22]. Optimization based methods [22, 27, 28, 57, 59] treat the estimated labels and worker reliability as unknowns and use an optimization approach to find them. Probabilistic graphical models (PGM) explicitly model workers' reliability to estimate the labels [11, 12, 24, 25, 44, 48]. Optimization and PGM based methods follow an iterative Expectation Maximization (EM)-based approach consisting of: 1) inferring the label of tasks given the estimated workers' reliability, and 2) computing workers' reliability given the current inferred labels of the tasks. Recently, unsupervised neural network based approaches [15, 50] have been proposed that input answers of each task in a neural network outputting the inferred label of the task. Other approaches based on tensor augmentation and completion with limited performance have also been suggested [58].

Worker Models. Some truth inference methods do not have an explicit worker model while others model workers by: 1) a single *worker reliability* or *penalty* parameter reflecting their trustworthiness [12,48], 2) a *confusion matrix* capturing workers' probability of providing a certain label given the true label [11, 25, 44].

The confusion matrix π^{w_i} is an $|L| * |L|$ matrix where element $\pi^{w_i}_{p,q}$ denotes the probability of worker w_i reporting label q given the true label p. Assuming a binary label set $\mathbf{L} = \{0, 1\}$, the matrix is reduced to two variables, α_i and β_i, with $\alpha_i = \text{pr}(c^{w_i}_{t_j} = 1 \mid z^* = 1)$ and $\beta_i = \text{pr}(c^{w_i}_{t_j} = 0 \mid z^* = 0)$, showing the probability of worker w_i correctly reporting a task with true label 1 or 0 respectively.

Selected Methods. We aim to be comprehensive in adopting applicable categories of techniques and worker models to investigate their role in the robustness of inference methods under data poisoning attacks. When possible (e.g. for optimization and PGM-based techniques), we leverage the findings of the previous study [56] by selecting the best-performing methods.

For direct computation, MV (majority voting) and its enhanced version, MV-Soft [22] are chosen. For optimization based methods, we include MV-hard [22] which employs a semi-matching optimization for the worker-task bipartite graph and PM [28] which poses an optimization problem on inferred labels and workers' reliability. For PGM based methods, we chose the best-performing D&S [11] and BCC [25] modeling worker reliability by confusion matrix and KOS [24] using a single worker reliability parameter. For neural network based methods, we chose LAA-S [50], the only one applicable to non-complete bipartite graphs. We refer readers to the appendix for a more detailed discussion regarding each method.

4 Data Poisoning Attacks

Due to their open nature, crowdsourcing systems are subject to *data poisoning attacks* [23, 40] where malicious workers intentionally and strategically report incorrect labels to mislead the system to infer the wrong answer for all or a targeted set of tasks. This is different from unreliable behavior that is typically non-malicious, unintentional, and non-strategic. We propose a taxonomy for data poisoning attacks in crowdsourcing and review some existing attacks.

4.1 Attack Taxonomy

Attack Goal. Assuming a certain percentage of malicious workers, the answer matrix $\mathbf{C}$ contains corrupt answers $\mathbf{C}'$ of adversaries inside. There are two cases:

Untargeted Attack. Adversaries aim to decrease the overall accuracy. Their goal is to mislead the system to infer the wrong label for as many tasks as possible which can be formulated as: $\max_{\mathbf{C}'} \sum_{j=1}^{M} 1(\hat{z_{t_j}} \neq z^*_{t_j})$.

Targeted Attack. Adversaries aim to reduce the accuracy on only a targeted subset of tasks $\mathbf{T}_{tar} \subseteq \mathbf{T}$, which can be written as: $\max_{\mathbf{C}'} \sum_{t_j \in \mathbf{T}_{tar}} 1(\hat{z_{t_j}} \neq z^*_{t_j})$.

Adversarial Knowledge. The malicious workers' knowledge splits attacks into:

Black-box attacks. Adversaries only know their assigned tasks.

White-box attacks. Adversaries know the inference method and all answers.

Attack Strategy. The attackers can adopt different strategies depending on their level of knowledge.

Heuristics-based Attacks (HeurAtt). Attackers can use heuristics, e.g., always reporting the wrong answers for the tasks, or occasionally reporting true answers for some tasks to disguise themselves as honest workers. While this attack may

not be as effective as the optimization-based one below, it is easy to carry out and does not require strong adversarial knowledge, and hence is applicable to all truth inference algorithms in all black-box and white-box settings.

Optimization-based (OptAtt). Given the full knowledge, attackers can formulate an optimization problem to provide answers maximizing the number of flipped labels before and after the attack [30, 31]. Though more effective, this attack depends on the truth inference algorithm being used and requires full or partial adversarial knowledge, thus it is applicable to white-box only.

4.2 Existing Attacks

A few Optimization-based attacks (OptAtt) [30,31] have been studied for representative inference methods, namely D&S [11] and PM [28], assuming attackers' full knowledge of all answers and the inference method used (i.e. white-box setting). Since the adversary does not know the tasks' ground truth, the optimization aims to maximize the number of inferred flipped labels from before attack to after the attack, while also maximizing the inferred reliability of the attackers by the truth inference method. Intuitively, this helps them to obfuscate their malicious nature and hence succeed in misleading the system.

Let $\hat{z^a_{t_j}}$ and $\hat{z^b_{t_j}}$ denote the inferred answer for task t_j by the D&S after and before attack respectively. $\hat{\alpha}_{w'}$ and $\hat{\beta}_{w'}$ show the inferred confusion matrix parameters of the malicious worker w'. The optimization is posed as Equation (1) where λ controls the trade-off between the objectives of maximizing the collective reliability of malicious workers and the number of flipped labels.

$\max_{\mathbf{C}'} \sum_{j=1}^{M} 1(\hat{z^a_{t_j}} \neq \hat{z^b_{t_j}}) + \lambda \sum_{w' \in \mathbf{W}'} (\hat{\alpha}_{w'} + \hat{\beta}_{w'})$ (1).

For our evaluation, we design HeurAtt and extend the OptAtt for D&S and PM to all truth inference methods (which we explain in Section 5.2).

5 Evaluation Methodology

5.1 Datasets

We used two real datasets of decision making tasks [4,21] and synthetic datasets with varying parameters for the evaluation. The datasets' properties are summarized in Table 2.

Table 2: Properties of Datasets

Dataset	Product	PosSent	Synthetic
N (# of tasks)	8,315	1,000	[200, 40,000]
M (# of workers)	176	85	[100, 500]
V (# of answers)	24,945	20,000	[10,000, 200,000]
Redundancy (# of answers per task)	3	20	[5, 30]
Engagement (# of answers per worker)	141	235	[100, 400]
Skewness	0.88	0.52	[0.5 0.9]
Avg workers' reliability	0.79	0.798	0.85
Truth Labels Ratio (negative,positive)	(88%, 12%)	(52.8%, 47.2%)	(50%, 50%)

Table 3: Data Poisoning Attack Parameters

Parameter	Description	Values
$\frac{\lvert W' \rvert}{\lvert W' \rvert + \lvert W \rvert}$	Percentage of malicious workers	[10%, 60%], step = 10%, **30%**
α	Reliability for tasks with truth 1	[0, 1], step= 0.1, **1.0**
β	Reliability for tasks with truth 0	[0, 1], step= 0.1, **1.0**
α'	Attacker reliability for tasks with truth 1	[0, 1], step= 0.1, **0.0**
β'	Attacker reliability for tasks with truth 0	[0, 1], step= 0.1, **0.0**
γ	Disguise	[0, 1], step= 0.1,**0.0**

Product Dataset. The task is comparing two products, e.g. "Are iPad Two 16GB WiFi White and iPad 2nd generation 16GB WiFi White the same?" [21].
PosSent Dataset. The task is classifying positive/neagtive tweets about the companies' reputation, e.g. "The recent products of Apple is amazing" [4].
Synthetic Dataset. Synthetic datasets are created to see the effect of redundancy, number of tasks, and number of workers on performance. The worker-task assignment graph comes from Power-law distribution. The ground truth for tasks comes from a Bernoulli distribution with prior 0.5, i.e a balanced dataset. We experimented changing the prior and observed no significant change and hence settled on 0.5. The workers' reliability α and β come from Beta distribution.

5.2 Attack Design

For evaluation, assuming adversaries W', with the fraction of malicious workers being $\frac{|W'|}{|W|+|W'|}$, we design comprehensive attacks, HeurAtt and OptAtt, for all the chosen inference methods in Section 3. Both targeted and untargeted attacks are analyzed when feasible. The parameters used in design of attacks are given in Table 3. Default values (highlighted) were used unless said otherwise.
HeurAtt. We design heuristics-based attacks applicable in black-box settings.

Untargeted Attacks. With only knowledge of the assigned tasks, the simplest heuristic for attackers is to always report the wrong answer for their tasks. However, this may be easily detected by most truth inference systems (except majority voting) with workers' reliability modeling. Therefore attackers desire to disguise themselves as honest workers by providing correct answers to some tasks to avoid detection and being discounted later. To model such behavior, we use the following enhanced heuristic applicable to all truth inference methods. Each malicious worker behaves as a normal worker modeled by π^w with a disguise probability γ and as a malicious worker with probability $1-\gamma$ modeled by $\pi^{w'}$. For example, a malicious worker with a moderate level of disguise may be modeled with $\pi^w = \begin{bmatrix} \beta=1 & 0 \\ 0 & \alpha=1 \end{bmatrix}$, $\pi^{w'} = \begin{bmatrix} \beta'=0 & 1 \\ 1 & \alpha'=0 \end{bmatrix}$, and $\gamma = 0.2$.

Targeted Attacks. For targeted attacks, the best strategy for the malicious workers is to flip the labels for the targeted tasks while acting truthfully for other tasks, escaping detection and building their reliability. Hence the disguise parameter (γ) is set to be 0.
OptAtt. We design optimization-based attacks applicable to white-box settings, i.e., the attacker know the inference method used and all or part of others' answers.

Untargeted Attacks. We extend the attack from [30] to all inference methods. For confusion matrix based ones, we use the same formulation as equation (1). For methods with a single reliability parameter or no worker model, we set $\lambda = 0$.

Targeted Attacks. Here the aim is to maximize the number of targeted tasks $\mathbf{T}_{tar} \subseteq \mathbf{T}$ whose label is flipped. The optimization is $\max_{\mathbf{C}'} \sum_{t_j \in \mathbf{T}_{tar}} 1(\hat{z^a_{t_j}} \neq \hat{z^b_{t_j}}) + \lambda \sum_{w' \in \mathbf{W}'} (\hat{\alpha}_{w'} + \hat{\beta}_{w'})$. Similar to untargeted attacks, the second term is considered only for confusion matrix based methods. We varied λ in the interval [0,1] and observed that a λ value in [0.9, 1] leads to the most successful attack.

5.3 Metrics

We use the following metrics to assess the robustness of inference methods.

Accuracy. Accuracy is the fraction of correctly inferred tasks, formulated as: $\sum_{j=1}^{N} 1(\hat{z_{t_j}} = z^*_{t_j})/N$, where $\hat{z_{t_j}}$ and $z^*_{t_j}$ are inferred and ground truth of the jth task. A lower accuracy means a more successful attack. We note that [30] defined the attack success metric as the percentage of inferred labels flipped due to attack, comparing labels before and after the attack. We believe this metric does not truly capture the attackers' success where some inferred labels may be wrong without attack and were flipped to correct due to the attack, i.e., adversaries help the system to correctly infer the label of an otherwise wrongly labeled task. Instead, we use the flipped labels w.r.t. the ground truth, i.e. accuracy, as the metric for attack success. We also report F1 score to account for the skewness of classes in the unbalanced dataset.

Accuracy_Targeted. *Accuracy_Targeted* is the fraction of the targeted tasks $\mathbf{T}_{tar}$ whose truth are inferred correctly, i.e $1 - \sum_{t_j \in \mathbf{T}_{tar}} 1(\hat{z_{t_j}} \neq z^*_{t_j})/|\mathbf{T}_{tar}|$.

Area Under Curve (AUC). Since the inference methods' accuracy changes over parameters, e.g. percentage of malicious workers, we use AUC to compare the global performance of methods on an interval of parameter values, if feasible.

Recognizability. To assess the adversary detection ability of inference methods with explicit worker models, we define *Recognizability* as the similarity between the simulated (ground truth) worker reliability and the inferred reliability. A higher *Recognizability* means the method is better at detecting malicious workers. The worker behavior is modeled by a normal confusion matrix with α and β, a malicious one with α' and β', and a disguise parameter γ. We aggregate these into a single value $r_{w'}$ showing the expected reliability of a worker and define *Recognizability* as $1 - \frac{1}{|\mathbf{W}'|}\sum_{w' \in \mathbf{W}'} |r_{w'} - \hat{r}_{w'}|$, where $r_{w'}$ and $\hat{r}_{w'}$ are the simulated and inferred reliability of malicious worker w' respectively.

$$r_{w'} = \frac{1}{|\ \mathbf{T}_{w'}\ |} \sum_{t_j \in \mathbf{T}_{w'}} (\alpha_{w'} \times \gamma + \alpha'_{w'} \times (1-\gamma)) \times 1(z^*_{t_j} = 1) + (\beta_{w'} \times \gamma + \beta'_{w'} \times (1-\gamma)) \times 1(z^*_{t_j} = 0) \quad (2)$$

$$\hat{\mathrm{r}}_{w'} = \frac{1}{|\mathbf{T}_{w'}|} \sum_{t_j \in \mathbf{T}_{w'}} \hat{\alpha_{w'}} \times 1(z^*_{t_j} = 1) + \hat{\beta_{w'}} \times 1(z^*_{t_j} = 0) \quad (3)$$

6 Evaluation Results

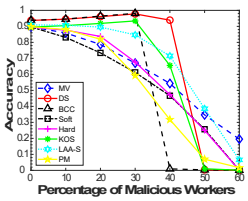
In this section, we report the robustness of various truth inference methods under heuristic-based (HeurAtt) and optimization-based (OptAtt) attacks.

6.1 Heuristic-Based Attacks (HeurAtt): Untargeted

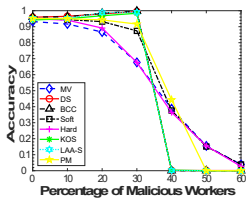
In untargeted attacks, the goal is decreasing the overall accuracy of the system.

Impact of Percentage of Malicious Workers. Here the number of normal workers is fixed and the percentage of added malicious workers varies from 0 to 60%. Adversary behavior setting is $\gamma = 0$, $\alpha' = 0$ and $\beta' = 0$.

Accuracy. Figure 2 shows the accuracy of the methods w.r.t. the percentage of malicious workers. We omitted the result for synthetic dataset showing

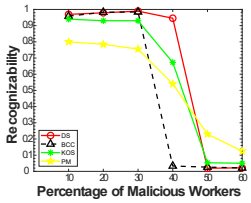


(a) Product

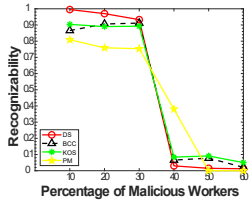


(b) PosSent

Fig. 2: Untargeted HeurAtt: Accuracy vs. % of Malicious Workers



(a) Product



(b) PosSent

Fig. 3: Untargeted HeurAtt: Recognizability vs. % of Malicious Workers

similar trends. Increasing the number of malicious workers drops the accuracy of all methods. The direct computation (MV, MV-Soft, MV-Hard) and neural network method's (LAA-S) drop is almost linear early on, PGM methods (D&S, BCC, and KOS) and probabilistic method (PM) are more resistant especially with few adversaries and drop to 0 once the percentage goes beyond (40% to 50%). Overall, D&S and LAA-S are the most resilient for this attack. Comparing the datasets, Product dataset is more susceptible to the attack due to its low redundancy. Table 4 shows the AUC of methods for Product dataset that confirms the same patterns for the relative global performance of methods. Since the AUC of PosSent dataset for all methods is around 34, there is no clear winner among them.

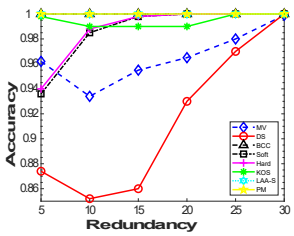
Table 4: AUC of Methods' Accuracy w.r.t. % of Malicious Workers: Untargeted HeurAtt

	MV	**DS**	**BCC**	**Soft**	**Hard**	**KOS**	**LAAS**	**PM**
Product	34.78	**42.9**	33.69	32.2	34.3	38.5	**40.09**	30.92
PosSent	32.88	34.01	34.2	34.99	34.215	33.6	34.062	34.12

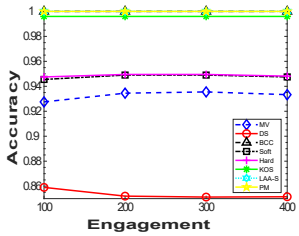
Recognizability. To show adversary detectability in inference methods, we report recognizability of methods with explicit worker modeling, i.e. D&S, BCC, PM, and KOS. We exclude MV and LAA-S as they do not explicitly model reliability. MV-Hard and MV-Soft are excluded too since recognizability is over the average adversaries' reliability and they only remove the least credible worker.

Figure 3 shows the methods' recognizability w.r.t. varying percentages of malicious workers. We omitted the result for synthetic dataset which was similar. D&S and KOS perform better than BCC in adversary detection, while PM performs the worst. This explains the robustness of the accuracy of D&S and KOS we observed earlier. Comparing Figure 2 and 3, the accuracy and recognizability of D&S and KOS decrease as the percentage of malicious workers increases, i.e. worker modeling with good detection is the key to a robust inference algorithm under attack.

Impact of Redundancy and Engagement. Redundancy is the mean number of workers assigned per task, while worker engagement is the mean number of tasks per worker. Figure 4 shows the accuracy w.r.t. varying redundancy and engagement values. As expected, with increased redundancy, it is harder for adversaries to reduce the accuracy (the percentage of attackers is set to 20%). D&S and MV-based models were more sensitive to redundancy in the sparse

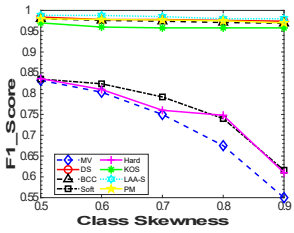


(a) Redundancy



(b) Engagement

Fig. 4: Untargeted HeurAtt: Accuracy vs Redundancy and Engagement (Synthetic)



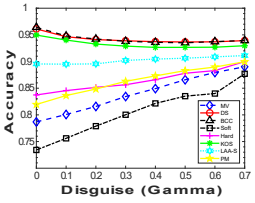
(a) Skewness

Fig. 5: Untargeted HeurAtt: F1 Score vs Class Skewness

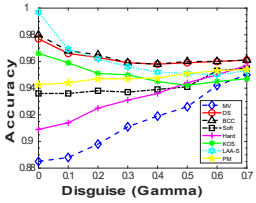
dataset (Product). Worker engagement had no significant impact, since it does not directly impact their reliability.

Impact of Class Skewness. We show the effect of skewness on F1-score in synthetic data. Figure 5a shows the F1-score w.r.t. varying ratios of the majority class. MV-based methods are vulnerable to imbalance while others are robust.

Impact of Disguise (γ). We show the trend of accuracy and recognizability w.r.t disguise. When in disguise, adversaries' behavior is governed by $\alpha = 1$ and $\beta = 1$ compared to $\alpha' = 0$ and $\beta' = 0$ in pure malicious mode.

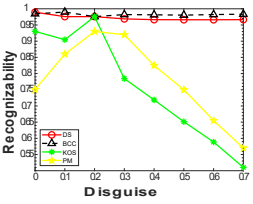


(a) Product

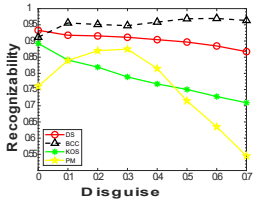


(b) PosSent

Fig. 6: Untargeted HeurAtt: Accuracy vs Disguise



(a) Product



(b) PosSent

Fig. 7: Untargeted HeurAtt: Recognizability vs. Disguise (γ)

Accuracy. Figure 6 shows the accuracy of methods w.r.t. varying disguise levels. We use a different scale for each dataset's y-axis to highlight their trend. Since increasing disguise after 0.7 resulted in monotonously increasing accuracy for all models, we terminate at 0.7. For methods (e.g. MV) with no inherent attacker recognition, disguising only boosts the accuracy. For more robust methods, as we increase γ slightly, the algorithms fail to identify adversaries leading to the success of the attack. However, as disguise further increases, the accuracy goes back up due to the correct answers by the disguised malicious workers. Hence there's an optimal level of disguise for attackers.

Recognizability. Figure 7 shows the recognizability of methods w.r.t. varying disguise levels on real datasets. Confusion matrix based models, BCC and D&S, are more robust to disguise and model the workers accurately regardless of adversarys' disguise. KOS uses a single reliability value and thus is more sensitive to disguise. Generally, the recognizability of adversaries drops as disguise increases, since their behavior more closely resembles normal workers.

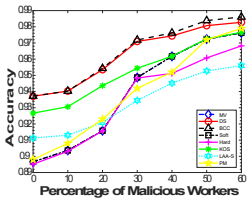
Impact of Malicious Worker Confusion Matrix. We also evaluated the impact of varying worker behavior parameter (α', β'). When varying α', β' is 0,

and vice versa. Increasing adversaries' reliability parameters is very similar to increasing disguise. For brevity, we omit the figures. Simple methods e.g. MV-based, have increasing accurasy due to the adversaries' true answers. In more sophisticated methods e.g PGM based, attack can be most successful with an optimal value.

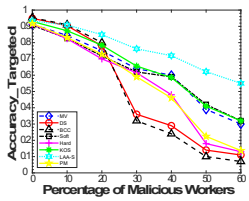
6.2 Heuristic-Based Attacks (HeurAtt): Targeted

We report the evaluation results for targeted attacks as outlined in Section 5.2. We focus on parameters relevant to targeted attacks, the percentage of malicious workers and the proportion of targeted tasks.

Impact of Percentage of Malicious Workers. Figure 8 and 9 show the accuracy of the methods w.r.t. varying percentage of malicious workers on the real datasets. We fixed the fraction of targeted tasks for Product and PosSent dataset to be 0.2 and 0.1, respectively, based on two factors: 1) targeted attack is impactful, 2) there is an observable difference among methods' performance. The general trend is that increasing the number of attackers, the overall accuracy of the system increases thanks to the truthful contributions of attackers to non-targeted tasks. However, the accuracy of targeted tasks is decreased by attackers.

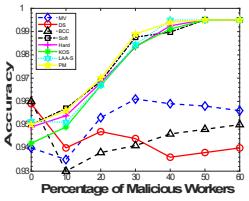


(a) Accuracy

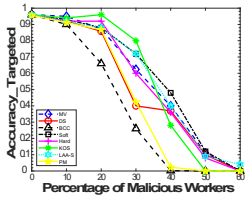


(b) Accuracy_Targeted

Fig. 8: Targeted HeurAtt: Accuracy & Accuracy_Targeted vs. % of Malicious Workers (Product dataset)



(a) Accuracy



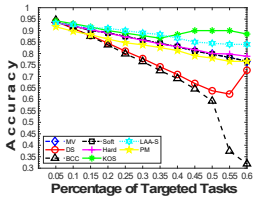
(b) Accuracy_Targeted

Fig. 9: Targeted HeurAtt: Accuracy & Accuracy_Targeted vs. % of Malicious Workers (PosSent dataset)

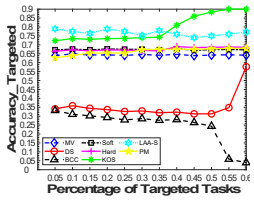
Remarkably, D&S and BCC which are more robust against untargeted attacks are more susceptible to targeted attacks. While they maintain high overall accuracy, their accuracy for the targeted tasks suffers the most due to their failure to differentiate targeted and untargeted tasks when modeling workers' behavior (i.e. being misled by malicious workers based on their true answers to the untargeted tasks). On the other hand, LAA-S is significantly more robust against targeted attacks, even though the overall accuracy is not as high as other methods, explained by the absence of explicit worker modeling. While MV-Soft performs worse than others in untargeted attacks (Figure 2), it is the most resilient alongside LAA-S for targeted attacks. MV-Soft's resilience is due to accurate detection and penalization of malicious workers when they are the majority contributing to a task with conflicting answers, that happen more frequently while focusing on limited tasks rather than all tasks.

Impact of Proportion of Targeted Tasks. Figure 10 and 11 show the accuracy of the methods w.r.t. varying percentage of targeted tasks on real datasets. As the ratio of targeted tasks increases, the overall accuracy and targeted accuracy decrease. However, when the ratio gets sufficiently large, accuracy increase

for D&S and BCC. Since malicious workers have to dilute their efforts among a larger set of targeted tasks, they are more discoverable and less effective.

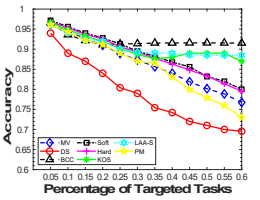


(a) Accuracy

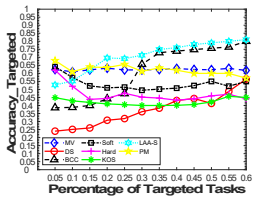


(b) Accuracy_Targeted

Fig. 10: Targeted HeurAtt: Accuracy vs. Ratio of Targeted Tasks (Product dataset)



(a) Accuracy



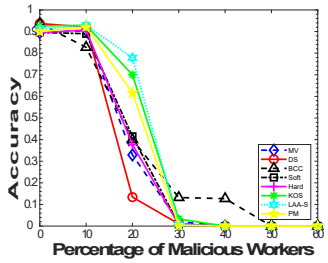
(b) Accuracy_Targeted

Fig. 11: Targeted HeurAtt:Accuracy vs. Ratio of Targeted Tasks (PosSent dataset)

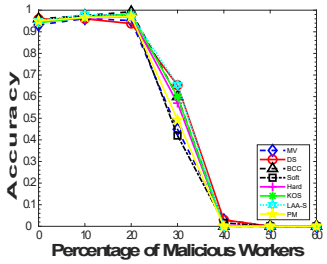
Comparing the datasets, answer redundancy inversely affect targeted attack's success, similar to untargeted attacks. Given the Product dataset's lower redundancy, this attack is successful even with a high ratio of targeted tasks.

6.3 Optimization Based Attacks (OptAtt): Untargeted

Impact of Percentage of Malicious Workers (White-Box Attack). We evaluate the OptAtt in white-box setting, i.e. given the full knowledge of the inference method used and all other workers' answers. Figure 12 shows the accuracy of the inference methods w.r.t. varying percentage of malicous workers.



(a) Product

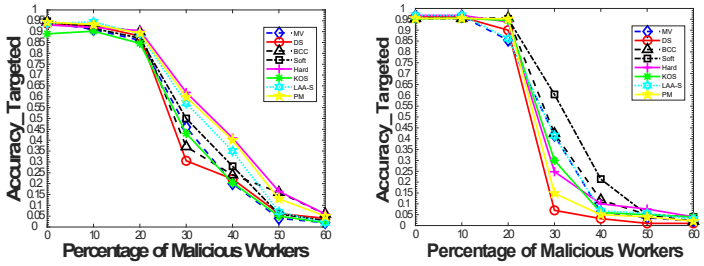


(b) PosSent

Fig. 12: Untargeted OptAtt: Accuracy vs. % of Malicious Workers

All methods' accuracy drops fairly quickly as the percentage of malicious workers increases. Comparing HeurAtt and OptAtt (Figure 2a vs Figure 12a for Product dataset, Figure 2b vs Figure 12b for PosSent dataset), the accuracy under OptAtt drops to zero at a much lower percentage of malicious workers for all methods. The attackers are indeed more successful when using the optimized scheme through stronger adversarial knowledge. Comparing methods, all perform similarly in resiliency and are susceptible to the attack, since it is optimized for that particular inference method. However, LAA-S has a slight edge over others.

The accuracy drops and attack is more successful with more adversarial knowledge. The attack is still quite successful even when a very low percentage of other's answers are known. One explanation is that the reliability of workers for both datasets is quite uniformly distributed around 0.79. Thus for all meth-



(a) Product (b) PosSent

Fig. 13: Targeted OptAtt: Accuracy_Targeted vs. % of Malicious Workers on Product and PosSent dataset

ods, even with a small fraction of available normal workers' answers (i.e. 0.2), the adversaries estimate the truth quite accurately.

6.4 Optimization Based Attacks (OptAtt): Targeted

Impact of Percentage of Malicious Workers. We set the ratio of targeted tasks to be 0.01 and 0.005 for Product and PosSent datasets respectively. Since OptAtt is more successful, a lower ratio of targeted tasks is chosen here compared to Section 6.2 to portray the same regions of interest for methods' performance. Figure 13 shows accuracy_targeted of the inference methods w.r.t. varying percentage of malicious workers on the real datasets. Overall accuracy is not shown due to space limitations. *Accuracy_Targeted* decreases as percentage of malicious workers increases. Comparing targeted attack in HeurAtt and OptAtt (Figures 8 & 9), HeurAtt is more effective in reducing accuracy at a small percentage of malicious workers. However, with a greater percentage of malicious workers, targeted OptAtt attack is more successful. One probable reason is since a subset of tasks is targeted, the chance of adversary's detection is lower compared to the untargeted setting. So, OptAtt that trades some accuracy drop in exchange for less detection will lose its edge in attack power over HeurAtt which only focuses on accuracy.

7 Conclusion & Discussion

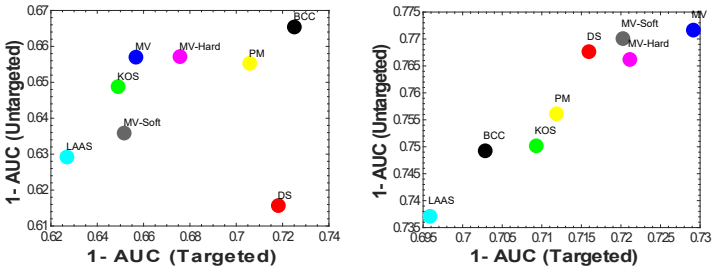
We summarize our key findings on the performance of leading inference methods using diverse techniques under various data poisoning attacks.

Table 5: Top 2 Robust Methods under Different Attacks

Strategy / Goal	**HeurAtt**	**OptAtt**
Untargeted	D&S (LAA-S)	LAA-S (KOS)
Targeted	LAA-S (MV-Soft)	LAA-S (BCC)

Comparison of Methods. Figure 14 shows the overall attack susceptibility of different inference methods along two dimensions (untargeted attacks and targeted attacks) under HeurAtt and OptAtt attacks. Susceptibility is defined as 1 - AUC. A more robust method should have a lower susceptibility across both dimensions. The AUC is the accuracy over an interval of fraction of attackers. Since in reality malicious workers are not the majority, we choose the interval [0, 0.5]. The most robust methods should be those dominating others (less vulner-

able) in both dimensions, the pareto optimal methods or skyline. Table 5 shows the top 2 performing methods for each category of attacks. We also discuss the main findings below.



(a) HeurAtt (b) OptAtt

Fig. 14: Susceptibility of Different Inference Methods

- Among direct computation methods, MV-Soft is more robust than MV, i.e. dominates MV, for all attacks, thanks to its modeling of worker reliability.
- Among OptAtt, MV-Hard is more robust than PM only under targeted HeurAtt. This can be attributed to its optimal matching algorithm that penalizes or removes malicious workers when they become the majority among the contributing workers for conflicting tasks, which happens when they all give wrong answers to a target set of tasks. On the other hand, PM is more robust than MV-Hard under other attacks.
- For untargeted attacks, among PGM based methods, D&S dominate BCC under HeurAtt. However, under OptAtt, BCC dominates D&S. Note that in HeurAtt the responses are designed for each individual worker, while under OptAtt, the collective effect of malicious workers is considered. Therefore correlation modeling in BCC can counter the possible collusion of malicious workers, reflected in BCC's more resilient behavior under OptAtt. Overall, D&S is most resilient if only untargeted HeurAtt is anticipated. However, D&S is vulnerable to all other attacks. Between KOS and BCC, KOS has a slight edge considering all four attacks.
- The neural network based method LAA-S dominates all others in all attacks except having a slightly higher susceptibility than D&S in untargeted HeurAtt in the low redundancy dataset (Product). We suspect this is due to the sparsity of the task representation vector in contrast to the large number of network parameters combined with the lower quality of malicious answers in HeurAtt. The superior performance in all other cases can be attributed to its network of parameters, which can be considered as an implicit and sophisticated (non-linear) model of worker reliability.
- Comparing different techniques, besides the best performing neural networks, PGM based methods are generally more robust than optimization based methods and direct computation methods. This is also consistent with the findings in [56] under normal settings with varying worker reliability.

– Comparing different worker models, the confusion matrix based methods generally outperform those with a single reliability model and MV method with no worker model. Neural network based method LAA-S, even though with no explicit worker model, achieves the best performance thanks to its network of parameters, which can be considered as an implicit and more sophisticated (non-linear) model of the reliability associated with each worker.

Comparison of Attacks. From the attack point of view, OptAtt is more effective compared to HeurAtt, especially for untargeted attacks. This is not surprising given the adversarial knowledge and OptAtt strategy. While OptAtts can only be carried out in white-box settings given full adversarial knowledge, as we have shown, they can be very effective, however it is not a realistic threat to crowdsourcing applications.

For targeted attacks, in a lower percentage of adversaries, HeurAtt can be more successful compared to OptAtt. OptAtt trades off some accuracy drop in exchange for less adversary detection, hence losing its edge in attack power over HeurAtt, solely optimizing for accuracy drop. There is an optimal level of disguise and percentage of targeted tasks for attackers under HeurAtt, however, these may not be easily identifiable as they vary substantially across inference methods and settings.

Comparison of Datasets and Other Factors. The comparison between the datasets reveals that a crowdsourcing system with a higher redundancy of answers is generally more robust. It remains an interesting question for a crowdsourcing system provider to find the best trade-off between redundancy and the overall platform cost to ensure the resiliency of the system. However, worker engagement does not have a major impact on the robustness of the system.

Future Works. The goal of our evaluation is to understand the resilience of existing leading-edge truth inference methods and ultimately build more robust systems. Towards this end, several directions for future work can be explored.

– Attack resistant truth inference: while existing methods provide certain level of resistance to data poisoning attacks, it remains an open question whether we can build more robust systems, e.g. by designing hybrid methods that combine the strength of existing techniques (e.g. LAA-S against OptAtt and targeted HeurAtt and D&S against untargeted HeurAtt) or designing entirely new techniques. A more fundamental question is whether we can have quantifiable guarantees for the robustness of the system against data poisoning attacks.
– Semi-supervised learning: All the truth inference approaches evaluated in this paper assume no access to the ground truth. This unsupervised nature makes the methods inherently susceptible to attacks. A semi-supervised approach [3, 42] may help provide resilience to attacks within which ground truth and workers' answers for historical tasks can be used for training the system.
– Dynamic behavior of workers: Our study is focused on a fixed set of workers and tasks. In practice, the workers could join and leave the system as they wish. They could also alter their behaviors dynamically in a strategic way to maximize the attack effect. Understanding the impact of such behaviors and building robust systems in a dynamic setting is also an important direction.

References

1. Al-Qurishi, M., Al-Rakhami, M., Alamri, A., Alrubaian, M., Rahman, S.M.M., Hossain, M.S.: Sybil defense techniques in online social networks: a survey. IEEE Access **5**, 1200–1219 (2017)
2. Alsuwat, E., Alsuwat, H., Rose, J., Valtorta, M., Farkas, C.: Detecting adversarial attacks in the context of bayesian networks. In: IFIP Annual Conference on Data and Applications Security and Privacy. pp. 3–22. Springer (2019)
3. Atarashi, K., Oyama, S., Kurihara, M.: Semi-supervised learning from crowds using deep generative models. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
4. Authors, M.: Twitter Sentiment (2018), https://raw.githubusercontent.com/zfz/twitter_corpus/master/full-corpus.csv, [Online; accessed 19-April-2018]
5. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
6. Brawley, A.M., Pury, C.L.: Work experiences on mturk: Job satisfaction, turnover, and information sharing. Computers in Human Behavior **54**, 531–546 (2016)
7. Bryan, K., O'Mahony, M., Cunningham, P.: Unsupervised retrieval of attack profiles in collaborative recommender systems. In: Proceedings of the 2008 ACM conference on Recommender systems. pp. 155–162. ACM (2008)
8. Cao, Q., Yang, X., Yu, J., Palow, C.: Uncovering large groups of active malicious accounts in online social networks. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 477–488. ACM (2014)
9. Chirita, P.A., Nejdl, W., Zamfir, C.: Preventing shilling attacks in online recommender systems. In: Proceedings of the 7th annual ACM international workshop on Web information and data management. pp. 67–74. ACM (2005)
10. Choi, H., Lee, K., Webb, S.: Detecting malicious campaigns in crowdsourcing platforms. In: Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 197–202. IEEE Press (2016)
11. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the em algorithm. Applied statistics pp. 20–28 (1979)
12. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st international conference on World Wide Web. pp. 469–478. ACM (2012)
13. Difallah, D.E., Demartini, G., Cudré-Mauroux, P.: Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In: CrowdSearch. pp. 26–30 (2012)
14. Douceur, J.R.: The sybil attack. In: International workshop on peer-to-peer systems. pp. 251–260. Springer (2002)
15. Gaunt, A., Borsa, D., Bachrach, Y.: Training deep neural nets to aggregate crowdsourced responses. In: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence. AUAI Press. p. 242251 (2016)
16. Gunes, I., Kaleli, C., Bilge, A., Polat, H.: Shilling attacks against recommender systems: a comprehensive survey. Artificial Intelligence Review **42**(4), 767–799 (2014)
17. Hong, C., Zhou, Y.: Label aggregation via finding consensus between models. arXiv preprint arXiv:1807.07291 (2018)
18. Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B.I., Tygar, J.: Adversarial machine learning. In: Proceedings of the 4th ACM workshop on Security and artificial intelligence. pp. 43–58. ACM (2011)
19. Hung, N.Q.V., Thang, D.C., Weidlich, M., Aberer, K.: Minimizing efforts in validating crowd answers. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data. pp. 999–1014. ACM (2015)
20. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on amazon mechanical turk. In: Proceedings of the ACM SIGKDD workshop on human computation. pp. 64–67. ACM (2010)
21. J. Wang, T. Kraska, M.J.F., Crowderd, J.F.: Crowdsourcing entity resolution. PVLDB **5**(11), 1483–1494 (2012)
22. Jagabathula, S., Subramanian, L., Venkataraman, A.: Reputation-based worker filtering in crowdsourcing. In: Advances in Neural Information Processing Systems. pp. 2492–2500 (2014)

23. Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., Li, B.: Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 19–35. IEEE (2018)
24. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: Advances in neural information processing systems. pp. 1953–1961 (2011)
25. Kim, H.C., Ghahramani, Z.: Bayesian classifier combination. In: Artificial Intelligence and Statistics. pp. 619–627 (2012)
26. Levine, B.N., Shields, C., Margolin, N.B.: A survey of solutions to the sybil attack. University of Massachusetts Amherst, Amherst, MA **7**, 224 (2006)
27. Li, Q., Li, Y., Gao, J., Su, L., Zhao, B., Demirbas, M., Fan, W., Han, J.: A confidence-aware approach for truth discovery on long-tail data. Proceedings of the VLDB Endowment **8**(4), 425–436 (2014)
28. Li, Q., Li, Y., Gao, J., Zhao, B., Fan, W., Han, J.: Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp. 1187–1198. ACM (2014)
29. Li, Y., Gao, J., Meng, C., Li, Q., Su, L., Zhao, B., Fan, W., Han, J.: A survey on truth discovery. ACM Sigkdd Explorations Newsletter **17**(2), 1–16 (2016)
30. Miao, C., Li, Q., Su, L., Huai, M., Jiang, W., Gao, J.: Attack under disguise: An intelligent data poisoning attack mechanism in crowdsourcing. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web. pp. 13–22. International World Wide Web Conferences Steering Committee (2018)
31. Miao, C., Li, Q., Xiao, H., Jiang, W., Huai, M., Su, L.: Towards data poisoning attacks in crowd sensing systems. In: Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing. pp. 111–120. ACM (2018)
32. Mobasher, B., Burke, R., Bhaumik, R., Sandvig, J.J.: Attacks and remedies in collaborative recommendation. IEEE Intelligent Systems **22**(3), 56–63 (2007)
33. Nguyen, A.T., Wallace, B.C., Lease, M.: A correlated worker model for grouped, imbalanced and multitask data. In: UAI (2016)
34. O'Mahony, M., Hurley, N., Kushmerick, N., Silvestre, G.: Collaborative recommendation: A robustness analysis. ACM Transactions on Internet Technology (TOIT) **4**(4), 344–377 (2004)
35. Raykar, V.C., Yu, S.: Eliminating spammers and ranking annotators for crowdsourced labeling tasks. Journal of Machine Learning Research **13**(Feb), 491–518 (2012)
36. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. Journal of Machine Learning Research **11**(Apr), 1297–1322 (2010)
37. Shafahi, A., Huang, W.R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., Goldstein, T.: Poison frogs! targeted clean-label poisoning attacks on neural networks. In: Advances in Neural Information Processing Systems. pp. 6106–6116 (2018)
38. Sheng, V.S., Zhang, J.: Machine learning with crowdsourcing: A brief summary of the past research and future directions. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 9837–9843 (2019)
39. Stringhini, G., Mourlanne, P., Jacob, G., Egele, M., Kruegel, C., Vigna, G.: {EVILCOHORT}: Detecting communities of malicious accounts on online services. In: 24th {USENIX} Security Symposium ({USENIX} Security 15). pp. 563–578 (2015)
40. Suciu, O., Marginean, R., Kaya, Y., Daume III, H., Dumitras, T.: When does machine learning {FAIL}? generalized transferability for evasion and poisoning attacks. In: 27th {USENIX} Security Symposium ({USENIX} Security 18). pp. 1299–1316 (2018)
41. Swain, R., Berger, A., Bongard, J., Hines, P.: Participation and contribution in crowdsourced surveys. PloS one **10**(4), e0120521 (2015)
42. Tang, W., Lease, M.: Semi-supervised consensus labeling for crowdsourcing. In: SIGIR 2011 workshop on crowdsourcing for information retrieval (CIR). pp. 1–6 (2011)
43. Vasudeva, A., Sood, M.: Survey on sybil attack defense mechanisms in wireless ad hoc networks. Journal of Network and Computer Applications **120**, 78–118 (2018)

44. Venanzi, M., Guiver, J., Kazai, G., Kohli, P., Shokouhi, M.: Community-based bayesian aggregation models for crowdsourcing. In: Proceedings of the 23rd international conference on World wide web. pp. 155–164. ACM (2014)
45. Wang, G., Konolige, T., Wilson, C., Wang, X., Zheng, H., Zhao, B.Y.: You are how you click: Clickstream analysis for sybil detection. In: Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13). pp. 241–256 (2013)
46. Wang, G., Wang, B., Wang, T., Nika, A., Zheng, H., Zhao, B.Y.: Defending against sybil devices in crowdsourced mapping services. In: Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. pp. 179–191. ACM (2016)
47. Wang, G., Wang, T., Zheng, H., Zhao, B.Y.: Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In: USENIX Security Symposium. pp. 239–254 (2014)
48. Whitehill, J., Wu, T.f., Bergsma, J., Movellan, J.R., Ruvolo, P.L.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Advances in neural information processing systems. pp. 2035–2043 (2009)
49. Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B.Y., Dai, Y.: Uncovering social network sybils in the wild. ACM Transactions on Knowledge Discovery from Data (TKDD) **8**(1), 2 (2014)
50. Yin, L., Han, J., Zhang, W., Yu, Y.: Aggregating crowd wisdoms with label-aware autoencoders. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 1325–1331. AAAI Press (2017)
51. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.: Sybilguard: defending against sybil attacks via social networks. ACM SIGCOMM Computer Communication Review **36**(4), 267–278 (2006)
52. Yu, H., Shi, C., Kaminsky, M., Gibbons, P.B., Xiao, F.: Dsybil: Optimal sybil-resistance for recommendation systems. In: 2009 30th IEEE Symposium on Security and Privacy. pp. 283–298. IEEE (2009)
53. Yuan, D., Li, G., Li, Q., Zheng, Y.: Sybil defense in crowdsourcing platforms. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1529–1538. ACM (2017)
54. Zhang, K., Liang, X., Lu, R., Shen, X.: Sybil attacks and their defenses in the internet of things. IEEE Internet of Things Journal **1**(5), 372–383 (2014)
55. Zhang, Y., Tan, Y., Zhang, M., Liu, Y., Chua, T.S., Ma, S.: Catch the black sheep: unified framework for shilling attack detection based on fraudulent action propagation. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
56. Zheng, Y., Li, G., Li, Y., Shan, C., Cheng, R.: Truth inference in crowdsourcing: is the problem solved? Proceedings of the VLDB Endowment **10**(5), 541–552 (2017)
57. Zhou, D., Basu, S., Mao, Y., Platt, J.C.: Learning from the wisdom of crowds by minimax entropy. In: Advances in neural information processing systems. pp. 2195–2203 (2012)
58. Zhou, Y., He, J.: Crowdsourcing via tensor augmentation and completion. In: IJCAI. pp. 2435–2441 (2016)
59. Zhou, Y., Ying, L., He, J.: Multic2: an optimization framework for learning from task and worker dual heterogeneity. In: Proceedings of the 2017 SIAM International Conference on Data Mining. pp. 579–587. SIAM (2017)

1 Appendix:Details of Selected Truth Inference Methods

MV-Soft Majority voting with soft penalty (MV-soft) [22] implicitly models workers' reliability via penalizing unreliable workers. To compute the penalty, the algorithm considers tasks without unanimous answer from all workers, i.e. conflicted tasks. A bipartite graph is created with two set of vertices $\mathbf{W}$ and $\mathbf{T}_{conf}$, where $\mathbf{W}$ and $\mathbf{T}_{conf} \subseteq \mathbf{T}$ are sets of workers and conflicted tasks. Each conflicted task is represented by two nodes, t_j^+ and t_j^-. An edge $\{w_i, t_j^+\}$ or $\{w_i, t_j^-\}$ is added to the graph depending on answer of 1 or 0 provided by worker w_i ' for task t_j. The penalty for worker w_i is inversely proportional to the number of other workers who have the same answer as w_i, which is measured by the degree of each node in the conflicted task set $\mathbf{T}_{conf}$.

$$Pen_{w_i} = \frac{\sum_{t_j \in \mathbf{T}_{conf}^{w_i}} \frac{1}{deg(t_j^+)} .1(c_{t_j}^{w_i}=1) + \frac{1}{deg(t_j^-)} .1(c_{t_j}^{w_i}=0)}{|\mathbf{T}_{conf}^{w_i}|} \tag{4}$$

Here, $\mathbf{T}_{conf}^{w_i}$ is the conflicted tasks assigned to worker w_i and $deg(t_j^+)$, $deg(t_j^-)$ are degrees of task t_j for label 1 and 0 respectively, $c_{t_j}^{w_i}$ shows worker w_i's answer for task t_j and 1 is the identifier function which is 1 if the condition holds and 0 otherwise.

MV-Hard Here an optimal semi-matching for bipartite graph is used to assign penalty [22]. A semi-matching is a matching for the bipartite graph where exactly one worker is chosen among all contributors for each label of each task. The optimal matching minimizes the sum of accumulated degree of all workers,a typical formulation of the semi-matching problem: $\min_{Match} \quad \sum_{w\in\mathbf{W}} \sum_{i=1}^{deg_{Match}(w)} i$, where $deg_{Match}(w)$ is the degree of worker w in matching $Match$. In the optimal semi matching tasks are distributed as evenly as possible among the workers, so one worker cannot dominate others by being the only decision maker for many tasks. Assuming the worker who is the sole contributor for many tasks based on semi matching is more malicious, workers' penalties are their degree in optimal semi-matching. The final label of tasks is the label connected to the worker with the lowest degree or penalty.

PM Here worker w_i's reliability is modeled using a single value r^{w_i} and the truth inference is posed as optimization problem: $\min_{\mathbf{r},\hat{\mathbf{Z}}} \quad \sum_{w_i\in\mathbf{W}} r^{w_i} \cdot \sum_{t_j\in\mathbf{T}_{w_i}} d(\hat{z}_{t_j}, c_{t_j}^{w_i})$ where $d(\hat{z}_{t_j}, c_{t_j}^{w_i})$ for task t_j, is the distance between the inferred truth and the answer provided by worker w_i and r^{w_i} is worker w_i's reliability [28]. Intuitively, workers with answers deviating from the inferred truth are more malicious. The algorithm iterative solves the optimization problem updating the truth labels and reliability of workers.

D&S It is a Probabilistic Graphical Modelling approach showing worker's reliability by confusion matrix. The D&S [11] algorithm uses Expectation Maximization (EM) to solve maximum likelihood estimation (MLE) for the inferred labels $\hat{\mathbf{Z}}$ and the confusion matrices $\pi^{\mathbf{W}}$ iteratively. $\mathbf{W}^{t_j}$ is the workers responding to task t_j. The objective function of this method is: $\max_{\hat{\mathbf{Z}},\pi^{\mathbf{W}}} \prod_{j=1}^{N} \sum_{l\in\mathbf{L}} pr(\hat{z}_{t_j} = l) \prod_{w_i\in\mathbf{W}^{t_j}} \pi_{l,c_{t_j}^{w_i}}^{w_i}$.

BCC Bayesian Classifier Combination is also a graphical model approach [25]. For easy tasks, a single shared confusion matrix is used for all workers, whereas for hard tasks, workers have a separate confusion matrix. The inference problem is posed as:

$$\max_{\mathbf{a},\mathbf{b},\hat{\mathbf{Z}},\pi^{\mathbf{W}}} \prod_{j=1}^{N} pr(\hat{\mathbf{Z}} \mid \mathbf{a}) \prod_{i=1}^{M} pr(\pi^{w_i} \mid \mathbf{b}) \prod_{i=1}^{M} pr(c_{t_j}^{w_i} \mid \pi^{w_i}, \hat{\mathbf{Z}}) \tag{5}$$

There are extensions of the BCC method that take into account the difficulty of tasks and workers collaboration, but we do not consider them in this study.

KOS The KOS method estimates the worker reliability by maximizing the joint probability distribution [24]. $\max_{\hat{\mathbf{Z}},\pi^{\mathbf{W}}} \quad \prod_{j=1}^{N} \sum_{l\in\mathbf{L}} pr(\hat{z}_{t_j} = l) \prod_{w_i\in\mathbf{W}^{t_j}} \pi_{l,c_{t_j}^{w_i}}^{w_i}$. Since direct estimation of the distribution is intractable, an iterative belief propagation is used to estimate a distribution for the worker reliability. The label of tasks is determined based on the weighted product of estimated worker's reliability and their answers.

LAAS It is an unsupervised approach where the intuition is learning the latent true label that best represents the original task vector. LAA-S [50] architecture contains two shallow neural networks: 1) an encoder or classifier (q_θ) to convert the task vector (v) into the latent feature (z) showing the true label, and 2) a decoder (p_ϕ) reconstructing a task vector based on the latent feature.

The network is trained on all task vectors by minimizing the reconstruction error between the original and reconstructed task vectors. The training also enforces the latent truth label distribution to resemble the original answer distribution. The objective function is: $\min_{q_\theta,p_\theta} \mathbb{E}_{q_\theta(z|v)} logp_\phi(v|z) - D_{KL}(q_\theta(z|v)||\ prior)$

Here the first term is the reconstruction error and the second term ensures the distribution of inferred labels follows a specific prior using the negative KL divergence D_{KL}. The prior is set based on the fraction of labels in the workers' answers for each task.