EdgeInfer: Robust Truth Inference under Data Poisoning Attack

Farnaz Tahmasebian
Computer Science
Emory University
ftahmas@emory.edu

Li Xiong
Computer Science
Emory University
lxiong@emory.edu

Mani Sotoodeh
Computer Science
Emory University
msotood@emory.edu

Vaidy Sunderam
Computer Science
Emory University
vss@emory.edu

***Abstract*—As crowdsourcing is becoming more widely used for annotating data from a large group of users, attackers have strong incentives to manipulate the system. Deriving the true answer of tasks in crowdsourcing systems based on user-provided data is susceptible to data poisoning attacks, whereby malicious users may intentionally or strategically report incorrect information to mislead the system into inferring the wrong truth for a set of tasks. Recent work has proposed several attacks on the crowdsourcing systems and showed that existing truth inference methods may be vulnerable to such attacks. In this paper, we propose solutions to enhance the robustness of existing truth inference methods. Our solutions base on 1) detecting and augmenting the answers for the boundary tasks in which users could not reach a strong consensus and hence are subjective to potential manipulation, and 2) enhancing inference method with a stronger prior. We empirically evaluate these defense mechanisms by designing attack scenarios that aim to decrease the accuracy of the system. Experiments show that our method is effective and significantly improves the robustness of the system under attack.**

***Index Terms*—Robustness, Data poisoning attack, Truth Inference, Crowdsourcing**

I. Introduction

Crowdsourcing is a paradigm that provides a cost-effective solution for obtaining services or data from a large group of users, or *crowd*. Amazon's Mechanical Turk (MTurk) and Waze are well-known examples of crowdsourcing systems that are aggregating human wisdom to estimate the true answer for their corresponding tasks. Many businesses use MTurk to complete simple tasks, for example, tagging images or completing a survey [20]. Another example is Waze [35], a crowd-driven navigation application. Users can report the traffic status at various locations which is then aggregated to update the traffic condition shown on the map.

Although crowdsourcing is a cost-effective solution, attackers could easily take advantage of it and exploit a large number of workers to artificially elevate or reduce support for products or opinions. For example, the rating system of restaurants in the Yelp application could be manipulated by creating fake reviews. Studies have shown [3], [25] that the revenue of restaurants in Yelp application is increased up to 9% when the rating score of that restaurant is increased just by one score.

Since in the crowdsourcing system the answers are collected from non-expert workers, the collected answers often contain inherent noise. One important component of crowdsourcing systems is *truth inference*, which infers the true labels from the answers provided by workers. Majority voting (MV) is a straightforward method to aggregate answers which naively assumes that all workers have the same reliability. Besides MV, advenced methods such as probabilistic graphical model (PGM) based, and neural network based [6], [7], [15], [17], [29], [39] methods have been proposed to improve the performance of truth inference by considering various parameters such as the reliabilities of workers or the difficulties of tasks.

Most truth inference methods were designed without consideration for malicious intents. However, crowdsourcing applications may be subject to *data poisoning attacks* [16], [32], [36] where malicious users may intentionally and strategically report incorrect information to mislead the system to infer the wrong truth for all or a targeted set of tasks. In the Waze example, the competitors may intend to tarnish Waze's reputation by providing wrong answers to decrease the overall accuracy of the system. This can be often achieved via Sybil attacks [5], [9], [38], [40] where an attacker creates a large number of Sybil workers to strategically report wrong answers.

Concretely, malicious workers may disguise themselves as normal workers by providing reliable answers for certain tasks such that they escape the worker reliability model while providing wrong answers for other tasks. With sufficient adversarial knowledge, they may even optimize their answers in a way that maximizes the error of the truth inference system, as outlined in [26], [27].

The two experimental studies [33], [42] evaluated the truth inference methods. Zheng et al. [42] evaluation is focused on "normal" settings where workers may have varying reliability but do not intentionally or strategically manipulate the answers. However, Tahmasebian et al. [33] evaluation is focused on "adversarial" settings. In the "normal" setting, the study [42] concluded that truth inference methods that utilize a probabilistic graphical model (PGM) have the best performances in most settings. Besides the study in the "adversarial" settings [33] showed that neural networks and PGM based methods are generally more robust than other methods. Also, it is shown that existing truth inference methods fail to infer the labels accurately under various attack setting, hence motivate the need for more robust truth inference algorithms.

In this paper, we propose a data augmentation method focused on boundary tasks that can be used to enhance the

robustness of existing truth inference methods against potential data poisoning attacks. Our key insight is that the boundary tasks for which workers fail to reach a strong consensus are particularly vulnerable to manipulations and may lead to wrong inferences. We can mitigate the risks by augmenting answers for these boundary tasks before applying inference algorithms. We further propose an improved inference algorithm with a stronger prior obtained from the answers. We summarize our contributions below.

- We present a data augmentation method (EdgeInfer) focused on boundary tasks that can be used to enhance the robustness of existing truth inference methods against potential data poisoning attacks. The intuition behind this is that boundary tasks are more likely to be targeted by the malicious workers to achieve a successful attack due to the weak agreement among contributing workers. This method can be used as a preprocessing step to enhance existing truth inference algorithms.
- As shown in the experimental survey [33] the state-of-the-art methods based on neural networks and PGM perform better and generally more robust. Therefore, we propose Edge-NN and Edge-PGM that are based on neural networks and PGM models and utilizing prior information to enhance these methods. In Edge-NN, we propose an enhanced neural network based inference method by replacing raw data distribution based prior with a stronger prior inferred from a probabilistic graphical model (PGM). By incorporating the prior, the method takes advantage of two sources of knowledge from two distinctive and complementary models, which promises a boosted performance in terms of accuracy and robustness. In Edge-PGM, we propose an enhanced truth inference method based on PGM by taking advantage of boundary tasks and curating a better prior for it. This PGM inference method incorporates the difficulty level of tasks into their model, however, the estimated difficulty level of tasks is not quite certain. Therefore, utilizing a stronger prior of the difficulty level of tasks can enhance the truth inference method.
- We conduct experiments using three real datasets under different data poisoning attacks in crowdsourcing. The results verify that the proposed approach outperforms state-of-the-art truth inference methods under a variety of attack scenarios.

The remainder of the paper is organized as follows. Section II covers the background of truth inference methods and attack models in crowdsourcing systems. Section III formally defines robustness in truth inference and describe the attack methodology. Section IV describes the design of our robust mechanism. Section V presents the experimental result of the proposed method. Open problems and conclusion of the paper is discussed in Section VI.

II. Related Work

In this section, we briefly review various truth inference methods and data poisoning attacks in crowdsourcing systems.

Truth Inference Methods. A key component of the crowdsourcing system is the truth inference method. There exist various approaches to infer the truth of tasks [6], [7], [11], [15], [17], [18], [23], [24], [34], [37], [37], [39], [42]–[45]. These approaches can be categorized into direct computing [15], optimization [15], [23], [24], [43], [45], probabilistic graphical model (PGM) [6], [7], [17], [18], [34], [37], and neural network based [11], [39]. A recent experimental study compared different truth inference methods [42]. There are also other approaches using matrix factorization for truth inference, [44] but they've failed to outperform the state-of-he-art inference methods.

Data Poisoning Attacks in Crowdsourcing. Data poisoning attacks [10], [12], [15], [26], [27], [37] have been recently studied against representative truth inference methods that can majorly be categorized as heuristic based methods and optimization based method.

The heuristic based attack scenarios [12], [15], [37] randomly or uniformly provide an answer for each task which is a rather naive strategy. This simple attack strategy can be applied without malicious workers having access to any further knowledge, such as the inference method or answer provided by other workers. Also, the study [33] proposes a smarter design for a heuristic based attack in which attacker are obfuscating their behavior by providing the true answer by probability p and the wrong answer by probability $1 - p$.

The optimization based attack [27] scenarios are studied on DS [6] and PM [24] inference method. These studies assume attackers have full knowledge of other workers' answers and the inference method being used. They formulate an optimization problem and assume the adversary does not know the ground truth of the tasks, hence the optimization goal is to maximize the number of flipped labels after the attack as compared to inferred labels before the attack. The attack also attempts to maximize the attackers' collective confusion matrix parameters (reliability) inferred by the system. Intuitively, this will help them to obfuscate their malicious nature and hence be more successful in misleading the system.

The comparative study [33] evaluated truth inference methods under "adversarial" settings where workers intentionally or strategically manipulate the answers. They concluded that optimization based attack is more effective compared to heuristic based attack, especially for untargeted attacks. However, for targeted attacks, in a lower percentage of adversaries, heuristic based attack can be more successful compared to optimization based attack.

Related Attacks. Data poisoning attacks on machine learning (ML) algorithms have drawn increasing interest recently [4], [16], [30], [32], [36]. However, data poisoning attack in ML and crowdsourcing differ from each other in four ways: (1) the attacks in ML mostly deal with supervised models and the goal is to degrade the performance of the model on a validation dataset, but crowdsourcing is formed as an unsupervised problem, (2) to carry out the attacks in ML, a certain number or percentage of records are poisoned by an

attacker, and also all the features associated with the poisoned record (e.g. an image) can be altered, but in crowdsourcing, only a certain number or percentage of workers answers can become malicious, and (3) ML problems typically have a rich set of features for each record while in crowdsourcing for each task only a set of ratings from workers is available.

Other related attacks include spammer [8], [28], [37] and sybil [13], [22], [35], [41] attacks. In a spammer attack, workers (bots) randomly submit answers to tasks. In sybil attacks, infiltrators create fake identities to affect the performance of the system. Sybil and spammer attacks mainly focus on the system infiltration as part of the attack and can be considered as means to achieve a data poisoning attack. The data poisoning attacks we consider in this paper assume adversaries have successfully created or compromised multiple workers and injected strategic answers.

III. Problem Definition and Attack Setup

In this section, we define robustness on the truth inference method in the crowdsourcing system and provide a high-level overview of attack settings in a crowdsourcing system.

Problem Definition. Given a set of tasks $\mathbf{T}$ and a pool of workers $\mathbf{W}$, each task $t \in \mathbf{T}$ is assigned to a subset of workers $w \in \mathbf{W}$. Each worker w_i provides an answer to each of their assigned tasks. The goal of truth inference is to determine the true answer $\hat{\mathbf{Z}}$ based on all the answers provided by the workers for each task. The tasks in a crowdsourcing system can be classified into 1) decision-making tasks where workers select a binary answer such as yes or no, 2) multi label tasks where workers select one label among multiple candidate labels, and 3) numeric tasks where workers provide answers with numeric values. The truth inference methods may consider different factors such as type of tasks, level of difficulty of tasks, and task assignment methods [42]. In this paper, we focus on the decision-making tasks, i.e. the binary truth inference problem, and do not consider other variations.

Definition 1: (Truth Inference) Given a set of tasks $\mathbf{T}$, set of workers $\mathbf{W}$ and a bipartite graph indicating tasks assigned to each worker, a truth inference method returns a set of predicted true label for tasks, denoted as $\hat{\mathbf{Z}}$.

In an adversarial environment, a certain percentage of attackers may behave maliciously and strategically attempt to flip the true label of tasks. The goal of robust truth inference is to effectively infer the truth through correct estimation of the label even in the presence of malicious workers.

Attack Setup. Based on the adversary's level of knowledge, attacks can be classified into black-box, gray-box, and white-box attacks. In black-box attacks, the adversaries only knows about their assigned tasks. In white-box attacks, the adversary has full knowledge about the inference method being used, the task assignment and answered provided by other workers. In gray-box attacks, the adversary may have partial knowledge of the above. The two attack methodologies based on different levels of adversarial knowledge are described as follow.

Heuristics Based Attacks. We adapt the heuristics based attacks in a black-box setting when the malicious workers do not know each other, as a non-collusive strategy. The simplest heuristic for an attacker would be to always report the wrong answer for each of their assigned tasks. However, this may be easily recognized by most of the truth inference systems (besides majority voting) which model the workers' reliability. The attackers could disguise themselves as honest workers by providing true answers for some tasks so that they won't be detected by the system. To model this behavior, we use the following heuristics approach [33].

The worker's behavior is modeled by defining a confusion matrix π^w that captures a worker's probability of providing a certain label given the true label. Given the label set $L = \{0, 1\}$, α and β indicate the probability of workers provide a correct label given the true label of 1 and 0, respectively. Each malicious worker w' is associated with a malicious confusion matrix $\pi^{w'}$ with α' and β' and a normal confusion matrix π^w with α and β, and a disguise parameter γ. w' behaves as a normal worker modeled by π^w with probability γ and as a malicious worker modeled by $\pi^{w'}$ with probability $1-\gamma$. This parameter helps attackers to obfuscate their behavior and deceive the system into not detecting them as malicious workers. For example, a malicious worker with a moderate disguise may have $\pi^w = \begin{bmatrix} \beta = 0.85 & 0.15 \\ 0.05 & \alpha = 0.95 \end{bmatrix}$ and $\pi^{w'} = \begin{bmatrix} \beta' = 0.05 & 0.95 \\ 0.9 & \alpha' = 0.1 \end{bmatrix}$, and γ = 0.2.

Optimization Based Attacks. We adopt the optimization based attack [26], [33] in a white box setting where an adversary has full knowledge of the truth inference algorithm being used, answers provided by other workers, and task assignments, therefore, can optimally inject manipulated answers to maximize the damage to the system. The attack goal is to maximize the number of flipped labels before and after the attack along with maximizing the attackers' collective confusion matrix parameters (reliability) inferred by the system. Intuitively, this will help them to obfuscate their malicious nature and cause more disruption in the system. Let $\hat{z^a_{t_j}}$ and $\hat{z^b_{t_j}}$ denote the inferred answer by the DS method after and before attack for task t_j, respectively, and $\hat{\alpha}_{w'}$ and $\hat{\beta}_{w'}$ represent the inferred confusion matrix parameters of the malicious worker w'. The optimization problem can be formulated as follows:

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

where λ controls the trade-off between the objectives of maximizing the inferred collective reliability of malicious workers and maximizing the number of flipped labels.

IV. Defense Methodology

In this section, we propose two solutions: 1) Edge-NN and 2) Edge-PGM that benefits from data augmentation technique followed by the enhanced inference method. Figure 1 depicts the overall framework of our solution.

Boundary Task based Data Augmentation. Intuitively, malicious workers can gain power in a crowdsourcing system and

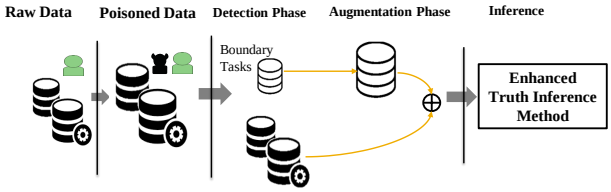


Fig. 1: Overall Framework of EdgeInfer Solution

Tasks	C (answers and assigned tasks)						P($\hat{z}$ = 0 |C)	P($\hat{z}$ = 1 |C)
	W_1	W_2	W_3	W_4	W_5	W_6		
t_1	1	0	1	0	1	0	50%	50%
t_2	0	1	1	1	1	1	17%	83%
t_3	1	0	0	1	1	1	34%	66%
t_4	0	0	0	0	0	0	100%	0%

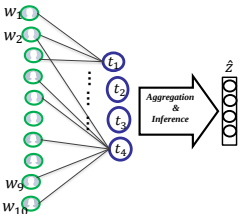


Fig. 2: Example of a Crowdsourcing System

manipulate the result of predicted labels for some but not all tasks, by preventing normal workers to reach consensus which normally reflects the true label of the task. These boundary tasks for which workers fail to reach a strong consensus are particularly vulnerable to manipulations and may lead to wrong inferences. Hence our proposed approach focuses on these boundary tasks. We present two phases here, 1) detection phase in which vulnerable tasks (i.e. boundary tasks) are identified; and 2) augmentation phase in which matrix completion technique is utilized to nullify the misleading engineered answers of malicious workers.

Detection Phase. The vulnerable boundary tasks for which workers do not reach a strong consensus are detected.

Definition 2: (Boundary task) Given a set of **M** tasks and a set of **N** workers as $t \in \mathbf{T} = \{t_1, ..., t_m\}$ and $w \in \mathbf{W} = \{w_1, ...w_n\}$, a label set $\mathbf{L} = \{0, 1\}$ and answer matrix $\mathbf{C}_{N,M}$, the subset of tasks are called boundary tasks if the certainty in workers' majority label is less than or equal to a threshold δ.

$$\mathbf{BT} = \{t : \ t \in T, \ \max\ (p_t^0, \ p_t^1) \ <= \delta\} \tag{2}$$

where p_t^0 and p_t^1 are the probability of the truth label of task t to be 0 and 1, respectively.

Figure 2 shows a crowdsourcing system consisting of 10 workers and 4 tasks where each task is answered by 6 workers. For predicting the true label of task t_2 and task t_4, workers strongly agree on label 1 and label 0 for task t_1 and task t_4, respectively. The certainty in workers' majority label is 83% and 100% for t_2 and t_4, which are quite high, so the malicious workers might not have enough power to flip the true label of these tasks. Therefore applying data augmentation for these tasks would be unnecessary and may even introduce noise. However, workers tie on the labels for task t_1 and have a weak consensus on the labels for task t_3. Therefore, malicious workers would potentially have a much greater chance of flipping the true label of these tasks, we call these tasks with less certainty for the majority label as boundary tasks. Since the inferred label of boundary tasks is more likely to be inaccurate, we run the matrix completion on boundary tasks and concatenate the completed matrix to the non-boundary tasks. Then, the truth inference method is run on this augmented answer matrix.

Augmentation Phase. In a comprehensive comparison of truth inference methods, Zheng et. al. [42] point out that MV outperform other inference methods in case of complete answer set, i.e all workers provide answers to all tasks. Based on this conclusion, having a complete answer can be advantageous in improving the accuracy of a crowdsourcing system. Furthermore, study [33] shows that redundancy, i.e. the average number of workers assigned per task, is an important factor in resilience against attacks in crowdsourcing.

However, real world data are typically sparse, meaning the number of answers per task is quite low, even though some workers can answer a large number of tasks. Motivated by this, we propose a data augmentation method to address the sparsity challenge and enhance the robustness of existing truth inference methods. The augmentation phase is only applied to boundary tasks as a pre-processing step before the inference to neutralize the potential contaminated data of adversaries.

Matrix Factorization is a common technique in data compression and feature learning [21], [31]. Using this technique for matrix completion shows the underlying interactions between workers, tasks, their corresponding level of worker reliability and task difficulty. This technique factorizes a matrix to find two matrices such that their product would generate the original matrix.

Definition 3: (Matrix Factorization): Given a set of **N** workers and a set of **M** of tasks. Let **C** of size $|N| \times |M|$ be the matrix that contains all the answers that the workers provided to the tasks. Find two matrices $\mathbf{P}_{|N|,|K|}$ and $\mathbf{Q}_{|M|,|K|}$ such that $P.Q^T$ approximate **C**.

Each row of **P** and **Q** draws the association of workers and tasks with features. To obtain **P** and **Q**, gradient descent is used to minimize the difference between their product and the answer set C, iteratively. To avoid overfitting, a parameter η is used to control the magnitude of the workers-feature and task-feature vectors such that **P** and **Q** produce an accurate approximation of **C** without having the elements of these matrices to be unnecessarily large.

$$\min \quad (\mathbf{C}_{n,m} - \sum_{k=1}^{K} p_{n,k} \ q_{k,m})^2 + \frac{\eta}{2} \sum_{k=1}^{K} (\|P\|^2 + \|Q\|^2) \tag{3}$$

Our augmentation approach is generalizable and agnostic of the inference method and it is applied as a processing step before inference to increase the redundancy of answers. There are studies based on tensor completion [12], [44] that use matrix completion as an inference method under a non-adversarial setting and apply it on the entire provided-answer matrix and their empirical results are generally not as impressive as the other state-of-the-art methods, therefor, we exclude them from our analysis.

Enhanced Inference Method. Inference involves aggregating provided answers to estimate the true label of each task. The details of inference methods in Edge-NN and Edge-PGM approaches are described here.

Edge-NN Inference. The recent survey paper [42] compares the traditional inference methods and conclude that DS is one of the overall winners. Also, this study shows [33] that LAA-S as the latest neural network based method outperforms previous methods. Here we present a neural network based approach enhanced with a stronger prior to achieve more robustness. The existing state-of-the-art neural network based approach LAA-S [39] adopts a variational autoencoder (VAE) network [19] to leverage the learned latent truth label distribution that best represents the original task vector for inference. However, LAA-S can also be subjective to poisoning attacks as it is employing the distribution of the original answers as a prior (essentially majority voting). Our main idea is to enhance it with a stronger prior that considers workers' reliability. We propose a hybrid model that utilizes the neural network model while incorporating the DS result as its prior. Since DS models the reliability of workers as a confusion matrix it outperforms majority voting in terms of approximating the true labels of the tasks. Therefore, utilizing the distribution labels of DS as the prior might result in some unsophisticated attackers being filtered out from the system.

Figure 3 shows our proposed enhanced inference method composed of two parts: 1) prior estimation that approximates the distribution of labels using the inferred truth of the tasks from DS inference method and 2) inference through the enhanced LAA-S algorithm, while leveraging the prior obtained in the previous step.

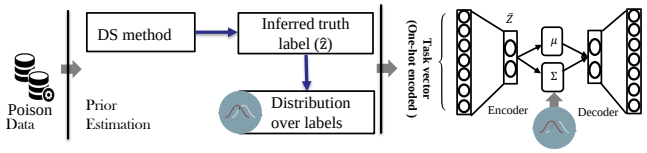


Fig. 3: Components of Inference Method in Edge-NN

DS method [6] is a PGM based method that models the reliability of each workers with a confusion matrix. It utilize the EM algorithm to calculate the maximum likelihood and estimate the item's true label and worker's reliability (i.e. their confusion matrix).

LAA-S [39] inference algorithm adopts a VAE network [19] to leverage the learned latent truth label distribution that best represent the original task vector. This model contains two shallow neural networks: 1) an encoder or classifier (q_θ) transforming the task vector (v) into the latent feature (z) indicating the truth label, and 2) a decoder or reconstructor (p_ϕ) which recovers a task vector based on the latent features.

Inputs and output of the VAE represent each task as a vector consisting of the one-hot encoding of the provided answers by each worker to the task. The network is trained on all one-hot encoded task vectors (**v**) by minimizing the reconstruction error between the original and recovered task vectors. Additionally, the training also considers how closely the learned estimated ground truth resemble those inferred by DS, supplied to the model as the prior. The objective function is shown below.

$$\min_{q_\theta, p_\theta} \mathbb{E}_{q_\theta(z|v)} log p_\phi(v|z) - D_{KL}(q_\theta(z|v)||\ prior) \qquad (4)$$

where the first term is the reconstruction error and the second term enforces the distribution of inferred labels to follow a specific prior through the negative KL divergence D_{KL} term.

Edge-PGM Inference. GLAD inference method [37] is a PGM based model that considers workers' reliability and the difficulty level of task to estimate the true answer of tasks. One drawback of GLAD method mentioned in Zheng et al. study [42] is that in some cases the estimation of the difficulty level parameter related to the tasks is inaccurate. Hence, the GLAD fails to outperform the other inference methods.

We adopt the GLAD truth inference method which incorporates the level of difficulty of tasks into their inference method by replacing the vanilla prior of task difficulty level with a better prior based on boundary tasks [37]. Our key insight is that the more difficult tasks are the one that workers fail to reach a strong consensus on and therefore are particularly vulnerable to manipulations and may lead to wrong inference. Figure 4 depicts the Components of Inference Method in Edge-PGM.

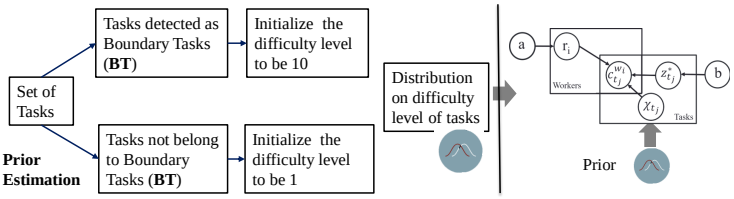


Fig. 4: Components of Inference Method in Edge-PGM

GLAD models each client i reliability as a single value $r_i \in [0, +\infty)$, a higher value implies a higher reliability, and each task's j difficulty level as a single value $1/\chi_{t_j} \in [0, +\infty)$. A high value of $1/\chi_{t_j}$ implies task t_j is more sensitive or difficult. The true label of tasks ($\hat{Z}$) are estimated based on workers' reliability (r) and tasks' difficulty (χ) parameters, as follow:

$$p(z_{t_j}|C, r, \chi) = \prod_{i \in \mathbf{W}^{t_j}} \frac{1}{1 + e^{-r_{w_i} \chi_{t_j}}} \qquad (5)$$

The Edge-PGM inference method initializes the difficulty level (χ) of detected boundary tasks as 10 and the other tasks are all treated equally and their difficulty level is set to 1, as follow:

$\{1/\chi_{t_j} = 10\, if\, t_j \in \mathbf{BT}\, else\, 1/\chi_{t_j} = 1\ for\, t_j \in \mathbf{T}\}$

Then the expectation–maximization (EM) algorithm is iteratively applied to estimate parameters (r,χ).

V. Experiments and Results

Experiment Setup. In this section, we introduce the datasets and conduct experiments on them to evaluate the performance of the proposed robust mechanism.

Datasets. We tested our method on three public benchmark datasets for decision making tasks. Table I shows the summary of the datasets and their key properties.

- Product Dataset. This dataset includes 8315 tasks where each task is a question about a comparison of two

TABLE I: Statistics and Properties of Datasets

Dataset	Product	PosSent	Temp
N (# of tasks)	8,315	1,000	462
M (# of workers)	176	85	76
V (# of answers)	24,945	20,000	4620
Redundancy (# of answers per task)	3	20	10
Engagement (# of answers per worker)	141	235	60
Avg workers' credibility	0.79	0.798	0.73
Truth Labels Ratio (negative,positive)	(88%, 12%)	(52.8%, 47.2%)	(50%, 50%)

products. An example is "are iPad Two 16GB WiFi White and iPad 2nd generation 16GB WiFi White the same?" [14].

- PosSent Dataset. This dataset contains information about the general sentiment of a tweet about the reputation of a company. Workers assess each tweet and provide positive label, meaning that the tweet will increase the reputation of the company, or negative to indicate the opposite [2].
- Temp Dataset. In this dataset, each task is to identify whether or not one event happened before another in a given context. [1]. An example news text is "John fell. Sam pushed him.", and the task is to decide if the events that the colored words describe happened before or after each other. "pushed" happened before "fell".

Poisoning Dataset. For poisoning answers, assuming adversaries W', with the fraction of malicious workers being $\frac{|W'|}{|W|+|W'|}$, we applied heuristics based (Black Box) and optimization based (White Box) attacks given the attack strategies described in Section III. Heuristic based attacks are designed based on black-box knowledge and the disguise parameter (γ) is set to 0.0. Optimization based attacks are designed based on white-box knowledge and use DS as a inference methods in which worker's reliability is modeled based on confusion matrix, we set the λ parameter to be 1.

Parameters. In Edge-NN method, for data augmentation phase, the learning rate for Temp, PosSent and Product dataset are set as [0.01, 0.01, 0.001], respectively. Also the number of latent dimensions for each of the Temp, PosSent and Product dataset are set to [13, 13, 20], respectively. The regularization parameter(η) is set to 0.005. In Edge-PGM method, the range for δ parameter is limited to [50%-70%] since other values are just complementary to this interval. The δ in Edge-PGM is chosen as 0.55 for all the experiments.

Metrics and Comparison. We evaluate inference performance using accuracy and F-score, computed based on the predicted labels and ground truth. Since the Product dataset is heavily unbalanced, F-score is chosen as the metric. For the other two datasets, we report accuracy. Accuracy is defined as the fraction of tasks whose truth are inferred correctly.

Experiment Results. We conduct several experiments with three real datasets to assess the effectivness of our solutions.

Effect of Certainty Threshold (δ) on Accuracy/F-score. Figure 5 shows the effect of the parameter δ (certainty threshold) on accuracy/f-score of the proposed method, Edge-NN, for different percentage of malicious workers. We run this experiment on three datasets, and contaminated these datasets based on heuristic and optimization attacks described in Section III.

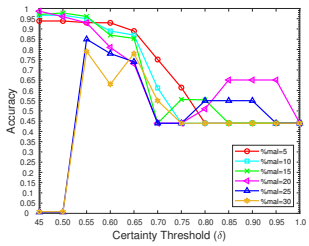
The δ= 1 corresponds to augmenting all tasks and δ= 0.45 corresponds to no augmentation. When the % of malicious workers (%mal) is low, having no augmentation performs the best and the accuracy drops when augmentation increases. This trend is as expected since the data has high quality and augmentation does not help. However, as the %mal increases, accuracy increases as augmentation increases and then drops back when augmenting all tasks. This verifies the benefit of augmenting the boundary tasks only when there is significant noise in the data.

As show in Figure 5, we observe that the accuracy of the system is sensitive to the value of noise added for augmentation on boundary tasks. Note that the number of candidate tasks is directly proportional to the certainty threshold (δ). Furthermore, it is shown that the optimal δ for an effective defense is dependent on %mal. Intuitively, at a higher %mal, there will probably be more contaminated tasks, and so by choosing a higher δ, we will apply augmentation on more tasks. We replicated this experiment on the PGM based methods and observed similar trends, which stresses that the success of boundary task augmentation is not bound to a specific inference method. This trend remains the same in both heuristic and optimization attack.

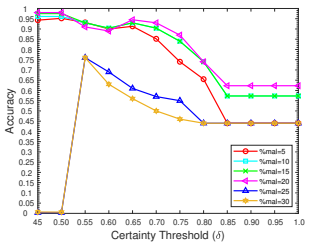
For the remaining comparison, we choose to set δ as 0.55, 0.55, 0.65 for Temp, PosSent and Product dataset for heuristic and optimization attack, respectively. In general, the range of [0.5, 0.65], corresponding to the boundary tasks, gives a good overall performance over varying percentage of malicious workers. This also confirms our intuition that it is beneficial to only perform data augmentation on the boundary tasks. We note that in practice, the percentage of malicious workers will not be too high due to the cost of creating sybil workers. Given that Product dataset is comparatively more sparse it would be harder to reach a strong consensus with fewer labels for each task. Therefore, to correctly infer the truth of those boundary tasks, more of them should be included in matrix completion, hence a higher $\delta = 0.65$ is selected.

Ablation Study. We assess the impact of augmentation as a preprocessing step and utilizing an enhanced prior on the three truth inference methods, DS, GLAD, and LAA-S (i.e. majority voting as prior). The methods that just consider the edge augmentation are called in the form of *Edge-$method-name$*. The methods with enhanced prior are named in the form of *$method-name$+*. Also, the methods that combine both of these techniques are named as *Edge-$method-name$+*. For example, Edge-DS is a DS inference method with edge augmentation, LAA-S+ is a LAA-S method with enhanced prior (i.e. DS as prior) and Edge-LAA-S+ is a LAA-S method with enhanced prior along with edge augmentation.

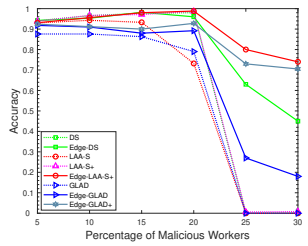
First, we assess the impact of augmentation, as shown in Figure 6, in all three datasets, Edge-GLAD, Edge-DS and Edge-LAA-S+ outperform GLAD, DS and LAA-S, especially in the most likely range for %mal, from 20 to 30. This confirms the benefit of the data augmentation technique in enhancing the performance of the existing truth inference methods.



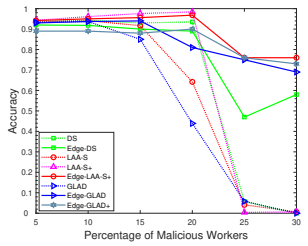
Temp Dataset: Heuristic



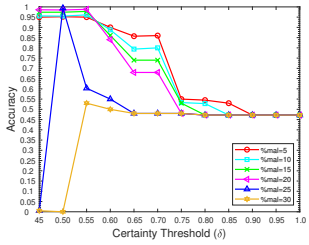
Temp Dataset: Optimization



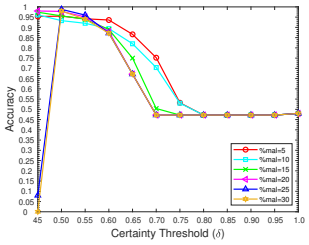
Temp Dataset: Heuristic



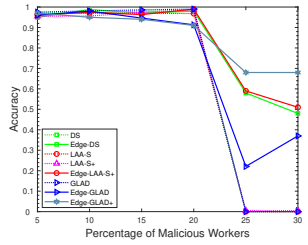
Temp Dataset: Optimization



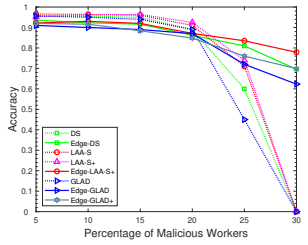
PosSent Dataset: Heuristic



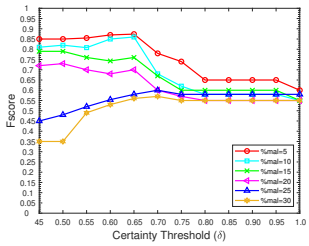
PosSent Dataset: Optimization



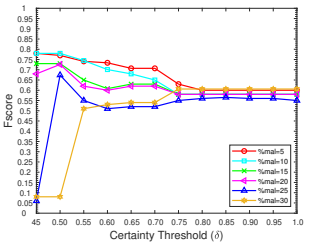
PosSent Dataset: Heuristic



PosSent Dataset: Optimization

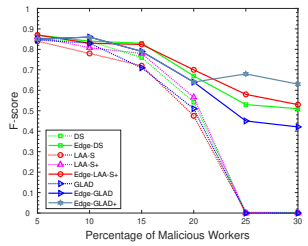


Product Dataset: Heuristic

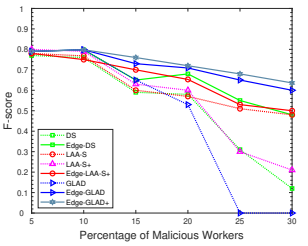


Product Dataset: Optimization

Fig. 5: The Effect of Certainty Threshold (δ) on Accuracy across different %mal



Product Dataset: Heuristic



Product Dataset: Optimization

Fig. 6: Ablation Study: Robustness vs $\%mal$

Moreover, we assess the impact of using DS as prior on the LAA-S method instead of majority voting. As it is shown in Figure 6, using the DS as a prior (LAA-S+) helps to slightly improve the performance of the model, however, if the number of malicious workers is higher than 20%, the DS model completely failed, therefore using DS could not outperform the original LAA-S model. Edge-LAA-S+ (i.e. Edge-NN) outperforms other methods and verifies the benefit of both augmentation as a preprocessing phase and using DS as prior.

Furthermore, we assess the impact of using different distribution for boundary tasks versus other tasks as prior on the GLAD method instead of using uniform distribution for all tasks. As it is shown in Figure 6, differentiating the difficulty level of tasks and using it as a prior (Edge-GLAD+) helps to improve the performance of the model.

Comparison of Edge-NN and Edge-PGM. Figure 6 shows that Edge-NN (i.e. Edge-LAA-S+) and Edge-PGM (i.e. Edge-GLAD+) are robust against both type of attacks (i.e. black-box and white-box) and reduce the attack's success rate.

Edge-NN performs better than Edge-PGM when the %mal is less than 20%, since the performance of Edge-NN depends on the prior and DS method at that specific interval performs well. By increasing the number of malicious workers in the system the DS method is unable to perform well and it effects the performance of Edge-NN method. However, Edge-PGM outperform Edge-NN when the %mal is greater than 20%.

VI. Conclusion & Future Work

We proposed two solutions Edge-NN and Edge-PGM to improve the robustness of existing truth inference methods against data poisoning attacks in crowdsourcing systems. The proposed solutions provide a novel algorithm that applies matrix completion on a subset of tasks in which workers cannot reach a sufficiently strong consensus. In addition, it is combined with two enhancements to existing state-of-the-art inference methods by utilizing prior information. For the evaluation of our work, we applied a heuristic based attack and optimization based attack and our results confirm the effectiveness of our defense solution.

Our future work includes studies on other types of attacks such as targeted attacks and further improvement on robustness especially against strong white-box attacks. Furthermore, we plan to study automatic selection of δ by analyzing the distance between the distribution of estimated ground truth obtained from a weak truth inference method (e.g. MV) and a strong truth inference method (e.g. DS) as a proxy for the percentage of malicious workers in the system.

REFERENCES

[1] TemporalOrdering. https://sites.google.com/site/nlpannotations/.

[2] Twitter Sentiment. https://raw.githubusercontent.com/zfz/twitter_corpus/master/full-corpus.csv.

[3] Michael Anderson and Jeremy Magruder. Learning from the crowd: Regression discontinuity estimates of the effects of an online review database. *The Economic Journal*, 122(563):957–989, 2012.

[4] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[5] Hongkyu Choi, Kyumin Lee, and Steve Webb. Detecting malicious campaigns in crowdsourcing platforms. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 197–202. IEEE Press, 2016.

[6] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[7] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*.

[8] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *CrowdSearch*.

[9] John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[10] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*.

[11] Alex Gaunt, Diana Borsa, and Yoram Bachrach. Training deep neural nets to aggregate crowdsourced responses. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence. AUAI Press*, page 242251, 2016.

[12] Arpita Ghosh, Satyen Kale, and Preston McAfee. Who moderates the moderators? crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 167–176, 2011.

[13] Nam Ky Giang, Michael Blackstock, Rodger Lea, and Victor C M Leung. Distributed data flow: A programming model for the crowdsourced internet of things. In *Proceedings of the Doctoral Symposium of the 16th International Middleware Conference*, pages 1–4, 2015.

[14] M. J. Franklin J. Wang, T. Kraska and J. Feng. Crowderd. Crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.

[15] Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Advances in Neural Information Processing Systems*.

[16] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

[17] David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, pages 1953–1961, 2011.

[18] Hyun-Chul Kim and Zoubin Ghahramani. Bayesian classifier combination. In *Artificial Intelligence and Statistics*, pages 619–627, 2012.

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[20] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

[21] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[22] Brian Neil Levine, Clay Shields, and N Boris Margolin. A survey of solutions to the sybil attack. *University of Massachusetts Amherst, Amherst, MA*, 7:224, 2006.

[23] 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. *Proceedings of the VLDB Endowment*, 8(4).

[24] 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 *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*.

[25] Michael Luca and Georgios Zervas. Fake it till you make it: Reputation, competition, and yelp review fraud. *Management Science*, 62(12).

[26] 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 *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 13–22. International World Wide Web Conferences Steering Committee, 2018.

[27] Chenglin Miao, Qi Li, Houping Xiao, Wenjun Jiang, Mengdi Huai, and Lu Su. Towards data poisoning attacks in crowd sensing systems. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 111–120. ACM, 2018.

[28] Vikas C Raykar and Shipeng Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13(Feb):491–518, 2012.

[29] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr).

[30] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6106–6116, 2018.

[31] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2005.

[32] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. When does machine learning fail? generalized transferability for evasion and poisoning attacks. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1299–1316, 2018.

[33] Farnaz Tahmasebian, Li Xiong, Mani Sotoodeh, and Vaidy Sunderam. Crowdsourcing under data poisoning attacks: A comparative study. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 310–332. Springer, 2020.

[34] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164. ACM, 2014.

[35] Gang Wang, Bolun Wang, Tianyi Wang, Ana Nika, Haitao Zheng, and Ben Y Zhao. Defending against sybil devices in crowdsourced mapping services. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*.

[36] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *USENIX Security Symposium*, pages 239–254, 2014.

[37] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[38] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y Zhao, and Yafei Dai. Uncovering social network sybils in the wild. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):2, 2014.

[39] Li'ang Yin, Jianhua Han, Weinan Zhang, and Yong Yu. Aggregating crowd wisdoms with label-aware autoencoders. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1325–1331. AAAI Press, 2017.

[40] Dong Yuan, Guoliang Li, Qi Li, and Yudian Zheng. Sybil defense in crowdsourcing platforms. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1529–1538. ACM, 2017.

[41] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal*, 1(5):372–383, 2014.

[42] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, 2017.

[43] Dengyong Zhou, Sumit Basu, Yi Mao, and John C Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in neural information processing systems*, pages 2195–2203, 2012.

[44] Yao Zhou and Jingrui He. Crowdsourcing via tensor augmentation and completion. In *IJCAI*, pages 2435–2441, 2016.

[45] Yao Zhou, Lei Ying, and Jingrui He. Multic2: an optimization framework for learning from task and worker dual heterogeneity. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 579–587. SIAM, 2017.