Black-Box Data Poisoning Attacks on Crowdsourcing

Pengpeng Chen[1,2,5], **Yongqiang Yang**[2,4], **Dingqi Yang**[3], **Hailong Sun**[*2,4], **Zhijun Chen**[2,4], **Peng Lin**[1,5]
[1]China's Aviation System Engineering Research Institute, Beijing, China
[2]SKLSDE Lab, Beihang University, Beijing, China
[3]State Key Laboratory of Internet of Things for Smart City and Department of Computer and Information Science, University of Macau, Macau SAR, China
[4]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China
[5]Chinese Aeronautical Establishment, Beijing, China
cplwting@163.com, yangyongqiang@buaa.edu.cn, dingqiyang@um.edu.mo, {sunhl, zhijunchen}@buaa.edu.cn, caebuaa@163.com

Abstract

Understanding the vulnerability of label aggregation against data poisoning attacks is key to ensuring data quality in crowdsourced label collection. State-of-the-art attack mechanisms generally assume full knowledge of the aggregation models while failing to consider the flexibility of malicious workers in selecting which instances to label. Such a setup limits the applicability of the attack mechanisms and impedes further improvement of their success rate. This paper introduces a black-box data poisoning attack framework that finds the optimal strategies for instance selection and labeling to attack unknown label aggregation models in crowdsourcing. We formulate the attack problem on top of a generic formalization of label aggregation models and then introduce a substitution approach that attacks a substitute aggregation model in replacement of the unknown model. Through extensive validation on multiple real-world datasets, we demonstrate the effectiveness of both instance selection and model substitution in improving the success rate of attacks.

1 Introduction

Crowdsourcing provides a cost-effective means to collect labeled data for machine learning tasks by engaging human workers in the form of an open call [Doan *et al.*, 2011; Wang and Zhou, 2016; Sheng and Zhang, 2019; Fang *et al.*, 2018]. While allowing the participation of a large group of workers, the openness of crowdsourcing brings opportunities for adversarial parties to launch data poisoning attacks for data sabotage purposes [Miao *et al.*, 2018a]. Understanding and assessing the vulnerability of crowdsourcing against the data poisoning attacks is essential to ensure data quality in crowdsourced label collection.

*Corresponding author

Previous studies on adversarial attacks generally assume simple attack strategies by malicious workers such as submitting random labels or labels that disagree with those submitted by normal workers [Gadiraju *et al.*, 2015; KhudaBukhsh *et al.*, 2014b; Yuan *et al.*, 2017]. Such malicious behaviors can be easily detected by label aggregation models designed to capture worker reliability, e.g., Dawid-Skene [Dawid and Skene, 1979], ZenCrowd [Demartini *et al.*, 2012]. In those models, malicious workers whose labels disagree with the normal workers—who generally constitute the majority—are considered to be of low reliability; consequently, labels from malicious workers will be assigned with low weights in label aggregation, thus generating limited impact on the aggregation result. Recently, Miao et al. [2018a] introduce an intelligent data poisoning mechanism that disguises the attacking behaviors from the Dawid-Skene model. In their work, data poisoning attack is formulated as an optimization problem where the objective is to maximize the error of aggregated labels, while improving the reliability estimate for malicious workers by guiding them to agree with normal workers on (preassigned) instances whose labels are unlikely to be overturned. Such an idea has been extended to attacking other aggregation models (e.g., the Gaussian truth model (GTM) and the conflict resolution on heterogeneous data (CRH) [Fang *et al.*, 2021; Miao *et al.*, 2018b]).

Those attack mechanisms, however, have only considered the crowdsourcing setting where malicious workers are randomly assigned data instances for labeling and their only room for action is determining the labels. In real-world settings, malicious workers can often *actively select* which instances to label [Jagabathula *et al.*, 2017; Wang *et al.*, 2014; Tran *et al.*, 2009; Molavi Kakhki *et al.*, 2013]. Such a higher degree of flexibility offers an opportunity to improve the success rate of the attack with less cost: malicious workers can strategically select and label instances for which normal workers provide divergent labels, thereby disguising malicious behavior while easily turn over the majority label. Another important limitation of the existing attack mechanism

is that it is designed specifically for the Dawid-Skene model with the assumption of full knowledge about the model, i.e., model parameters are known to the attacker. In real-world scenarios, the specific label aggregation model used by crowdsourcing platforms can be a black box, rendering such attack mechanisms inapplicable.

In this work, we introduce SubPac, a substitution-based approach for black-box data poisoning attack on crowdsourcing with unknown label aggregation models. We first introduce a unified representation of label aggregation that covers a broad family of aggregation models. Building on top of that, we then formulate the data poisoning attack problem as an optimization problem, where the objective is to find the best strategy for attacking the unified model under a specific budget constraint of the attacker. In modeling the attack strategy, we consider the general crowdsourcing setting that allows the selection of instances to be labeled by malicious workers in addition to the determination of the labels themselves. From the computational perspective, the new formulation results in a bilevel *min-max* optimization problem where the outer problem is to find the optimal attack strategies for both instance selection and labeling and the inner problem is to optimize label aggregation. The problem is generally considered as NP-hard; in our specific formulation, it is further complicated by the mixture of both continuous and discrete variables that represent the instance selection and labeling strategies. To handle such an issue, we introduce a dual gradient-descent algorithm with a reparameterization trick that converts discrete variables into continuous ones and then learns the parameters of the aggregation model and those of the attack strategy.

Our approach allows to attack a chosen aggregation model in substitution for the unknown black-box model in crowdsourcing. To select the substitute model, we use a success rate metric that quantifies the transferability between the substitute model and the target, unknown aggregation model. Through empirical experiments, we study the transferability between a set of widely-used aggregation models and provide guidelines for choosing the substitute model.

In summary, we make the following key contributions:

- We propose SubPac, a substitution-based approach for data poisoning attacks on crowdsourcing with unknown label aggregation models;
- We introduce a cost-effective attack strategy considering both instance selection and labeling and derive an algorithm with a reparameterization trick for learning the optimal strategy;
- We conduct an extensive evaluation on four real-world datasets and show that SubPac substantially improves the state of the art under the same budget.

To the best of our knowledge, we are the first to consider data poisoning attacks under the black-box assumption of aggregation models in crowdsourcing. Our empirical results show both the instance selection and substitution are effective methods to improve the success rate of attacks. In particular, we find that probabilistic models such as Dawid-Skene (DS) [Dawid and Skene, 1979] and ZenCrowd [Demartini *et al.*, 2012] are effective substitutes due to their higher transferability for other models.

2 Related Work

2.1 Label Aggregation in Crowdsourcing

Label aggregation aims to infer the true label of each instance by aggregating its labels from multiple workers [Chen *et al.*, 2022b; Jiang *et al.*, 2022; Chen *et al.*, 2022a]. The simplest voting model, majority voting (MV) [Sheng *et al.*, 2008], derives the majority labels by counting the workers' votes for each alternative label. Due to the ignorance of varying reliability among workers, MV is error-prone. In contrast, WMV [Li and Yu, 2014] and CATD [Li *et al.*, 2014] assigns different weights to workers' votes considering workers' abilities. Besides, a major type of label aggregation models leverage probabilistic modeling to estimate worker reliability. DS [Dawid and Skene, 1979] models each worker's reliability with a confusion matrix and uses the EM algorithm to iteratively updates the true label of each instance and the workers' confusion matrices. ZC [Demartini *et al.*, 2012] is a simplified version of DS: it does not consider the priors and models each worker's reliability with a single probability of correct labeling. There also exist some other models that can be viewed as the extensions of ZC, e.g., GLAD [Whitehill *et al.*, 2009], KOS [Karger *et al.*, 2011], VI-BP [Liu *et al.*, 2012] and the extensions of WMV [Chen *et al.*, 2022b].

The impact of poisoning attacks on those different aggregation models has not been compared or systematically analyzed. Our work, to the best of our knowledge, is the first to introduce a generic attacking strategy for a wide range of aggregation models and to present an analysis of attack performance across those models.

2.2 Data Poisoning Attacks on Machine Learning

Understanding adversarial attacks[Dong *et al.*, 2019a; Dong *et al.*, 2019b; Dai *et al.*, 2018; Mei and Zhu, 2015a; Li *et al.*, 2016; Checco *et al.*, 2020] helps to develop more robust machine learning system, which is essential for achieving trustworthy artificial intelligence. Data poisoning [Wang *et al.*, 2014; Biggio *et al.*, 2012; Mei and Zhu, 2015b; KhudaBukhsh *et al.*, 2014a; Zhao *et al.*, 2017; Ma *et al.*, 2019; Zhang *et al.*, 2019; Liu and Shroff, 2019] has been used to analyze the vulnerabilities of many popular machine learning technologies, e.g. SVMs [Biggio *et al.*, 2012], regression learning [Jagielski *et al.*, 2018] and multi-task learning [Zhao *et al.*, 2018]. In a poisoning attack, the attacker attempts to affect or even dominate the final trained model by manipulating *the feature values or annotations* of training instances [Chen *et al.*, 2020; Chen *et al.*, 2021; Chen *et al.*, 2018]. Due to the heterogeneous characteristics of crowdsourcing data, most of the existing approaches cannot be directly applied to attack label aggregation models of crowdsourcing. They are designed for the setting in which the label of an instance is from a single reliable expert. However, in the setting of crowdsourcing, each instance is labeled by multiple workers with varying levels of reliability.

Our work focus on the analysis of the vulnerability of crowdsourcing by designing optimal poisoning attacks. We show that the attacker can instigate an effective attack on crowdsourcing even when only a small number of malicious workers are present.

3 Problem Formalization

We formally define our problem of finding the optimal data poisoning attack strategy as follows.

Notations. We use capital letters (e.g. $\mathcal{A}$) in calligraphic math font to denote sets. We use boldface uppercase letters to denote matrices, e.g., $\mathbf{M}$, in which the entry (i, j) is denoted by the corresponding lowercase letters m_{ij} and the entries of i-th row is denoted by $\mathbf{M}_{i*}$. We use boldface lowercase letters to denote vectors (e.g., $\mathbf{v}$). Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \ldots\}$ be the instance set, $\mathcal{U} = \{u_1, u_2, \ldots, u_j, \ldots\}$ be the normal worker set, $\Omega = \{l_1, l_2, \ldots, l_k, \ldots\}$ be the set of possible labels, and $\mathbf{Y} = (y_{ij})_{|\mathcal{X}|\times|\mathcal{U}|}$ be the set of normal labels, where y_{ij} is the label from u_j to $\mathbf{x}_i$. Let $\mathbf{T} = (t_{ij})_{|\mathcal{X}|\times|\mathcal{U}|}$ be the indicator matrix where $t_{ij} = 1$ indicates that u_j provides a label to instance $\mathbf{x}_i$, and 0 otherwise. Similarly, we denote the malicious worker set, their label matrix, and their indicator matrix by $\tilde{\mathcal{U}} = \{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_{j'}, \ldots\}$, $\tilde{\mathbf{Y}} = (\tilde{y}_{ij'})_{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}$, and $\tilde{\mathbf{T}} = (\tilde{t}_{ij'})_{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}$. For each instance $\mathbf{x}_i$, there is an *unobserved* ground truth z_i to be estimated by the label aggregation model $f' : \Omega^{|\mathcal{U}|+|\tilde{\mathcal{U}}|} \to \Omega$ from the labels of crowd workers $\mathcal{U}$ and $\tilde{\mathcal{U}}$.

Problem statement. Given a budget $B = \sum_i \sum_{j'} \tilde{t}_{ij'}$ and a victim label aggregation model f', the attacker attempts to find an optimal attack strategy for the instance selection $\tilde{\mathbf{T}}$ and labeling $\tilde{\mathbf{Y}}$ to subvert a maximum number of the aggregated labels while disguising the malicious behavior. In our setting, an additional complexity is that the attacker does not know the label aggregation model f' used by the crowdsourcing platform and the attacker uses a substitute model f for the estimate of the aggregated labels.

4 Unified Representation of Label Aggregation Models

To allow for black-box data poisoning attacks, we first provide a unified representation for the label aggregation models. We start with the simple yet common setting of binary labeling which does not sacrifice the generality of our approach. We will extend the proposed method to the multi-option setting of the labeling task later. For the sake of convenience, we introduce the following notions.

$$\mathbf{v}^{(ij)} = \frac{1}{2} t_{ij} \left(1 - y_{ij}, 1 + y_{ij}\right), \tag{1}$$

where vector $\mathbf{v}^{(ij)}$ is constructed by one-hot encoding for each label from worker u_j to $\mathbf{x}_i$: $\mathbf{v}^{(ij)} = (0, 1)$ when worker u_j provides $+1$ to $\mathbf{x}_i$; $\mathbf{v}^{(ij)} = (1, 0)$ when worker u_j provides -1 to $\mathbf{x}_i$; and $\mathbf{v}^{(ij)} = (0, 0)$ when worker u_j does not provide label to $\mathbf{x}_i$.

$$\mathbf{W}^{(ij)} = \begin{pmatrix} w^{(ij)}_{-1,-1} & w^{(ij)}_{-1,+1} \\ w^{(ij)}_{+1,-1} & w^{(ij)}_{+1,+1} \end{pmatrix}, \tag{2}$$

$w^{(ij)}_{kh}$ denotes the weight of class l_k on $\mathbf{x}_i$, when $y_{ij} = l_h$ and $l_h \in \Omega$.

Without loss of generality, we define the unified label aggregation model f as follows.

$$f\left(\mathbf{Y}_{i*}\right) = \mathrm{argmax}_{l_k} \bar{w}^{(i)}_k, \tag{3}$$

where

$$\bar{w}^{(i)}_k = \sum\nolimits_j \mathbf{v}^{(ij)} (\mathbf{W}^{(ij)}_{k*})^{\mathrm{T}} + w^*_k, \tag{4}$$

$\bar{w}^i_k$ denotes the weight of class l_k on $\mathbf{x}_i$, when we observe $\mathbf{Y}_{i*}$, and w^*_k is a shift constant.

We show in the following theorems that Equation 3 is a universal representation of the label aggregation models. First, for the Dawid-Skene model, we have Theorem 1.

Theorem 1. *let $\mathbf{P}^{(j)}$ denote the confusion matrix and π^*_k denote the prior of class l_k, f is equivalent to the Dawid-Skene model when $\mathbf{W}^{(ij)} = \ln \mathbf{P}^{(j)}$ and $w^*_k = \ln \pi^*_k$.*

Another widely used aggregation model, ZenCrowd [Demartini *et al.*, 2012], can be viewed as the homogeneous version of the DS model: it characterizes the worker ability with the symmetric confusion matrix. For ZenCrowd, we have Corollary 1 based on Theorem 1.

Corollary 1. *f is equivalent to ZenCrowd, when $w^*_k = 0$ and*

$$\mathbf{W}^{(ij)} = \begin{pmatrix} \ln(p^*_j) & \ln(1 - p^*_j) \\ \ln(1 - p^*_j) & \ln(p^*_j) \end{pmatrix}, \tag{5}$$

where p^*_j denotes the reliability parameters of workers.

Finally, we have the following theorems for majority voting and weighted majority voting [1].

Theorem 2. *When $\mathbf{W}^{(ij)} = \mathbf{I}$ and $w^*_k = 0$, f is equivalent to majority voting, where $\mathbf{I}$ is the identity matrix.*

Theorem 3. *When $\mathbf{W}^{(ij)} = d_j \mathbf{I}$ and $w^*_k = 0$, f is equivalent to weighted majority voting, where d_j denotes the weight of u_j who provides a label to instance $\mathbf{x}_i$.*

5 Data Poisoning Attacks on Crowdsourcing

In this section, we first formulate the problem of data poisoning attack on the unified label aggregation model, and then present our substitution approach that attacks a substitute model in replacement of the unknown target model. We first generalize Equation (3) to support the modeling of an adversarial environment.

As for normal workers, we introduce $\tilde{\mathbf{v}}^{(ij')}$ and $\tilde{\mathbf{W}}^{(ij')}$ for malicious workers. The label aggregation model f is defined as follows.

$$f(\mathbf{Y}'_{i*}) = \mathrm{argmax}_{l_k} \hat{w}^{(i)}_k, \tag{6}$$

where

$$\hat{w}^{(i)}_k = \bar{w}^{(i)}_k + \sum\nolimits_{j'} \tilde{\mathbf{v}}^{(ij')} (\tilde{\mathbf{W}}^{(ij')}_{k*})^{\mathrm{T}} + w^*_k, \tag{7}$$

$\hat{w}^{(i)}_k$ denotes the weights attached to class l_k, when the crowd labels $\mathbf{Y}'_{i*}$ from $\mathcal{U}$ and $\tilde{\mathcal{U}}$ are provided to $\mathbf{x}_i$.

[1] Refer to the supplementary material for proofs of Theorems, which is available at https://github.com/yongqiangyang/SubPac.

5.1 Optimal Adversarial Strategy

We now formulate the problem of finding the optimal attack strategy for attacking f, which represents a broad family of substitute models. The strategy gives the indication of both instance selection (i.e., which instances to select) and labeling (i.e., how to label) under a limited budget. To formally define the problem, we translate the goal of subverting a maximum number of the aggregated labels while disguising the malicious behavior to the following objectives: minimizing the similarity of the estimated true labels before and after attacks, meanwhile minimizing the discrepancy of malicious labels and the estimated true labels before attacks. Considering these two objectives together, we formulate the problem of finding optimal adversarial strategy as a bilevel *min-max* optimization problem.

$$\min_{\tilde{\mathbf{Y}},\tilde{\mathbf{T}}} L = d_1 + d_2 \tag{8}$$

$$s.t.\ f(\mathbf{Y}'_{i*}) = \max_{l_k} \hat{w}_k^{(i)},$$
$$\sum\nolimits_i \sum\nolimits_{j'} \tilde{t}_{ij'} = B, \tilde{\mathbf{T}} \in \{0,1\}^{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}, \tag{9}$$

where L denotes the loss function, which contains two components. $d_1 = -\frac{1}{|\mathcal{X}|}\sum_i v\left(f\left(\mathbf{Y}'_{i*}\right), f\left(\mathbf{Y}_{i*}\right)\right)$, represents the average similarity between the aggregated labels before and after attacks, where $v(p,q)$ measures the discrepancy between p and q, such as cross entropy. $d_2 = \frac{\lambda}{|\tilde{\mathcal{U}}|}\sum_{j'} \frac{\sum_i \tilde{t}_{ij'} v\left(\tilde{y}_{ij'}, f(\mathbf{Y}_{i*})\right)}{\sum_i \tilde{t}_{ij'}}$, represents the discrepancy between the malicious worker's label $\tilde{y}_{ij'}$ and the estimated true label $f\left(\mathbf{Y}_{i*}\right)$ before attacks.

In this optimization problem, the label aggregation model is a constraint, making the problem a bilevel *min-max* problem. Here specifically, the outer problem is relatively straightforward to optimize, while the inner optimization problem is highly non-linear and non-convex, for which a closed-form solution is hard to obtain. The constraint that the summation of all the elements in $\tilde{\mathbf{T}}$ is limited by the budget B, further complicates the optimization problem.

5.2 Computing Optimal Attack Strategies

We first provide a theoretical solution for the problem defined in Equation 8. Then, based on the solution, we present our algorithm to obtain the optimal attack strategy with substitute models.

Gradient Computation

We first construct the Lagrangian Ψ of the outer problem of Equation 8. The following optimization problem needs to be solved in order to achieve the adversarial aim.

$$\min_{\tilde{\mathbf{Y}},\tilde{\mathbf{T}}} \Psi = L + \psi(\sum\nolimits_i \sum\nolimits_{j'} \tilde{t}_{ij'} - B)$$
$$s.t.\ f(\mathbf{Y}'_{i*}) = \max_{l_k} \hat{w}_k^{(i)}$$
$$\tilde{\mathbf{T}} \in \{0,1\}^{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}, \tag{10}$$

where ψ is the Lagrangian multiplier. Since the elements of $\tilde{\mathbf{Y}} \in \{-1,1\}^{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}$ and $\tilde{\mathbf{T}} \in \{0,1\}^{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}$ are discrete, we cannot directly compute the gradients of them. To address the problem, we introduce a reparameterization trick that converts the discrete variables into continuous ones such that it can be updated in a gradient-based optimization algorithm.

We first relax each element in $\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{T}}$ into $[0,1]$. In other words, $\tilde{y}_{ij'}$ denotes the probability that worker $u_{j'}$ provides class 1 to instance $\mathbf{x}_i$ and $\tilde{t}_{ij'}$ denotes the probability that worker $u_{j'}$ provides a label to instance $\mathbf{x}_i$. To account for the constraints $\tilde{\mathbf{Y}} \in [0,1]^{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}$ and $\tilde{\mathbf{T}} \in [0,1]^{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}$, we posit that $\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{T}}$ are derived from their *ancestral matrices*.

$$\tilde{\mathbf{Y}} = \text{sigmoid}(\tilde{\mathbf{Y}}'), \tag{11}$$

$$\tilde{\mathbf{T}} = \text{sigmoid}(\tilde{\mathbf{T}}'), \tag{12}$$

where $\tilde{\mathbf{Y}}' = (\tilde{y}'_{ij'})_{|\mathcal{X}|\times|\tilde{\mathcal{U}}|} \in \mathbb{R}^{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}$ and $\tilde{\mathbf{T}}' = (\tilde{t}'_{ij'})_{|\mathcal{X}|\times|\tilde{\mathcal{U}}|} \in \mathbb{R}^{|\mathcal{X}|\times|\tilde{\mathcal{U}}|}$. With reparameterization, we can compute the gradients of Ψ with respect to $\tilde{y'}_{ij'}$ and $\tilde{t'}_{ij'}$. Note that $\tilde{y}'_{ij'} \in (-\infty, +\infty)$ and $\tilde{t}'_{ij'} \in (-\infty, +\infty)$ which can be updated in a gradient-based optimization algorithm.

Attacks Using Substitution

To attack crowdsourcing systems with unknown label aggregation models, we first introduce our algorithm based on the dual gradient descent approach that can attack a family of substitute models in replacement of the unknown targeted model. Then, we use a success rate metric that quantifies the transferability between the substitute and the targeted model, which enables us to find a good substitute model.

To learn the optimal strategy, we iteratively update the parameters of the substitute models and those of the attack strategy.

Phase 1. Fixing the Lagrange multiplier ψ computed in the previous iteration, this phase is responsible for updating the attack strategy $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{Y}}$, which contains two iterative steps. *Step 1*: it is responsible for estimating the parameters $\tilde{\mathbf{W}}^{(ij')}$ and $\mathbf{W}^{(ij)}$ by inputting the $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{Y}}$ generated in step 2 into the substitutes. If the substitute model is majority voting (MV), it directly obtains that $\tilde{\mathbf{W}}^{(ij)} = \mathbf{W}^{(ij)} = \mathbf{I}$. *Step 2.*Fixing the parameters, we adopt the gradient descent method to update $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{Y}}$. In iteration t, we update $\tilde{y}'_{ij'}$ and $\tilde{t}'_{ij'}$ as follows.

$$\tilde{y}'^{(t+1)}_{ij'} \leftarrow \tilde{y}'^{(t)}_{ij'} - \eta \nabla_{\tilde{y}'_{ij'}} \Psi, \tag{13}$$

$$\tilde{t}'^{(t+1)}_{ij'} \leftarrow \tilde{t}'^{(t)}_{ij'} - \eta' \nabla_{\tilde{t}'_{ij'}} \Psi, \tag{14}$$

where η and η' are the step size.

Phase 2. Fixing $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{Y}}$ computed in Phase 1, this phase involves computing the Lagrange multiplier ψ as follows. In iteration r, we update ψ as follows.

$$\psi^{(r+1)} \leftarrow \psi^{(r)} + \eta'' \nabla_{\psi} \Psi, \tag{15}$$

where $\nabla_{\psi}\Psi$ is the gradient of Ψ with respect to ψ and η'' is the step size.

Algorithm 1: SubPac

Input: The budget B, the number of malicious workers $|\tilde{\mathcal{U}}|$
Output: Optimal attack strategy concerning instance selection $\tilde{\mathbf{T}}$ and labeling $\tilde{\mathbf{Y}}$

Initialize $\tilde{\mathbf{Y}}$, and $\tilde{\mathbf{T}}$;
while *the Lagrangian multiplier* ψ *does not converge* **do**
  while *the change of* $\tilde{\mathbf{T}}$ *or* $\tilde{\mathbf{Y}} >$ *tolerance* **do**
    Update the parameters of the substitute label aggregation model;
    for *each* $\tilde{t}'_{ij'}$ **do**
      Update $\tilde{t}'_{ij'}$ with Equation 14;
      Update $\tilde{t}_{ij'}$ with Equation 12;
      $\tilde{t}_{ij'} = \frac{1}{2}(1 + \text{sign}(\tilde{t}_{ij'} - 1/2))$;
    for *each* $\tilde{y}'_{ij'}$ **do**
      Update $\tilde{y}'_{ij'}$ with Equation 13;
      Update $\tilde{y}_{ij'}$ with Equation 11;
      $\tilde{y}_{ij'} = \text{sign}(\tilde{y}_{ij'} - 1/2)$;
  Update the Lagrangian multiplier ψ with Equation 15
return *Optimal attack strategy* $\tilde{\mathbf{T}}$ *and* $\tilde{\mathbf{Y}}$*;*

We summarize the procedure of the proposed attack in Algorithm 1, which computes the optimal attack strategy against a broad family of substitutes for the unknown target label aggregation model.

We use transferability to measure the effectiveness of substitution-based attacks on the target model and find a good substitute model that possesses high transferability. We adopt the attack success rate [Dong *et al.*, 2019b] of substitution-based attacks on the target model for the measurement of attack transferability. We will analyze four substitutes on eight target models in the experiments and find good substitutes based on transferability analysis.

6 Experiments

This section presents our experimental results for evaluating the effectiveness of the attack strategy obtained by our proposed approach SubPac [2]. Specifically, we answer the following questions:

- **Q1**: How well does the attack strategy perform in subverting the label aggregation result with varying proportion of malicious labels?
- **Q2**: How well does the attack strategy perform in disguising malicious behaviors in gold test?
- **Q3**: How effective is the attack strategy in attacking different target models and which are good substitute models?
- **Q4**: How effective is the proposed attack strategy perform with limited accessibility to normal labels?

6.1 Experimental Setup

Real-world datasets. We experiment with the following four real-world datasets. 1) ***Temp*** [Snow *et al.*, 2008]: labels of this dataset are provided by workers from Amazon Mechanical Turk. Annotators are presented with dialogue and verbs in it and are encouraged to identify the temporary order of given verbs. 2) ***rte*** [Snow *et al.*, 2008]: the task is to recognize textual entailment, *i.e.,* the annotator needs to determine whether a given hypothesis sentence can be inferred from another sentence. 3) ***sentiment*** [Zheng *et al.*, 2017]: in this dataset, each instance contains a review text of a company and the labels of workers reflect their opinions about the sentiment of the review. 4) ***ER*** [Wang *et al.*, 2012]: the task is entity resolution, i.e., each instance contains two products (with descriptions), the workers judge whether the two products are the same. For the 4 real-world datasets, Figure 1 presents the boxplots concerning the distribution of the number of labels per worker and the distribution of the worker accuracy among the workers.

Dataset	N	M	A	N^*	M^*
Temp	462	76	4,620	60.8	10
rte	800	164	8,000	48.8	10
sentiment	1,000	85	20,000	235.3	20
ER	8,315	176	24,945	141.7	3

Table 1: *Real-world datasets.* N is the number of instances and M is the number of normal workers. M^* is the average worker redundancy of instances. N^* is the average number of instances handled by each worker. A is the number of normal labels.

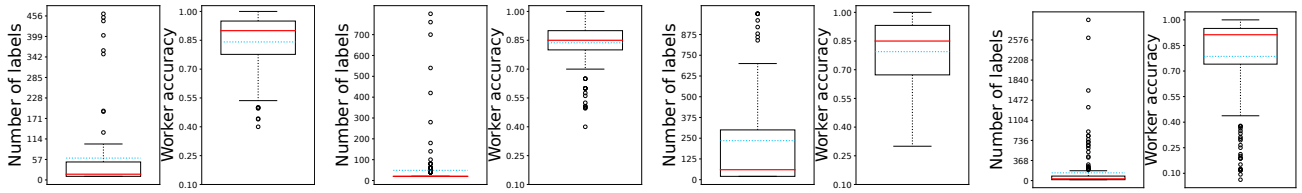
Target and substitute models. In our experiments, we consider 8 target models: Dawid and Skene (DS) [Dawid and Skene, 1979], ZenCrowd (ZC) [Demartini *et al.*, 2012], majority voting (MV) [Sheng *et al.*, 2008], weighted majority voting (WMV) [Li and Yu, 2014], GLAD [Whitehill *et al.*, 2009], KOS [Karger *et al.*, 2011], VI-BP [Liu *et al.*, 2012] and CATD [Li *et al.*, 2014]. Among them, 4 models are considered as substitute models, i.e., Dawid and Skene, ZenCrowd, majority voting, and weighted majority voting.

Comparison methods. We compare the proposed method with the following methods. 1) **TIA** [Miao *et al.*, 2018a], a state-of-the-art attacking method designed to subvert the labels inferred by Dawid and Skene model. 2) **Rand**, malicious workers provide random labels on each given item. 3) **Flip**, malicious workers indiscriminately provide bad labels on every given instance [Ipeirotis *et al.*, 2010].

Details of parameter settings. The budget B possessed by the attacker is set to $M' \times \tilde{N}$, where M' is the number of malicious workers and $\tilde{N}$ is the number of instances labeled by each malicious worker. Each element of $\tilde{\mathbf{T}}$ is initialized by 1. Each element of $\tilde{\mathbf{Y}}$ is initialized as a random option different from aggregated result. We consider the scenario where the proportion of malicious workers is very low, namely $M' \leq 5$. The attacker has a limited budget; thus we set $\tilde{N} \leq 0.5 \cdot N$, where N is the number of instances. We discuss how to set the parameter λ in the supplementary material [3] . Each gold task involves an instance randomly selected and the number of the gold tasks is set to 10 according to [Yuan *et al.*, 2017]. We use cross-entropy for the discrepancy function $v(p, q)$.

[2] Our code is available at https://github.com/yongqiangyang/SubPac.

[3] Refer to the supplementary material for more details about the experimental setup.



(a) Temp dataset (b) Rte dataset (c) Sentiment dataset (d) ER Dataset

Figure 1: Boxplots concerning the distribution of the number of labels from per worker and the distribution of the worker accuracy among the workers in Temp dataset, Rte dataset, Sentiment dataset, and ER dataset.

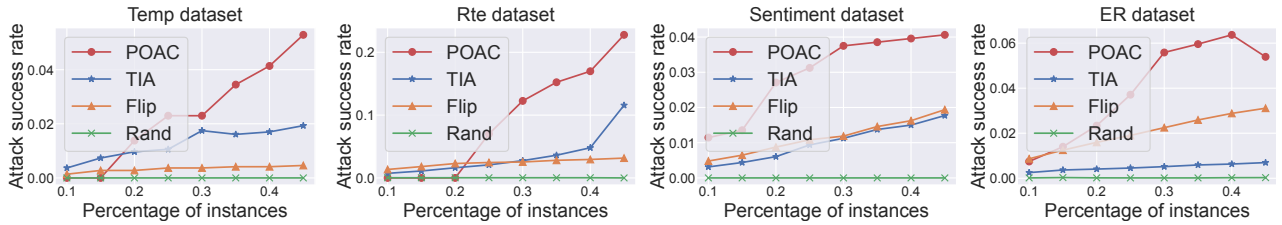


Figure 2: Attack success rate with varying proportions of the instances labeled by each malicious worker.

Evaluation metrics. The evaluation metric of experiments 1, 3, and 4 is the attack success rate [Dong *et al.*, 2019b] which is computed as the ratio between (numerator.) the number of instances whose aggregation results are correct before the attack and become incorrect after the attack, and (denominator.) the number of instances whose aggregation results are correct before the attack. The evaluation metric of experiment 2 is the ability of workers which is computed with golden tasks.

6.2 Different Attack Budget (Q1)

We consider the overall number of malicious workers to be not more than 50% of the average number of normal workers per data instance, thus making the adversarial attack a challenging problem. This gives us the number of malicious workers to be 5, 5, 5, and 1 in the four datasets.

Figure 2 compares the performance of different attacks with varying proportions of instances labeled by malicious workers, with the target model being DS. We observe that the attacks computed by the proposed method SubPac significantly outperform other strategies across all the four datasets. We further observe that the proposed method improves steadily when the proportion of instances labeled by malicious workers increases, and outperforms the baseline methods by a larger margin. This shows that when the proportion of instances labeled by malicious workers increases (the budget of the attacker increases), SubPac can consistently select vulnerable and easier-to-subvert instances, for which the normal workers provide the divergent labels. Those results clearly demonstrate the effectiveness of our proposed approach in the attack as well as the cost-efficiency.

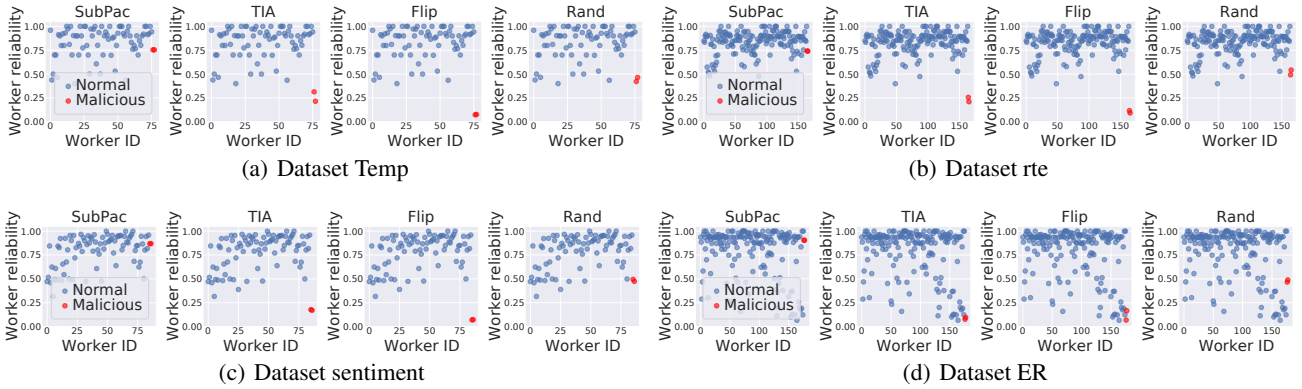
6.3 Disguising Malicious Behaviors (Q2)

We analyze the performance of our approach in disguising malicious behaviors, by showing the estimated reliability of malicious ones in the golden test. The number of malicious workers is set to 2 and the proportion of instances labeled by malicious workers is set to the average number of instances labeled by normal workers.

Results are given in Figure 3. From the figures, we observe that malicious workers generated with SubPac exhibit even higher abilities than normal participants on average, thus effectively disguising their malicious behavior. On dataset *Temp*, *rte*, *sentiment*, and *ER*, the average reliability of malicious workers manipulated by SubPac is 0.7540, 0.7410, 0.8690, and 0.9050, respectively. The average reliability of normal workers is 0.8399, 0.8351, 0.7951, and 0.7864, respectively. As a result, malicious workers can avoid their labels to be filtered out by strategically select instances and providing wrong labels. As comparison, we observe that the reliability of malicious workers following the baseline attack strategies – including the state-of-the-art method TIA – are estimated much lower than normal workers, making them easily detectable and their labels useless for attack purposes.

6.4 Transferability Analysis (Q3)

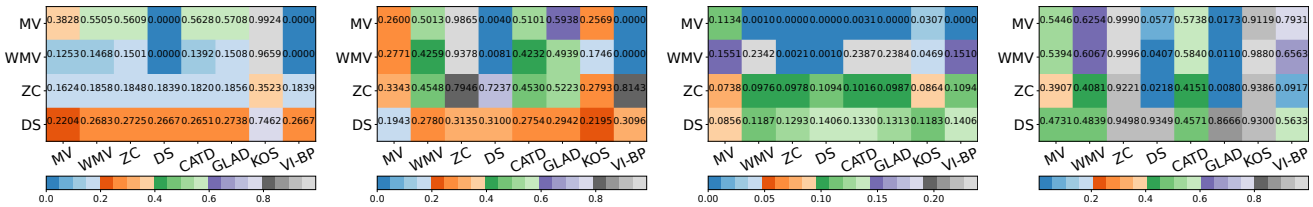
We analyze the transferability of different substitutes for finding suitable ones and demonstrate the effectiveness of substitute-based black-box attacks. We compute the attacks on the four substitute models using the proposed framework SubPac and evaluate the attack success rate of eight victim models under the calculated attacks. We set the small number of malicious workers for the four datasets (*Temp*: 6.17%, *rte*: 2.96%, *sentiment*: 5.56%, and *ER*: 2.76%), and the proportion of instances labeled by them is 0.50 in each dataset. Figure 4 shows the transferability of the four attacks on the eight victim models on the four real-world datasets. First, the attack success rate of the attack designed for probabilistic models on other victim models is comparable to that of the attack designed for the victim model, which means that the probabilistic substitutes-based attacks have good transferability. The average attack success rate of the probabilistic



(a) Dataset Temp (b) Dataset rte

(c) Dataset sentiment (d) Dataset ER

Figure 3: Reliability distribution of all the participating workers (normal or malicious) in the golden test for four datasets.



(a) Dataset Temp (b) Dataset rte (c) Dataset sentiment (d) Dataset ER

Figure 4: The transferability of attacks based on different substitutes. We compute the attacks on the four representative methods using the unified framework SubPac and evaluate the attack success rate of eight victim models under the computed attacks.

Figure 5: Attack with small proportion of known normal labels.

models is 26.25%, 41.06%, 11.08%, and 55.34% on the four datasets, respectively. Second, the variance of the attack success rate of the probabilistic models is smaller than that of others, which demonstrates the stability of substitution-based attacks. This is because Subpac is based on the generic aggregation function whose parameter matrix is constructed based on the form of a confusion matrix in DS and ZC.

6.5 Limited Accessibility to Normal Labels (Q4)

We now investigate the performance of our approach with limited accessibility to normal labels, assuming that an attacker can know the normal workers' labels to a small fraction of the instances in each dataset. Specifically, since SubPac and TIA entail analyzing the normal labels for generating malicious labels, in these two strategies, the malicious parties only annotate the instances whose normal labels they can observe. We are concerned with the following situation: malicious workers have no quantitative advantage, that is, there exist only 5, 5, 5, 1 malicious workers for the four datasets. Figure 5 shows the result with the proportion of instances labeled by malicious workers set to 0.3 and the proportion of instances accessible by malicious workers also set to 0.3. We consider the target model being the non-probabilistic model WMV and the substitute model being the probabilistic one DS, on which TIA is applicable. From the figure, we observe that SubPac consistently outperforms the other methods across all the four datasets. The attack success rate of SubPac is on average 5.7 times larger than that of TIA. The result shows that although there are only a small number of malicious workers and these workers know only a small fraction of normal workers' labels, the malicious workers as instructed by our strategy can still effectively attack those instances whose labels of honest parties are observable.

7 Conclusion

This paper has presented SubPac, a black-box data poisoning attack framework for crowdsourcing. SubPac is built on a generic formulation of label aggregation and leverages a substitution approach to attack unknown label aggregation models. It finds the optimal attack strategy by suggesting malicious both the instances to label and the labels themselves for the maximization of success rate. Extensive validation on several real-world datasets shows that SubPac is an effective attack framework that substantially outperforms the state of the art and can be applied for black-box attacks. In future work, we plan to investigate approaches to defending against data poisoning attacks via precise identification of strategic instance selection and labeling behaviors.

Acknowledgments

This work was supported partly by Aeronautical Science Foundation of China under Grant No.2022Z071020002, partly by National Natural Science Foundation of China under Grant Nos (61932007, 61972013, 62141209), partly by Science and Technology Development Fund, Macau SAR (0038/2021/AGJ and SKL-IOTSC(UM)-2021-2023).

References

[Biggio *et al.*, 2012] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of ICML*, pages 1807–1814, 2012.

[Checco *et al.*, 2020] Alessandro Checco, Jo Bates, and Gianluca Demartini. Adversarial attacks on crowdsourcing quality control. *Journal of Artificial Intelligence Research*, 67:375–408, 2020.

[Chen *et al.*, 2018] Peng-Peng Chen, Hai-Long Sun, Yi-Li Fang, and Jin-Peng Huai. Collusion-proof result inference in crowdsourcing. *Journal of Computer Science and Technology*, 33(2):351–365, 2018.

[Chen *et al.*, 2020] Pengpeng Chen, Hailong Sun, Yili Fang, and Xudong Liu. Conan: A framework for detecting and handling collusion in crowdsourcing. *Information Sciences*, 515:44–63, 2020.

[Chen *et al.*, 2021] Pengpeng Chen, Hailong Sun, and Zhijun Chen. Data poisoning attacks on crowdsourcing learning. In *Web and Big Data - 5th International Joint Conference, APWeb-WAIM 2021*, volume 12858 of *Lecture Notes in Computer Science*, pages 164–179, 2021.

[Chen *et al.*, 2022a] Pengpeng Chen, Hailong Sun, Yongqiang Yang, and Zhijun Chen. Adversarial learning from crowds. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 5304–5312, 2022.

[Chen *et al.*, 2022b] Ziqi Chen, Liangxiao Jiang, and Chaoqun Li. Label augmented and weighted majority voting for crowdsourcing. *Inf. Sci.*, 606:397–409, 2022.

[Dai *et al.*, 2018] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *ICML*, pages 1123–1132, 2018.

[Dawid and Skene, 1979] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society*, 28(1):20–28, 1979.

[Demartini *et al.*, 2012] 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*, pages 469–478, 2012.

[Doan *et al.*, 2011] Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[Dong *et al.*, 2019a] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[Dong *et al.*, 2019b] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, pages 7714–7722, 2019.

[Fang *et al.*, 2018] Yili Fang, Hailong Sun, Pengpeng Chen, and Jinpeng Huai. On the cost complexity of crowdsourcing. In *IJCAI*, pages 1531–1537, 2018.

[Fang *et al.*, 2021] Minghong Fang, Minghao Sun, Qi Li, Neil Zhenqiang Gong, Jin Tian, and Jia Liu. Data poisoning attacks and defenses to crowdsourcing systems. *arXiv preprint arXiv:2102.09171*, 2021.

[Gadiraju *et al.*, 2015] 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*, pages 1631–1640, 2015.

[Ipeirotis *et al.*, 2010] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67, 2010.

[Jagabathula *et al.*, 2017] Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Identifying unreliable and adversarial workers in crowdsourced labeling tasks. *The Journal of Machine Learning Research*, 18(1):3233–3299, 2017.

[Jagielski *et al.*, 2018] Matthew Jagielski, Alina Oprea, and Battista Biggio. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *SP*, pages 19–35, 2018.

[Jiang *et al.*, 2022] Liangxiao Jiang, Hao Zhang, Fangna Tao, and Chaoqun Li. Learning from crowds with multiple noisy label distribution propagation. *IEEE Trans. Neural Networks Learn. Syst.*, 33(11):6558–6568, 2022.

[Karger *et al.*, 2011] David Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. *Advances in neural information processing systems*, 24:1953–1961, 2011.

[KhudaBukhsh *et al.*, 2014a] Ashiqur KhudaBukhsh, Jaime Carbonell, and Peter Jansen. Detecting non-adversarial collusion in crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 2, 2014.

[KhudaBukhsh *et al.*, 2014b] Ashiqur R. KhudaBukhsh, Jaime G. Carbonell, and Peter J. Jansen. Detecting non-adversarial collusion in crowdsourcing. In *AAAI*, pages 104–111, 2014.

[Li and Yu, 2014] Hongwei Li and Bin Yu. Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*, 2014.

[Li *et al.*, 2014] 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):425–436, 2014.

[Li *et al.*, 2016] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *NIPS*, pages 1885–1893, 2016.

[Liu and Shroff, 2019] Fang Liu and Ness B. Shroff. Data poisoning attacks on stochastic bandits. In *ICML*, pages 4042–4050, 2019.

[Liu *et al.*, 2012] Qiang Liu, Jian Peng, and Alexander T Ihler. Variational inference for crowdsourcing. volume 25, pages 692–700, 2012.

[Ma *et al.*, 2019] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *IJCAI*, pages 4732–4738, 2019.

[Mei and Zhu, 2015a] Shike Mei and Xiaojin Zhu. The security of latent dirichlet allocation. In *Artificial Intelligence and Statistics*, pages 681–689, 2015.

[Mei and Zhu, 2015b] Shike Mei and Xiaojin Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI*, pages 2871–2877, 2015.

[Miao *et al.*, 2018a] 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*, pages 13–22, 2018.

[Miao *et al.*, 2018b] 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, 2018.

[Molavi Kakhki *et al.*, 2013] Arash Molavi Kakhki, Chloe Kliman-Silver, and Alan Mislove. Iolaus: Securing online content rating systems. In *WWW'13*, pages 919–930, 2013.

[Sheng and Zhang, 2019] Victor S Sheng and Jing Zhang. Machine learning with crowdsourcing: A brief summary of the past research and future directions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9837–9843, 2019.

[Sheng *et al.*, 2008] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622, 2008.

[Snow *et al.*, 2008] Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263, 2008.

[Tran *et al.*, 2009] Dinh Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In *NSDI'09*, number 1, pages 15–28, 2009.

[Wang and Zhou, 2016] Lu Wang and Zhi-Hua Zhou. Cost-saving effect of crowdsourcing learning. In *IJCAI*, pages 2111–2117, 2016.

[Wang *et al.*, 2012] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *arXiv preprint arXiv:1208.1927*, 2012.

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

[Whitehill *et al.*, 2009] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems*, 22:2035–2043, 2009.

[Yuan *et al.*, 2017] 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, 2017.

[Zhang *et al.*, 2019] Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. Data poisoning attack against knowledge graph embedding. In *IJCAI*, pages 4853–4859, 2019.

[Zhao *et al.*, 2017] Mengchen Zhao, Bo An, Wei Gao, and Teng Zhang. Efficient label contamination attacks against black-box learning models. In *IJCAI*, pages 3945–3951, 2017.

[Zhao *et al.*, 2018] Mengchen Zhao, Bo An, Yaodong Yu, Sulin Liu, and Sinno Jialin Pan. Data poisoning attacks on multi-task relationship learning. In *AAAI*, pages 2628–2635, 2018.

[Zheng *et al.*, 2017] 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.