Defending Data Poisoning Attacks in DP-Based Crowdsensing: A Game-Theoretic Approach

Zhirun Zheng, Zhetao Li, *Member, IEEE*, Cheng Huang, *Member, IEEE*, Saiqin Long, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Differential privacy (DP) is widely used for protecting privacy in crowdsensing by adding noises. However, malicious attackers can exploit noise to launch covert data poisoning attacks. In this paper, we propose a game-based defense approach to resist such data poisoning attacks in DP-based crowdsensing systems. In this approach, attackers are believed to be powerful as they can refine their attack strategy based on the observations of deployed defenders' defense strategy. Specifically, *the defenders* formulate the defense as a functional minimization problem (which cannot be directly solved by numerical optimization algorithms because its decision variable is a set of functions), resisting data poisoning attacks by deleting data shared by identified malicious workers through the log-likelihood ratio test. To obtain a current defense strategy, the decision variable of the problem is relaxed into the coefficients of basis-based linear combinations through the variable-basis approximation, and then solved using the simulated annealing genetic algorithm. Correspondingly, *the attackers* formulate their attack strategy as a bi-level maximization problem (which is an NP-hard problem), biasing crowdsensing results as much as possible while remaining undetected. Since the attackers can know the defense strategy, they may bypass the defenders by constraining the expected log-likelihood ratio test. Additionally, the attackers can evade truth discovery methods deployed in crowdsensing using DP noise. To determine a current attack strategy, the bi-level problem is decomposed into upper-level and lower-level sub-problems, wherein the upper-level sub-problem is solved by the variational methods, and then these sub-problems are alternately optimized. Finally, we propose a local minimax points calculating algorithm to obtain an equilibrium point in the defenders-attackers game, thereby finding an optimal defense strategy to resist the powerful data poisoning attack. Extensive experiments on real-world and synthetic datasets show that the proposed game-based defense approach can effectively defend powerful and covert attackers.**

Received 21 January 2024; revised 14 September 2024; accepted 23 October 2024. Date of publication 28 October 2024; date of current version 5 February 2025. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the National Natural Science Foundation of China under Grant 62032020, Grant 62076214, and Grant U23B2027, in part by the National Key Research and Development Program of China under Grant 2021YFB3101200. Recommended for acceptance by H. Shen. *(Corresponding author: Zhetao Li.)*

Zhirun Zheng is with the School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China (e-mail: zhengzhirun2019@gmail.com).

Zhetao Li and Saiqin Long are with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China (e-mail: liztchina@hotmail.com; xxgcxyxtu@sina.com).

Cheng Huang is with the School of Computer Science, Fudan University, Shanghai 200438, China (e-mail: chuang@fudan.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2024.3486689, provided by the authors.

Digital Object Identifier 10.1109/TMC.2024.3486689

***Index Terms*—Data poisoning attacks, differential privacy, privacy-preserving crowdsensing, zero-sum Stackelberg game.**

I. Introduction

CROWDSENSING has emerged as a prevalent and extensively embraced approach for gathering information. The sensors integrated into mobile devices act as workers, sensing various data including environmental monitoring and intelligent transportation [1], [2]. Specifically, a crowdsensing server accepts sensing tasks from requesters, each task with a set of objectives, and then assigns them to specific workers for collecting sensory data. In addition, each object is completed by multiple workers, alleviating mistakes arising from individual workers. Ultimately, the server aggregates the sensory data shared by all workers, providing the aggregated results back to requesters. For example, the objects might comprise some specific locations, and the requester aims to assess the noise level at each of these locations.

Nevertheless, the sensory data may encompass sensitive personal information, such as location and health information [3], [4], [5], [6], [7], [8], [9], triggering notable privacy concerns and potentially discouraging worker involvement. To mitigate these concerns, differential privacy (DP) [10] has been widely applied in crowdsensing to protect the sensory data privacy of workers [4], [11], [12], [13], [14]. Specifically, the sensory data is injected with DP noise before sharing, balancing privacy preservation with minimal degradation of data utility. As a result, the server only observes the perturbed (or noisy) sensory data, utilizing them to estimate the aggregated results.

Although privacy remains a significant concern in crowdsensing, it is regrettably not the only risk, as attackers could manipulate malicious workers to inject fabricated sensory data into crowdsensing for personal profit (a.k.a. data poisoning attacks) [15], [16], [17], [18], [19], [20]. For instance, attackers could tamper with weather forecasts by spreading fabricated extreme weather reports, such as a severe storm or heavy rainfall, leading to a negative user experience and eroding the credibility of crowdsensing systems for weather monitoring. Achieving this attack goal becomes notably formidable when truth discovery methods, a form of reliability-based aggregation method, are deployed in crowdsensing. The methods could capture the reliability of each worker by estimating the quality of sensory data they upload and then aggregating the data accordingly. Therefore, malicious workers could obtain lower reliability when they

always dissent from the majority, reducing their influence in the aggregation.

Many state-of-the-art works [15], [16], [17], [18], [19], [20], [21] have demonstrated that the truth discovery methods can perform rather poorly under intelligent data poisoning attacks. Specifically, malicious workers can uphold consensus with the majority on objects that are unlikely to undergo alterations, striving to improve their reliability and enhance their influence on other objects. To mitigate malicious workers, existing works [22], [23] focused on the crowdsensing systems without deploying DP, wherein each worker uploads sensory data directly, and have proposed various robust truth discovery methods. Furthermore, a series of studies [24], [25], [26], [27], [28], [29], [30] have found that DP could facilitate the success of data poisoning attacks despite it can provide provable privacy protection for workers. These studies focus on various data types (such as tabular, key-value, and graph data) in aggregation scenarios, ignoring the detection mechanism such as truth discovery poses challenges to data poisoning attacks. A similar study [5] has revealed that malicious workers could bypass the truth discovery methods by hiding behind the DP noise. However, research results on mitigating the malicious workers concealed within DP noise remain quite limited.

Different from existing works, we focus on the DP-based privacy-preserving crowdsensing systems, and strive to find an effective defense strategy to mitigate a data poisoning attack launched by the powerful attackers. We consider that the attackers aim to skew the outputs of truth discovery methods (i.e., the aggregated results), knowing the underlying defense strategy and the DP protocol and truth discovery method deployed in the crowdsensing. The DP noise is not identified as malicious behavior by the truth discovery methods, as it is drawn from the same distribution and does not alter the differences between the distributions of sensing data [5], [31]. As a result, the attackers could bypass the truth discovery methods by exploiting DP noise. To this end, we are facing two nontrivial challenges. The first challenge lies in *finding the effective defense strategy when the poisoning strategy remains undisclosed to the defenders.* The second challenge is *determining the powerful data poisoning attack strategy that seeks to evade detection by exploiting DP noise.* In addition, we also assume that the attackers know the raw data sensed by some particular workers.

To address the challenges, we propose a game-based defense approach, allowing the defenders to find an effective defense strategy against the data poisoning attack launched by powerful attackers. In this game, the defenders determine the defense strategy with the awareness that it will be observed by the attackers, who subsequently optimize their attack strategy based on this observation. Specifically, *the defenders (a.k.a. leaders) play first, aiming to mitigate the damage caused by malicious workers.* The defenders formulate the defense as a functional optimization problem, minimizing the damage by removing sensory data shared by malicious workers identified through the log-likelihood ratio test. The defenders could find the current defense strategy by solving this problem. *The attackers (a.k.a. followers) play next, aiming to damage the aggregated results as much as possible.* The strategy of remaining stealthy is as follows: First, the attackers could bypass the truth discovery methods by hiding behind DP noise, reducing the distance between their attack strategy and the distribution followed by the perturbed sensory data; Second, the attackers could avoid the defenders since they know the current defense strategy, decreasing the expected log-likelihood ratio test. Thus, the attacks are formulated as a bi-level optimization problem with the object of maximizing the damage to aggregated results, and the constraints are given by the proposed stealth strategy. By solving this problem, the attackers could determine the current attack strategy. As a result, this game accurately simulates the powerful attackers being aware of the defense strategy adopted by defenders and subsequently improving their attack strategy based on this knowledge.

Since both optimization problems of defenders and attackers are NP-hard, finding a global minimax point of the formulated game becomes exceptionally demanding. Hence, we give the definition and crucial lemma of local minimax points, proposing a local minimax point calculating algorithm accordingly. By this algorithm, we can determine the powerful data poisoning attack strategy while finding the corresponding defense strategy. We evaluate this game approach using both datasets, confirming its ability to find an effective defense strategy against the powerful data poisoning attack in the DP-based privacy-preserving crowdsensing systems. For instance, on the Emotion dataset, the attack performance (as measured by the attack gain defined in Definition 6) decreases 181.9455 (or 94.5856) in the Laplace (or Gaussian) mechanism when $\varepsilon = 0.1$. The main contributions of this work are summarized as follows:

- We develop a game-theoretic defense approach for the DP-based privacy-preserving crowdsensing, finding an effective defense strategy against the powerful data poisoning attack. In particular, we formulate the defenders-attackers interaction as a zero-sum Stackelberg game, wherein the attackers can improve their attack strategy according to the current defense strategy, proposing a local minimax point calculating algorithm to find an equilibrium point in this game.
- Defenders formulate their defense as a functional optimization problem, minimizing the damage caused by malicious workers. Since the decision variable of this problem is a set of functions and cannot be directly solved, we relax the defense problem from optimizing over functions to optimizing over the coefficients of basis-based linear combinations through the variable-basis approximation. Then, we adopt the simulated annealing genetic algorithm to solve the relaxed problem, obtaining a current defense strategy.
- Attackers formulate the powerful data poisoning attacks as a bi-level optimization problem, maximizing the damage to the outputs of truth discovery methods. This problem is divided into upper-level and lower-level sub-problems, wherein the upper-level sub-problem is analytically solved by the variational methods, determining the current attack strategy by alternately optimizing them.

The remainder of this paper is organized as follows. We review related works in Section II. Section III introduces the system

model and two basic concepts including differential privacy and truth discovery methods. We give the problem statement in Section IV. After that, the problem of finding an effective defense strategy against the powerful data poisoning attack is formulated as a zero-sum Stackelberg game in Section V. We develop a Stackelberg-game-based defense approach in Section VI, and evaluate this approach in Section VII. Finally, we conclude this paper in Section VIII.

II. Related Work

Data poisoning attacks, also commonly referred to as false data injection attacks, involve the deliberate manipulation of data in order to corrupt computational results derived from that data. These attacks can significantly impact the reliability of data-driven systems, such as crowdsensing. Next, we review the data poisoning attacks and defenses to crowdsensing and differential privacy.

A. Data Poisoning Attacks and Defenses in Crowdsensing

The truth discovery methods, a category of aggregation techniques extensively utilized in crowdsensing systems for extracting accurate information, could mitigate conflicting data, but well-crafted malicious inputs can still bypass them [5], [15], [16], [17], [18], [19], [20]. For example, focusing on categorical (or discrete) data, Miao et al. [19] proposed intelligent data poisoning attacks to crowdsensing with conflict resolution on heterogeneous (CRH, a type of truth discovery methods), and Miao et al. [16] studied data poisoning attacks against crowdsensing empowered with Dawid-Skene model (a type of truth discovery methods). Additionally, focusing on continuous data, Fang et al. [15] proposed optimization-based data poisoning attacks against crowdsensing empowered with CRH and Gaussian truth model (a type of truth discovery methods), and Zhang et al. [17] studied multi-round data poisoning attacks to crowdsensing with TruthFinder (a type of truth discovery methods). In these attacks, malicious workers can masquerade as normal workers, tricking truth discovery methods. Different from those works, Li et al. [5] initially proposed disguised-based data poisoning attacks against differentially private crowdsensing. They employed an innovative stealth strategy that leverages DP perturbation to conceal malicious attacks, aiming to evade truth discovery methods.

In recent years, robust truth discovery methods [22], [23] are proposed to mitigate data poisoning attacks. For instance, Huang et al. [22] investigated the data poisoning attacks on crowdsensing with truth discovery methods, and proposed a robust approach to counter such attacks. This robust approach integrates the reliability evaluation and worker filtering processes into CRH, where the reliability evaluation process estimates the reliability of workers, and then the worker filtering process removes unreliable workers. Zhang et al. [23] studied robust truth discovery methods against multi-round data poisoning attacks in crowdsensing, including the detection of malicious workers. In addition, Fang et al. [15] aims to arm the crowdsensing with malicious workers detection capability, and proposed two defenses to reduce the impact of malicious workers, i.e., the median-of-weighted-average defense and the maximize influence of estimation defense.

B. Data Poisoning Attacks and Defenses to Differential Privacy Protocols

Recent years have witnessed a remarkable surge in the field of data poisoning attacks against differential privacy, highlighting the escalating attention and momentum in this critical research domain [24], [25], [26], [27], [28], [29], [30]. For example, Giraldo et al. [24] found optimal data poisoning attacks to mislead a classifier that detects anomalies, disguising false data into normal data by exploiting DP noise. Cao et al. [25] focused on local differential privacy (LDP) protocols for frequency estimation and heavy hitter identification, which are two fundamental data analytics tasks. They put forth targeted data poisoning attacks with the objective of manipulating these LDP protocols to inaccurately estimate high frequencies for attacker-selected items or falsely identify them as heavy hitters. A concurrent and independent work [26] studied untargeted data poisoning attacks to noninteractive LDP protocols, which aim to degrade the overall performance of these protocols. Following these works, Wu et al. [27] directed their attention toward LDP protocols for key-value data, and proposed novel targeted data poisoning attacks. In such LDP protocols, an aggregator aims to simultaneously estimate the frequency and mean value of each key among the collected key-value data, where the key-value data is a potentially heterogeneous data type (i.e., keys are categorical and values are numerical). Particularly, they formulated the attacks as a two-objective optimization problem, aiming to simultaneously maximize the frequencies and mean values of some attacker-chosen target keys. Li et al. [28] focused on LDP protocols for mean and variance estimation, and proposed a fine-grained poisoning attack with the goals of fine-tuning and simultaneously manipulating mean and variance estimations. Imola et al. [30] observed LDP makes graph degree estimation protocols more vulnerable to poisoning attacks. *These studies strongly advocate exercising caution when considering the deployment of differential privacy measures: differential privacy exerts a detrimental influence on systems, rendering them inherently vulnerable to data poisoning attacks.* In addition, Imola et al. [30] focused on differentially private graph analysis, and designed robust degree estimation protocols under LDP that can significantly reduce the threats caused by LDP noise.

III. Preliminaries

A. System Model

A typical DP-based privacy-preserving crowdsensing system consists of a *semi-honest crowdsensing server*, some *requesters*, and some *normal and malicious workers*. As shown in Fig. 1, we consider the server to be semi-honest, providing high-quality services but with a potential for curiosity towards workers' private information. Specifically, the requesters submit sensing tasks, such as air quality monitoring, to the server. Then, the server assigns the tasks to a group of participating workers, and asks them to utilize sensors embedded in their mobile

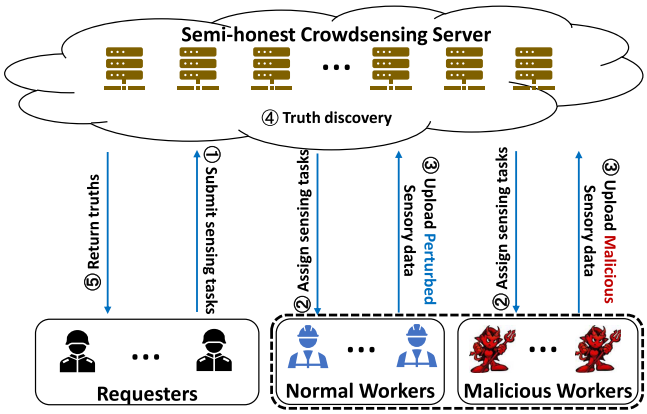


Fig. 1. System model.

devices to accomplish the tasks and upload sensory data. To protect sensory data privacy, the normal workers will add noise (or perturbation) to real sensory data by a DP protocol before sharing. Furthermore, the malicious workers can launch data poisoning attacks, wherein they directly upload carefully crafted sensory data (a.k.a. malicious sensory data) to the server with the intention of disrupting the crowdsensing for malicious purposes. After collecting all sensory data, the server infers truths from the collected data by performing the truth discovery method, and then returns the truths (i.e., aggregated results outputted by the truth discovery method) to requesters.

Let each task contain objects $O = \{o_1, \ldots, o_{|O|}\}$ (o_n is n-th object) and be completed by workers $U = \{u_1, \ldots, u_{|U|}\}$ (u_m denotes m-th worker), where the workers U are composed of normal workers $\widehat{U} = \{\widehat{u}_1, \ldots, \widehat{u}_{|\widehat{U}|}\}$ ($\widehat{u}_m$ stands for m-th normal worker) and malicious workers $\widetilde{U} = \{\widetilde{u}_1, \ldots, \widetilde{u}_{|\widetilde{U}|}\}$ ($\widetilde{u}_m$ represents m-th malicious worker). Then, the sensory data $Y = \{y_n^m \mid \forall o_n \in O, \forall u_m \in U\}$ (y_n^m is the sensory data uploaded by m-th worker for n-th object) shared by all workers U is composed of the perturbed sensory data $\widehat{Y} = \{\widehat{y}_n^m \mid \forall o_n \in O, \forall \widehat{u}_m \in \widehat{U}\}$ ($\widehat{y}_n^m$ is the perturbed sensory data uploaded by m-th normal worker for n-th object, and its real version is $\widehat{x}_n^m$) and the malicious sensory data $\widetilde{X} = \{\widetilde{x}_n^m \mid \forall o_n \in O, \forall \widetilde{u}_m \in \widetilde{U}\}$ ($\widetilde{x}_n^m$ denotes the malicious sensory data uploaded by m-th malicious worker for n-th object). Finally, the server infers truths $X^{truth} = \{x_n^{truth} \mid \forall o_n \in O\}$ (x_n^{truth} is the aggregated result of n-th object) from the collected data $Y = \widehat{Y} \cup \widetilde{X}$ by performing weight-based truth discovery methods. Besides, the credibility of workers (or the quality of the sensory data uploaded by workers) is denoted as worker weights W, and its detailed definition is given below.

Definition 1 (Worker Weights): Let $U = \{u_1, \ldots, u_{|U|}\}$ be the set of all workers, the weights set of all workers are defined as

$$W = \{w_m \mid \forall u_m \in U\}, \tag{1}$$

where w_m is the weight of m-th worker. Particularly, a lower value of w_m indicates lower quality in the sensory data uploaded by user u_m.

In addition, the weights W are composed of normal worker weights $\widehat{W} = \{\widehat{w}_m \mid \forall \widehat{u}_m \in \widehat{U}\}$ ($\widehat{w}_m$ represents the weight of m-th normal worker) and malicious worker weights $\widetilde{W} = \{\widetilde{w}_m \mid \forall \widetilde{u}_m \in \widetilde{U}\}$ ($\widetilde{w}_m$ denotes the weight of m-th malicious worker). Table I summarizes the key notations in our paper.

B. Differential Privacy

To prevent workers' private information from leaking, a type of perturbation-based privacy protection mechanism, such as DP [32], is widely adopted in crowdsensing systems. Particularly, DP provides provable privacy protection while not overly degrading data quality, and is defined as follows.

Definition 2 (Differential Privacy): Let $\mathcal{R}$ be the universal set of perturbation mechanism $\mathcal{A}$'s outputs. Given any subset $\mathcal{S} \subset \mathcal{R}$, and any two different sensory data $\widehat{x}^{(1)}$ and $\widehat{x}^{(2)}$, a randomized perturbation mechanism $\mathcal{A}$ is said to be (ε, δ)-differential privacy ((ε, δ)-DP) if and only if

$$Pr(\mathcal{A}(\widehat{x}^{(1)}) \in \mathcal{S}) \leq e^{\varepsilon} \times Pr(\mathcal{A}(\widehat{x}^{(2)}) \in \mathcal{S}) + \delta. \tag{2}$$

The randomized perturbation mechanism $\mathcal{A}$ is ε-differential privacy (ε-DP) when $\delta = 0$.

According to the Definition 2, a smaller tuple (ε, δ) means more indistinguishability of the two outputs under different inputs, providing stronger sensory data privacy protection. Next, we define the concept of sensitive information for each normal worker as below.

Definition 3 (Sensitive Information): Let $\widehat{x}^{m,(1)}$ and $\widehat{x}^{m,(2)}$ are two sensory data uploaded by normal worker $\widehat{u}_m$. The sensitive information of any normal worker $\forall \widehat{u}_m \in \widehat{U}$ is denoted as

$$\triangle_{\widehat{u}_m, p} = \max_{\widehat{x}^{m,(1)}, \widehat{x}^{m,(2)} \in \{\widehat{x}_n^m \mid \forall o_n \in O\}} \|\widehat{x}^{m,(1)} - \widehat{x}^{m,(2)}\|_p, \tag{3}$$

where $\|\widehat{x}^{m,(1)} - \widehat{x}^{m,(2)}\|_p$ is L_p-normal. Because $\|\widehat{x}^{m,(1)} - \widehat{x}^{m,(2)}\|_p = (|\widehat{x}^{m,(1)} - \widehat{x}^{m,(2)}|^p)^{1/p}$, the (3) can be reformulated as

$$\triangle_{\widehat{u}_m} = \max_{\widehat{x}^{m,(1)}, \widehat{x}^{m,(2)} \in \{\widehat{x}_n^m \mid \forall o_n \in O\}} |\widehat{x}^{m,(1)} - \widehat{x}^{m,(2)}|. \tag{4}$$

Laplace and Gaussian mechanisms [33] are widely used to achieve (ε, δ)-DP. Here, we detail these two mechanisms as follows.

Definition 4 (Laplace Mechanism): To achieve ε-DP, Laplace mechanism adds noise $\xi_n^m \in \mathcal{R}$ selected from $Lap(\triangle_{\widehat{u}_m}/\varepsilon)$ to sensory data $\widehat{x}_n^m$ as follows,

$$\widehat{y}_n^m = \widehat{x}_n^m + \xi_n^m. \tag{5}$$

Definition 5 (Gaussian Mechanism): The Gaussian mechanism achieves (ε, δ)-DP by adding noise $\xi_n^m \in \mathcal{R}$ to sensory data $\widehat{x}_n^m$ as follows,

$$\widehat{y}_n^m = \widehat{x}_n^m + \xi_n^m. \tag{6}$$

The noise ξ_n^m drawn from $N(0, \sigma_m^2)$, where $\sigma_m \geq \sqrt{2\ln(1.25/\delta)} \cdot \triangle_{\widehat{u}_m}/\varepsilon$.

TABLE I
NOTATIONS USED IN THE PROPOSED STACKELBERG-GAME-BASED DEFENSE APPROACH

Notation	Definition	Notation	Definition
$O/\|O\|$	Set/Number of all objects	o_n	n-th object, $o_n \in O$
$\widehat{U}/\|\widehat{U}\|$	Set/Number of all normal workers	$\widehat{u}_m$	m-th normal worker, $\widehat{u}_m \in \widehat{U}$
$\widetilde{U}/\|\widetilde{U}\|$	Set/Number of all malicious workers	$\widetilde{u}_m$	m-th malicious worker, $\widetilde{u}_m \in \widetilde{U}$
$U/\|U\|$	Set/Number of all workers, $\widehat{U} \cup \widetilde{U} = U$	u_m	m-th worker, $u_m \in U$
U^{known}	Set of known workers, $U^{known} \subset \widehat{U}$	pk	Percentage of known workers
$\|U^{known}\|$	Number of known workers, $\|U^{known}\| < \|\widehat{U}\|$	pm	Percentage of malicious workers
$\widehat{W}/\widetilde{W}$	Set of all normal/malicious worker weights	$\widehat{w}_m/\widetilde{w}_m$	Weight of m-th normal/malicious worker
W	Set of all worker weights, $\widehat{W} \cup \widetilde{W} = W$	w_m	Weight of m-th worker u_m
$\widehat{X}/\widehat{Y}$	Set of real/perturbed data sensed by all normal workers	$\widehat{x}_n^m/\widehat{y}_n^m$	Real/perturbed data sensed by $\widehat{u}_m$ for o_n
$\widetilde{X}$	Set of malicious data shared by all malicious workers	$\widetilde{x}_n^m$	Malicious data provided by $\widetilde{u}_m$ for o_n
Y	Set of sensory data collected by the server, $Y = \widehat{Y} \cup \widetilde{X}$	y_n^m	Sensory data uploaded by m-th worker for n-th object
X_a^{truth}	Set of the inferred truths after attacks	$x_{a,n}^{truth}$	Inferred truth after attacks for n-th object
X_b^{truth}	Set of the inferred truths before attacks	$x_{b,n}^{truth}$	Inferred truth before attacks for n-th object
$\mathcal{F}_a$	Set of attack strategy	$F_{a,n}$	Attack strategy for n-th object, $F_{a,n} \in \mathcal{F}_a$
$\mathcal{F}_a^*$	Set of defense strategy	$F_{a,n}^*$	Defense strategy for n-th object, $F_{a,n}^* \in \mathcal{F}_a^*$
η	Disguised level hidden behind DP noise	ζ	Stealth level of evading defenders

C. Truth Discovery Methods

The collected sensory data is conflicting, so various weight-based truth discovery methods have been proposed [34], [35], [36] to extract truths from the conflicting data. However, the rationale behind them is that we estimate the reliability (i.e., weight) of each worker based on its sensory data and then perform weight-based aggregation on all sensory data. In this paper, therefore, we use a state-of-the-art truth discovery method, named Conflict Resolution on Heterogeneous (CRH) [34], [36], as a concrete example to illustrate the basic idea of defending against data poisoning attacks in the DP-based privacy-preserving crowdsensing. CRH is formulated as the following optimization problem,

$$\min_{X^{truth},W} f(X^{truth}, W, Y) = \sum_{u_m \in U} w_m \sum_{o_n \in O} (y_n^m - x_n^{truth})^2, \tag{7a}$$

$$s.t. \sum_{u_m \in U} \exp(-w_m) = 1, \ |U| > 1, \tag{7b}$$

where $X^{truth} = \{x_n^{truth} \mid o_n \in O\}$ is the set of inferred truths for all objects, $W = \{w_m \mid u_m \in U\}$ is the set of weights for all workers, and $Y = \{y_n^m \mid o_n \in O, u_m \in U\}$ is the set of sensory data uploaded by all workers for all objects. This problem can be solved by iteratively alternating the following two steps.

Truths Estimate. In this step, according to the current worker weights W, the truths X^{truth} can be estimated by

$$x_n^{truth} = \frac{\sum_{u_m \in U} w_m \cdot y_n^m}{\sum_{u_m \in U} w_m}, \forall x_n^{truth} \in X^{truth}. \tag{8}$$

It can be seen from (8) that sensory data uploaded by high-weight workers are counted more in the aggregation. Therefore, when malicious workers are assigned higher weights, the malicious sensory data will have a greater impact on the truths X^{truth}.

Worker Weights Estimate: In this step, according to the current truths X^{truth}, the weights W can be calculated by

$$w_m = \log \frac{\sum_{u_m \in U} \sum_{o_n \in O} (y_n^m - x_n^{truth})^2}{\sum_{o_n \in O} (y_n^m - x_n^{truth})^2}, \forall w_m \in W. \tag{9}$$

(9) shows that a worker will be assigned a higher weight when its sensory data is close to most workers, and vice versa. Therefore, malicious workers can obtain higher weights by agreeing with the normal workers on objects where the truths cannot be successfully biased, and in turn, can exert a stronger impact on other objects.

Most weight-based truth discovery methods infer the truths by iteratively estimating truths and worker weights. They only adopt different truths-updating function (8) and worker-weights-updating function (9). Therefore, the proposed game-based defense approach can be easily applied to other truth discovery methods.

IV. PROBLEM STATEMENT

A. Defenders and Attackers

We consider *defenders* deployed on the crowdsensing server, aiming to mitigate the damage caused by malicious workers to the outputs of truth discovery methods while simultaneously ensuring that the DP protocol is not violated. As a result, the crowdsensing server is equipped with two detection mechanisms, i.e., the truth discovery methods and the defense strategy, to mitigate malicious workers. In particular, the truth discovery methods can estimate the reliability of workers, but they can easily be deceived by intelligent malicious workers who masquerade as higher-reliability workers [5], [15], [16], [17], [18], [19], [20]. Therefore, designing an additional defense mechanism to assist the truth discovery methods in detecting malicious workers hidden behind DP noise is extremely important.

In addition, we consider a worst-case scenario where *attackers*, who know the defense strategy and truth discovery method deployed in crowdsensing, can manipulate some workers to upload malicious sensory data while not influencing the behavior

of normal workers. Powered by that knowledge, the attackers could launch data poisoning attacks against DP-based privacy-preserving crowdsensing systems, aiming to introduce the maximum deviation in the outputs of truth discovery methods. The attacks are practical in real-world crowdsensing systems, e.g., the attackers might seek to manipulate a real-time navigation system in such a way that the system provides users with inaccurate or hazardous directions.

B. Attack Gain Metric

The attack gain is defined in terms of the amount of deviation an attacker can introduce into the truths, which can be measured by the euclidean distance. Let $X_b^{truth} = \{x_{b,n}^{truth} \mid \forall o_n \in O\}$ ($x_{b,n}^{truth}$ is the truth of n-th object before attacks) denotes the inferred truths before attacks, which can be inferred from the perturbed sensory data $\widehat{Y}$. Similarly, the truths after attacks can be inferred from all sensory data $Y = \widehat{Y} \cup \widetilde{X}$, which is represented as $X_a^{truth} = \{x_{a,n}^{truth} \mid \forall o_n \in O\}$ ($x_{a,n}$ is the truth of n-th object after attacks). Then, the attack gain on object o_n can be formulated as $(x_{b,n}^{truth} - x_{a,n}^{truth})^2$. Since the sensory data uploaded by normal workers are added with DP noise, the truths X_a^{truth} and X_b^{truth} are random and the deviation measured by $(x_{b,n}^{truth} - x_{a,n}^{truth})^2$ is always inaccurate. As a result, we adopt expectation, i.e., $(E[X_b^{truth}] - E[X_a^{truth}])^2$, to address the inaccuracy randomness (or DP noise), and the attack gain metric can be defined as follows.

Definition 6 (Attack Gain Metric): Let $X_b^{truth} = \{x_{b,n}^{truth} \mid \forall o_n \in O\}$ and $X_a^{truth} = \{x_{a,n}^{truth} \mid \forall o_n \in O\}$ be the inferred truths before and after attacks, respectively. The profit of data poisoning attacks can be defined as

$$U(X_b^{truth}, X_a^{truth}) = \sum_{o_n \in O} \left(E[x_{b,n}^{truth}] - E[x_{a,n}^{truth}]\right)^2, \quad (10)$$

where $E[x_{b,n}^{truth}]$ and $E[x_{a,n}^{truth}]$ represent the expectation of truths before and after attacks, respectively.

C. Problem Definition

Given the situations where

1) the DP-based privacy-preserving crowdsensing systems are suffering data poisoning attacks, and
2) the attackers aim to damage crowdsensing results as much as possible, i.e., maximize the attack gain $U(X_b^{truth}, X_a^{truth})$,

the problem is finding the effective defense strategy $\mathcal{D}_f$ that identifies malicious workers by minimizing the attack gain $U(X_b^{truth}, X_a^{truth})$. The defense strategy $\mathcal{D}_f$ must consider that the attackers

1) know the real sensory data of a subset of normal workers, i.e., $\{x_n^m \mid \forall o_n \in O, \forall u_m \in U^{known}\}$,
2) also know the DP protocol $\mathcal{A}$ and truth discovery method $\mathcal{T}_d$ implemented in crowdsensing, and
3) are aware of the defense strategy $\mathcal{D}_f$ adopted by the defenders.

Hence, the powerful attackers launch data poisoning attacks that skew the outputs of truth discovery methods, meanwhile avoiding the truth discovery method $\mathcal{T}_d$ and the defense strategy $\mathcal{D}_f$. Let the attack strategy be denoted as $\widetilde{X} \sim \mathcal{F}_a$ ($\mathcal{F}_a\{F_{a,n} \mid \forall o_n \in O\}$ and $F_{a,n}$ is the attack strategy for n-th object), i.e., the malicious sensory data $\forall \widetilde{x}_n^m \in \widetilde{X}$ is selected from $F_{a,n} \in \mathcal{F}_a$ for any object $o_n \in O$. The attacks can be formulated as

$$\max_{\widetilde{X} \sim \mathcal{F}_a} U(E[\mathcal{T}_d(\mathcal{A}(\widehat{X}))], E[\mathcal{T}_d(\mathcal{D}_f(\mathcal{A}(\widehat{X}) \cup \widetilde{X}))]), \quad (11)$$

where $X_b^{truth} := \mathcal{T}_d(\mathcal{A}(\widehat{X}))$ and $X_a^{truth} := \mathcal{T}_d(\mathcal{D}_f(\mathcal{A}(\widehat{X}) \cup \widetilde{X}))$. Particularly, $\mathcal{T}_d(\mathcal{A}(\widehat{X}))$ denotes that the truths before attacks can be inferred from the perturbed sensory data $\mathcal{A}(\widehat{X})$ by $\mathcal{T}_d$, where $\mathcal{A}(\widehat{X})$ represents that the DP noise is added to real sensory data through a perturbation mechanism $\mathcal{A}$. Similarly, $\mathcal{T}_d(\mathcal{D}_f(\mathcal{A}(\widehat{X}) \cup \widetilde{X}))$ indicates that the truths after attacks can be inferred from $\mathcal{D}_f(\mathcal{A}(\widehat{X}) \cup \widetilde{X})$ by $\mathcal{T}_d$, where $\mathcal{D}_f(\mathcal{A}(\widehat{X}) \cup \widetilde{X})$ means that the strategy $\mathcal{D}_f$ detects and deletes the malicious workers in crowdsensing by analyzing the collected sensory data $\mathcal{A}(\widehat{X}) \cup \widetilde{X}$. In addition, we quantify the attackers' capability and background knowledge by introducing two essential definitions:

Definition 7 (Percentage of Known Workers): Let the attackers have access to the real sensory data of a subset of normal workers, which is denoted as U^{known} and $U^{known} \subset \widehat{U}$ ($\widehat{U}$ is the set of normal workers). In other words, the attackers could illicitly obtain or closely monitor the real sensory data from known workers. The percentage of known workers is defined as

$$pk = |U^{known}|/|\widehat{U}|, \quad (12)$$

where $|U^{known}|$ and $|\widehat{U}|$ represent the number of known workers and normal workers, respectively.

Definition 8 (Percentage of Malicious Workers): Let the set of workers $\widetilde{U}$ is under the control of attackers, and let U encompasses all participating workers, where $\widetilde{U} \subset U$. In other words, the attackers can inject malicious workers $\widetilde{U}$ into the privacy-preserving crowdsensing systems. The percentage of malicious workers is defined as

$$pm = |\widetilde{U}|/|U|, \quad (13)$$

where $|\widetilde{U}|$ is the number of malicious workers, and $|U|$ represents the number of participating workers.

V. Game Formulation

A. Zero-Sum Stackelberg Game

The defenders detect malicious workers by deciding whether a random variable y_n^m belongs to hypothesis H_b (normal behavior) or H_a^* (malicious behavior). One of the most well-known results of hypothesis testing is that we could find the optimal receiver operating characteristic (ROC) curve between the exact distribution $\mathcal{F}_b = \{F_{b,n} \mid \forall o_n \in O\}$ of the values under H_b and the exact distribution $\mathcal{F}_a^* = \{F_{a,n}^* \mid \forall o_n \in O\}$ of the values under the alternative hypothesis H_a^* by the log-likelihood ratio test. Thus, the defenders detect malicious workers by the log-likelihood ratio test, which can be formulated as

$$\mathcal{D}_f := p(u_m) = \frac{\wedge(u_m) - \min(V)}{\max(V) - \min(V)}, \forall u_m \in U, \quad (14)$$

where

$$\wedge(u_m) = \sum_{o_n \in O} \log \frac{F^*_{a,n}(y^m_n)}{F_{b,n}(y^m_n)}, \forall u_m \in U, \tag{15}$$

and $V = \{\wedge(u_m) \mid \forall u_m \in U\}$. The defenders identify worker u_m as malicious if $p(u_m) > \tau$, where τ is an empirical parameter.

Unfortunately, the defenders do not know the attack strategy $\mathcal{F}_a$, i.e., $\mathcal{F}^*_a \neq \mathcal{F}_a$, because they act first. We describe this defenders-attackers interaction as a zero-sum Stackelberg game, where the defenders and attackers are the leaders and followers, respectively. Particularly, the defenders play first as *the leaders*, i.e., minimizing the damage caused by malicious workers, and disclose the selected defense strategy $\mathcal{D}_f$ (depends on $\mathcal{F}^*_a$ and detailed in (14)). The attackers play next as *the followers*, given the current defense strategy $\mathcal{D}_f$, and optimize their attack strategy $\mathcal{F}_a$ by maximizing attack gain with a fixed $\mathcal{D}_f$. A strategy of the game is a pair $(\mathcal{F}^*_a, \mathcal{F}_a)$, where the defenders' (or attackers') strategy is $\mathcal{F}^*_a$ (or $\mathcal{F}_a$). Besides, we assume that both attackers and defenders are rational, meaning they always choose the strategy that maximizes their utilities. This game is zero-sum because the defenders' gain (or loss) is exactly balanced by the attackers' loss (or gain). In other words, the skew introduced by attack is the gain of the attackers and the loss of the defenders, respectively. After the above discussion, the defenders-attackers interaction is formulated as a minimax optimization problem, i.e.,

$$\min_{\mathcal{D}_f (\text{or}\, \mathcal{F}^*_a)} \max_{\widetilde{X} \sim \mathcal{F}_a} U(E[\mathcal{T}_d(\mathcal{A}(\widehat{X}))], E[\mathcal{T}_d(\mathcal{D}_f(\mathcal{A}(\widehat{X}) \cup \widetilde{X}))]). \tag{16}$$

In this Min-Max optimization problem, the min-players (i.e., defenders) are the first player, and the max-players (i.e., attackers) are the second. Next, we give a detailed definition of the attackers' utilities and the defenders' utilities below.

Definition 9 (Players' Utilities): Given the attack strategy $\mathcal{F}_a$ and the defense strategy $\mathcal{F}^*_a$. According to the attack gain metric (detailed in Definition 6), the attackers' utilities and the defenders' utilities are defined as

$$U_{attackers} = U(X^{truth}_b, X^{truth}_a), \tag{17}$$

$$U_{defenders} = -U(X^{truth}_b, X^{truth}_a), \tag{18}$$

where $U(X^{truth}_b, X^{truth}_a)$ is the attack gain, X^{truth}_b represents the truths before attacks, and X^{truth}_a denotes the truths after attacks. Specifically, X^{truth}_b can be obtained by aggregating the perturbed sensory data of known workers using truth discovery methods. X^{truth}_a can be obtained by following three steps: First, the defenders deploy their defense strategy $\mathcal{F}^*_a$; Second, the attackers upload malicious sensory data according to the poisoning strategy $\mathcal{F}_a$; Third, X^{truth}_a is calculated by feeding the successfully poisoned sensory data into truth discovery methods.

B. Unveiling the Normal Behavior of Workers

As detailed in Section V-A, the computation of exact distribution $\mathcal{F}_b = \{F_{b,n} \mid \forall o_n \in O\}$ (i.e., the normal behavior of workers) is paramount. Next, we present two theorems about the normal behavior of workers under the Laplace mechanism and the Gaussian mechanism, respectively.

Theorem 1 (Normal Behavior under Laplace Mechanism): Let the inferred truths before poisoning attacks is $X^{truth}_b = \{x^{truth}_{b,n} \mid \forall o_n \in O\}$. If the sensory data is added DP noise by the Laplace mechanism, i.e., $Lap(\triangle_{\widehat{u}_m}/\varepsilon), \forall \widehat{u}_m \in \widehat{U}$, the normal behavior $\mathcal{F}_b = \{F_{b,n} \mid \forall o_n \in O\}$ of workers can be formulated as

$$F_{b,n}(y) = \frac{\varepsilon \cdot |\widehat{U}|}{2\sum_{\widehat{u}_m \in \widehat{U}} \triangle_{\widehat{u}_m}} e^{-\frac{\varepsilon \cdot |y - x^{truth}_{b,n}| \cdot |\widehat{U}|}{\sum_{\widehat{u}_m \in \widehat{U}} \triangle_{\widehat{u}_m}}}, \forall F_{b,n} \in \mathcal{F}_b, \tag{19}$$

where $\triangle_{\widehat{u}_m}$ is the sensitive information of user $\widehat{u}_m$ and defined in Definition 3.

Proof: To protect normal workers' sensory data privacy, under the Laplace mechanism, real data is added with Laplace noise before sharing. That is, the noise $\xi^m_n \sim Lap(0, \triangle_{\widehat{u}_m}/\varepsilon), \forall \widehat{u}_m \in \widehat{U}$. Next, we calculate the distribution that average noise $\xi_n := \frac{\sum_{\widehat{u}_m \in \widehat{U}} \xi^m_n}{|\widehat{U}|}$ (random variable) follows. Since $\forall \xi^m_n$ follow $Lap(0, \frac{\triangle_{\widehat{u}_m}}{\varepsilon})$, we can estimate the scale parameter of distribution that ξ_n obeys using expectation scale $\frac{\sum_{\widehat{u}_m \in \widehat{U}} \triangle_{\widehat{u}_m}}{\varepsilon \cdot |\widehat{U}|}$. Thus, we have $\xi_n \sim Lap(0, \frac{\sum_{\widehat{u}_m \in \widehat{U}} \triangle_{\widehat{u}_m}}{\varepsilon \cdot |\widehat{U}|})$. As a result, the normal behavior $F_{b,n}(y) = Lap(x^{truth}_{b,n}, \frac{\sum_{\widehat{u}_m \in \widehat{U}} \triangle_{\widehat{u}_m}}{\varepsilon \cdot |\widehat{U}|}), \forall F_{b,n} \in \mathcal{F}_b$, i.e., (19) is satisfied. □

Theorem 2 (Normal Behavior under Gaussian Mechanism): Let the inferred truths before poisoning attacks is $X^{truth}_b = \{x^{truth}_{b,n} \mid \forall o_n \in O\}$. If the sensory data is added DP noise by the Gaussian mechanism, i.e., $N(0, \sigma^2_m), \forall \widehat{u}_m \in \widehat{U}$, the normal behavior $\mathcal{F}_b = \{F_{b,n} \mid \forall o_n \in O\}$ of workers can be formulated as

$$F_{b,n}(y) = \frac{1}{\upsilon_n \cdot \sqrt{2\pi}} e^{-\frac{\left(y - x^{truth}_{b,n}\right)^2}{2\upsilon_n}}, \forall F_{b,n} \in \mathcal{F}_b, \tag{20}$$

where υ^2_n is the variance of distribution $F_{b,n}$ and can be calculated by

$$\upsilon^2_n = \frac{\sum_{u_m \in \widehat{U}} \sigma^2_m}{|\widehat{U}|}. \tag{21}$$

Proof: The normal workers' real data is added with Gaussian noise before sharing, i.e., the noise $\xi^m_n \sim N(0, \sigma^2_m), \forall \widehat{u}_m \in \widehat{U}$. Because of the additivity of the Gaussian distribution, i.e., if $\xi^1_n \sim N(0, \sigma^2_1)$ and $\xi^2_n \sim N(0, \sigma^2_2)$ then $\xi^1_n + \xi^2_n \sim N(0, \sigma^2_1 + \sigma^2_2)$, we have $\frac{\sum_{\widehat{u}_m \in \widehat{U}} \xi^m_n}{|\widehat{U}|} \sim N(0, \frac{\sum_{\widehat{u}_m \in \widehat{U}} \sigma^2_m}{|\widehat{U}|})$. Thus, the normal behavior $F_{b,n}(y) = N(x^{truth}_{b,n}, \frac{\sum_{\widehat{u}_m \in \widehat{U}} \sigma^2_m}{|\widehat{U}|}), \forall F_{b,n} \in \mathcal{F}_b$, i.e., (20) is satisfied. □

VI. Game-Based Defense Approach

As reported in [5], [31], DP protocols are vulnerable to data poisoning attacks, and can even facilitate the success of such attacks against crowdsensing systems. In other words,

malicious workers can disguise themselves as normal workers by reducing the distance between their poisoning strategy and the distribution followed by perturbed sensory data, thereby bypassing truth discovery methods. This new poisoning strategy is known as "hiding behind the DP noise". To defend against those poisoning attacks, we propose a Stackelberg-game-based defense approach. In this game, the defenders play first by determining their defense strategy, and the attackers play next by launching data poisoning attacks. Specifically, the defenders adopt an identify-then-drop method to defend against the data poisoning attacks, i.e., the defenders identify malicious workers using the log-likelihood ratio test and then drop the sensory data shared by those identified workers. Powered by this method, the defenders formulate their defense strategy as a functional minimization problem $(\mathcal{O}_D)$, aiming to minimize the bias caused by poisoning attacks (detailed in Section VI-A). In addition, we consider the powerful attackers, who are aware of the DP protocol and defense strategy deployed in crowdsensing systems, can bypass truth discovery methods by exploiting DP noise and even evade the defenders by constraining the expected log-likelihood ratio test. Accordingly, the attackers formulate their poisoning strategy as a bi-level maximization problem $(\mathcal{O}_A)$, aiming to maximize the attack gain (detailed in Section VI-B). To this end, we formulate the attackers-defenders interaction as a zero-sum Stackelberg game, finding the optimal defense strategy to defend against the powerful data poisoning attack (detailed in Section VI-C).

A. Defense Strategy for Defenders

The defenders, acting as the role of leaders, take the initiative by conducting the log-likelihood ratio test to identify malicious workers. This task can be formulated as a minimization problem, aiming to minimize the damage caused by malicious workers by optimizing the defense strategy $\mathcal{F}_a^*$, i.e.,

$$(\mathcal{O}_D): \min_{\mathcal{F}_a^*} \sum_{o_n \in O} \left(E[x_{b,n}^{truth}] - E[x_{a,n}^{truth}]\right)^2, \tag{22a}$$

$$\text{s.t. } U_D = \{u_m \mid p(u_m) > \tau, \forall u_m \in U\}, \tag{22b}$$

$$Z = \{z_n^m = y_n^m \mid \forall u_m \in U \backslash U_D, \forall o_n \in O\}, \tag{22c}$$

$$E[x_{b,n}^{truth}] = \int_{-\infty}^{+\infty} y \cdot F_{b,n}(y) dy, \tag{22d}$$

$$E[x_{a,n}^{truth}] = \int_{-\infty}^{+\infty} y \cdot F_{a,n}^*(y) dy, \tag{22e}$$

$$\{X_a^{truth}, W^*\} := \underset{X_a^{truth}, W^*}{\arg\min}\ f(X_a^{truth}, W^*, Z), \tag{22f}$$

where $p(u_m), \forall u_m \in U$ is determined by the defense strategy $\mathcal{F}_a^*$ and can be calculated by (14)–(15). Next, we will detail the objective and constraints of problem $(\mathcal{O}_D)$. The object (22a) represents that the defenders aim to maximize their utilities, i.e., maximizing $U_{defenders}$ (detailed in Definition 9). The constraints (22b) and (22c) represent that the defenders perform the identify-then-drop method, finding malicious workers by the log-likelihood ratio test and deleting the sensory data shared by the identified workers from the collected data. The constraints (22d) and (22e) represent the expectations of random variables $x_{b,n}^{truth}$ and $x_{a,n}^{truth}$, respectively. The constraints (22f) are the truth discovery method (detailed in (7a) and (7b)), where $W^* = \{w_m \mid u_m \in U \backslash U_D\}$ denotes the set of weights for all workers after the identify-then-drop method has been completed.

Due to the decision variable (i.e., $\mathcal{F}_a^*$) is a function set, the problem $(\mathcal{O}_D)$ is a functional optimization problem and is difficult to solve directly [37]. Consequently, we adopt the variable-basis approximation schemes [38], a class of schemes that approximate the decision variable using linear combinations of fixed basis functions, to relax the optimization problem from optimizing over functions to optimizing over the coefficients of the linear combinations. In our approximation scheme, we employ Hermite polynomials [39], a well-known orthogonal polynomial sequence and suitable for approximating any function over $(-\infty, +\infty)$, as the basis functions. Particularly, the recurrence relation of Hermite polynomials is formulated as $H_{e_{r+1}}(x) = x \cdot H_{e_r}(x) - \frac{d}{dx} H_{e_r}(x)$ ($H_{e_r}(x)$ is the r-th Hermite polynomial), and the first four Hermite polynomials are: $H_{e_0}(x) = 1$, $H_{e_1}(x) = x$, $H_{e_2}(x) = x^2 - 1$, and $H_{e_3}(x) = x^3 - 3x$. The decision variable $\mathcal{F}_a^*$ can be approximated by the rth-order Hermite polynomials and can be formulated as

$$F_{a,n}^*(x) = \psi_{n,0}^* + \psi_{n,1}^* \cdot H_{e_1}(x) + \psi_{n,2}^* \cdot H_{e_2}(x) + \cdots + \psi_{n,r-1}^* \cdot H_{e_{r-1}}(x), \forall o_n \in O, \tag{23}$$

where $\Psi_n^* = [\psi_{n,0}^*, \psi_{n,1}^*, \ldots, \psi_{n,r-1}^*]_{1\times r}$ is a coefficient vector of $F_{a,n}^*(x)$. As a result, since the basis functions (i.e., $H_{e_r}(x)$) are fixed in advance, the optimization problem shifts from optimizing the functions $\mathcal{F}_a^* = \{F_{a,n}^* \mid \forall o_n \in O\}$ to optimizing the coefficient matrix $\Phi^* = [\Psi_0^*, \Psi_1^*, \ldots, \Psi_{|O|}^*]_{|O|\times r}^T$. This relaxation simplifies the problem $(\mathcal{O}_D)$ and enables the simplified problem (denoted as $(\mathcal{O}_D^1)$) to be effectively solved utilizing numerical optimization algorithms.

Given that the problem $(\mathcal{O}_D^1)$ is a large-scale global optimization, i.e., the decision variable is a $|O| \times r$-dimensional matrix, we employ the simulated annealing genetic algorithm (SAGA) [40] for solving it and are outlined in the Algorithm 1. SAGA is a metaheuristic that combines the advantages of both simulated annealing (SA) algorithm [41] (a single-solution based metaheuristic) and genetic algorithm (GA) [42] (a population-based metaheuristic) to approximate the global optimum in a vast search space. The algorithm starts with *initialization* and then iteratively performs *crossover*, *mutation*, *fitness evaluation*, and *simulated annealing*, which are described in detail below.

In the initialization operation, we utilize a normal distribution for generating the initial population with SN individuals. The initial (or 0-th) population is denoted as $\{\Phi_i^{*,0} \mid \forall i \in [0, SN]\}$, where $\Phi_i^{*,0} = [\Psi_{i,0}^{*,0}, \Psi_{i,1}^{*,0}, \ldots, \Psi_{i,|O|}^{*,0}]_{|O|\times r}^T$ is the i-th individual (or solution candidate) and $\Psi_{i,n}^{*,0} = [\psi_{i,n,0}^{*,0}, \psi_{i,n,1}^{*,0}, \ldots, \psi_{i,n,r-1}^{*,0}]_{1\times r}$ is the coefficient vector candidate of the defense strategy $F_{a,n}^*(x)$. Particularly, the coefficient

Algorithm 1: Defense Strategy Algorithm.

Input : $Y = \{y_n^m \mid \forall o_n \in O, \forall u_m \in U\}$: the collected sensory data set; $\mathcal{F}_b = \{F_{b,n} \mid \forall o_n \in O\}$: the distribution followed by the perturbed sensory data; τ: the empirical parameter utilized in log-likelihood ratio test; SN: the population size; G_{max}: the maximum number of generations; p_c: the crossover probability; p_m: the mutation probability; T_{max}, T_{min}: the initial and final temperatures; β: the cooling factor;

Output: $\Phi^* = [\Psi_0^*, \cdots, \Psi_{|O|}^*]_{|O|\times r}^T$: the coefficient matrix of rth-order Hermite polynomials; U_D: the detected malicious workers set.

$\{\Phi_0^{*,0}, \cdots, \Phi_{SN}^{*,0}\} \leftarrow$ Initialize a population with SN individuals utilizing the initialization operation;
Evaluate the fitness of 0-th population $\{\Phi_0^{*,0}, \cdots, \Phi_{SN}^{*,0}\}$ utilizing the fitness evaluation operation;
$g \leftarrow 0$;
while $g \leq G_{max}$ *and* $T_{max} > T_{min}$ **do**
  $\{\Theta_0^{*,g}, \cdots, \Theta_{SN}^{*,g}\} \leftarrow$ Generate offspring by selecting parent individuals from g-th population and then performing crossover operation;
  $\{\Xi_0^{*,g}, \cdots, \Xi_{SN}^{*,g}\} \leftarrow$ Generate mutated offspring by performing mutuation operation on each $\Theta_k^{*,g}$;
  Evaluate the fitness of each offspring in $\{\Xi_0^{*,g}, \cdots, \Xi_{SN}^{*,g}\}$ utilizing the fitness evaluation operation;
  $\{\Phi_0^{*,g+1}, \cdots, \Phi_{SN}^{*,g+1}\} \leftarrow$ Select SN individuals from $\{\Xi_0^{*,g}, \cdots, \Xi_{SN}^{*,g}\} \cup \{\Phi_0^{*,g}, \cdots, \Phi_{SN}^{*,g}\}$ utilizing the simulated annealing operation;
  $T_{max} \leftarrow \beta \cdot T_{max}$;
  $g \leftarrow g + 1$;
end
$\Phi^* \leftarrow$ optimal individual in the g-th population;

candidate $\forall \psi_{i,n,j}^{*,0} \in \Psi_{i,n}^{*,0} \in \Phi_i^{*,0}$ drawn from $N(\mu, \sigma^2)$, i.e., $\psi_{i,n,j}^{*,0} \sim N(\mu, \sigma^2)$, where μ and σ^2 represent expectation and variance, respectively.

In the fitness evaluation operation, we evaluate the fitness of each individual in g-th population $\{\Phi_0^{*,g}, \ldots, \Phi_{SN}^{*,g}\}$ ($0 \leq g \leq G_{\max}$). The fitness of $\forall \Phi_i^{*,g}$ (i-th individual in g-th population) can be calculated by

$$Fit(\Phi_i^{*,g}) = -\sum_{o_n \in O} (E[x_{b,n}^{truth}] - E[x_{a,n}^{truth}])^2, \forall i \in [0, SN], \tag{24}$$

where $E[x_{b,n}^{truth}]$ is a fixed value and $E[x_{a,n}^{truth}]$ is dependent on $\Phi_i^{*,g}$. Specifically, we can obtain the defense strategy $\mathcal{F}_a^*$ by (23) according to the coefficient matrix $\Phi_i^{*,g}$ (i.e., rth-order Hermite polynomials approximation), secondly identify malicious workers through the log-likelihood ratio test (i.e., constraint (22b)), thirdly delete the sensory data uploaded by the identified worker U_D from the collected data (i.e., constraint (22c)), fourthly infer the truths $x_{a,n}^{truth}$ from the deleted data $Z = \{z_n^m \mid \forall u_m \in U \backslash U_D, \forall o_n \in O\}$ (i.e., constraint (22f)), and finally calculate the expectation $E[x_{a,n}^{truth}]$ by constraint (22e). As detailed in Section III-C, the expectation $x_{a,n}^{truth}$ can be inferred (or the sub-problem (22f) can be solved) by iterating the following two steps:

1) Fixing the weights $W = \{w_m \mid \forall u_m \in U \backslash U_D\}$, the truths $X_a^{truth} = \{x_{a,n}^{truth} \mid \forall o_n \in O\}$ are calculated by

$$x_{a,n}^{truth} = \frac{\sum_{u_m \in U \backslash U_D} w_m \cdot z_n^m}{\sum_{u_m \in U \backslash U_D} w_m}; \tag{25}$$

2) After obtaining the truths X_a^{truth}, the weights $W = \{w_m \mid \forall u_m \in U \backslash U_D\}$ are estimated by

$$w_m = \log \frac{\sum_{u_m \in U \backslash U_D} \sum_{o_n \in O} (z_n^m - x_{a,n}^{truth})^2}{\sum_{o_n \in O} (z_n^m - x_{a,n}^{truth})^2}. \tag{26}$$

In the crossover operations, we utilize the tournament selection method [43] to select two-parent individuals from the g-th population $\{\Phi_0^{*,g}, \ldots, \Phi_{SN}^{*,g}\}$. Given two-parent individuals $\Phi_i^{*,g} = [\Psi_{i,0}^{*,g}, \ldots, \Psi_{i,|O|}^{*,g}]$ ($i \in [0, SN]$) and $\Phi_j^{*,g} = [\Psi_{j,0}^{*,g}, \ldots, \Psi_{j,|O|}^{*,g}]$ ($j \in [0, SN]$), the crossover operation is performed between them when a randomly generated number ω_c, drawn from a uniform distribution $U(0,1)$, satisfies $\omega_c < p_c$. Particularly, we randomly select the position vector $\mathcal{I}_c = [I_c^0, I_c^1, \ldots, I_c^{|O|}]$ ($I_c^k \in [0, r-1]$ represents the k-th crossover position between $\Phi_i^{*,g}$ and $\Phi_j^{*,g}$), and then generate offspring individuals by swapping elements (or coefficients) after the designated positions. That is, by swapping the elements after I_c^k ($\forall I_c^k \in \mathcal{I}_c$) between $\Psi_{i,k}^{*,g} = [\psi_{i,k,0}^{*,g}, \ldots, \psi_{i,k,I_c^k}^{*,g}, \ldots, \psi_{i,k,r-1}^{*,g}]$ ($\forall \Psi_{i,k}^{*,g} \in \Phi_i^{*,g}$) and $\Psi_{j,k}^{*,g} = [\psi_{j,k,0}^{*,g}, \ldots, \psi_{j,k,I_c^k}^{*,g}, \ldots, \psi_{j,k,r-1}^{*,g}]$ ($\forall \Psi_{j,k}^{*,g} \in \Phi_i^{*,g}$), we can obtain $\Gamma_{i,k}^{*,g} = [\psi_{i,k,0}^{*,g}, \ldots, \psi_{i,k,I_c^k-1}^{*,g}, \psi_{j,k,I_c^k}^{*,g}, \ldots, \psi_{j,k,r-1}^{*,g}]$ and $\Gamma_{j,k}^{*,g} = [\psi_{j,k,0}^{*,g}, \ldots, \psi_{j,k,I_c^k-1}^{*,g}, \psi_{i,k,I_c^k}^{*,g}, \ldots, \psi_{i,k,r-1}^{*,g}]$. After the crossover between $\Phi_i^{*,g}$ and $\Phi_j^{*,g}$, we can obtain the offspring individuals $\Theta_i^{*,g} = [\Gamma_{i,0}^{*,g}, \Gamma_{i,1}^{*,g}, \ldots, \Gamma_{i,r-1}^{*,g}]$ and $\Theta_j^{*,g} = [\Gamma_{j,0}^{*,g}, \Gamma_{j,1}^{*,g}, \ldots, \Gamma_{j,r-1}^{*,g}]$. In addition, we denote the g-th population after crossover as $\{\Theta_0^{*,g}, \Theta_1^{*,g}, \ldots, \Theta_{SN}^{*,g}\}$.

In the mutation operation, we perform the mutation operation on the offspring population $\{\Theta_0^{*,g}, \Theta_1^{*,g}, \ldots, \Theta_{SN}^{*,g}\}$. If the randomly selected number ω_m, drawn from a uniform distribution $U(0,1)$, is less than the mutation probability p_m, the offspring individual $\Theta_i^{*,g} = [\Gamma_{i,0}^{*,g}, \Gamma_{i,1}^{*,g}, \ldots, \Gamma_{i,r-1}^{*,g}]$ ($\forall i \in [0, SN]$) undergoes mutation to produce $\Xi_i^{*,g}$. Specifically, we randomly choose the position vector $\mathcal{I}_m = [I_m^0, I_m^1, \ldots, I_m^{|O|}]$ ($I_m^k \in [0, r-1]$ denotes the k-th mutation position for $\Theta_i^{*,g}$). Then, we choose a random number $\chi_{i,k,I_m^k}^{*,g}$ ($\forall I_m^k \in \mathcal{I}_m$) from a normal distribution $N(\mu, \sigma^2)$, i.e., $\chi_{i,k,I_m^k}^{*,g} \sim N(\mu, \sigma^2)$, to replace the I_m^k-th element in the $\Gamma_{i,k}^{*,g} = [\psi_{i,k,0}^{*,g}, \ldots, \psi_{i,k,I_m^k}^{*,g}, \ldots, \psi_{i,k,r-1}^{*,g}]$ ($\forall \Gamma_{i,k}^{*,g} \in \Theta_i^{*,g}$), i.e., $\Gamma_{i,k}^{*,g} = [\psi_{i,k,0}^{*,g}, \ldots, \chi_{i,k,I_m^k}^{*,g}, \ldots, \psi_{i,k,r-1}^{*,g}]$. As a result, we can obtain the mutated population $\Xi_i^{*,g} = [\Gamma_{i,0}^{*,g}, \Gamma_{i,1}^{*,g}, \ldots, \Gamma_{i,r-1}^{*,g}]$.

In the simulated annealing operation, we select SN individuals from both the parents population (g-th population $\{\Phi_0^{*,g},\ldots,\Phi_{SN}^{*,g}\}$) and the offspring population (g-th population $\{\Xi_0^{*,g},\ldots,\Xi_{SN}^{*,g}\}$) as the next generation population ($g+1$-th population $\{\Phi_0^{*,g+1},\ldots,\Phi_0^{*,g+1}\}$). Particularly, given parent $\Phi_k^{*,g}$ ($\forall k\in[0,SN]$) and offspring $\Xi_k^{*,g}$ ($\forall k\in[0,SN]$), if offspring $\Xi_k^{*,g}$ is better (i.e., $Fit(\Xi_k^{*,g})>Fit(\Phi_k^{*,g})$), remain offspring $\Xi_k^{*,g}$ instead of parent $\Phi_k^{*,g}$; else if parent $\Phi_k^{*,g}$ is better (i.e., $Fit(\Phi_k^{*,g})>Fit(\Xi_k^{*,g})$), still remain offspring $\Xi_k^{*,g}$ with a certain probability. That is, the offspring $\Xi_k^{*,g}$ is remained when $\exp(-(Fit(\Phi_k^{*,g})-Fit(\Xi_k^{*,g}))/T_{\max})\geq\omega_{sa}$ is holds (ω_{sa} is a random number drawn from a uniform distribution $U(0,1)$).

We provide a detailed analysis of the computational complexity of Algorithm 1 (i.e., defense strategy algorithm). Specifically, the defense strategy algorithm begins with initialization, followed by crossover, mutation, fitness evaluation, and simulated annealing during each iteration. The computational complexity of these five steps, i.e., initialization, crossover, mutation, fitness evaluation, and simulated annealing, are $O(|O|\times r\times SN)$, $O(|O|\times|U|)$, $O(SN\times|r|\times|O|)$, $O(SN\times|O|)$, and $O(SN\times|O|\times|U|)$ respectively. As a result, the computational complexity of Algorithm 1 is $O(G_{\max}\times SN\times|O|\times|U|)$.

B. Attack Strategy for Attackers

The defenders are leaders play first by giving defense mechanism $\mathcal{F}_a^*$, and the attackers are followers play next by launching data poisoning attacks. Since the defense strategy $\mathcal{F}_a^*$ and the truth discovery method are known, the attacks can remain undetected by disguising. First, the attackers utilize DP noise to hide malicious behavior, and ultimately bypass the truth discovery method. Particularly, the attackers can estimate the distributions followed by the perturbed sensory data, i.e., $\widehat{y}_n^m\sim F_{b,n},\forall o_n\in O$, and then hide within the noise by ensuring that the attack strategy $\mathcal{F}_a=\{F_{a,n}\mid\forall o_n\in O\}$ aligns closely with the estimated distributions $\mathcal{F}_b=\{F_{b,n}\mid\forall o_n\in O\}$. The distance between probability distributions can be measured by Kullback-Leibler (KL) divergence [44], which is formulated as

$$D_{KL}(F_{a,n}\|F_{b,n})=\int_{-\infty}^{+\infty}F_{a,n}(x)\cdot\log\frac{F_{a,n}(x)}{F_{b,n}(x)}dx. \tag{27}$$

Second, the attackers can effectively limit the expected log-likelihood ratio test to avoid detection, considering that the defenders rely on this test to identify malicious workers. When the defenders employ the attack strategy $\mathcal{F}_a^*$, the expectation can be formulated as

$$E[\wedge]=\sum_{o_n\in O}\int_{-\infty}^{+\infty}F_{a,n}(y)\cdot\log\frac{F_{a,n}^*(y)}{F_{b,n}(y)}dy. \tag{28}$$

After the above discussion, the attack strategy can be formulated as the following maximization problem when the defense strategy $\mathcal{F}_a^*$ is fixed,

$$(\mathcal{O}_A):\max_{\widetilde{X}\sim\mathcal{F}_a}\sum_{o_n\in O}\left(E[x_{b,n}^{truth}]-E[x_{a,n}^{truth}]\right)^2, \tag{29a}$$

$$\text{s.t.}\sum_{o_n\in O}D_{KL}(F_{a,n}\|F_{b,n})\leq\eta, \tag{29b}$$

$$E[\wedge]\leq\zeta, \tag{29c}$$

$$\int_{-\infty}^{+\infty}F_{a,n}(y)dy=1,\ \forall o_n\in O, \tag{29d}$$

$$E[x_{b,n}^{truth}]=\int_{-\infty}^{+\infty}y\cdot F_{b,n}(y)dy, \tag{29e}$$

$$E[x_{a,n}^{truth}]=\int_{-\infty}^{+\infty}y\cdot F_{a,n}(y)dy, \tag{29f}$$

$$\{X_a^{truth},W\}:=\arg\min_{X_a^{truth},W}f(X_a^{truth},W,Z), \tag{29g}$$

where the threshold η is the disguise level hidden behind DP noise, and the threshold ζ dictates the stealth level of evading defenders. Particularly, a smaller value of η (or ζ) implies a lower probability of being detected by the truth discovery methods (or defenders). Next, we will detail the object and constraints of problem $(\mathcal{O}_A)$. The object (29a) represents that the attackers aim to maximize their utilities, i.e., maximizing $U_{attackers}$ (detailed in Definition 9). The attackers employ constraints (29b) and (29c) to bypass the truth discovery method and the defenders. The constraints (29e) and (29f) are the expectations of random variables $x_{b,n}^{truth}$ and $x_{a,n}^{truth}$, respectively. The constraint (29g) is the truth discovery method, constituting a lower-level sub-problem.

The problem $(\mathcal{O}_A)$ is a bi-level optimization [45], which can be decomposed into a lower-level sub-problem (29g) and a upper-level sub-problem $(\mathcal{O}_A^U)$ and solved by optimizing them alternately. Next, we describe in detail how to solve problems $(\mathcal{O}_A^U)$ and (29g) respectively.

1) Upper-Level Sub-Problem: The upper-level sub-problem $(\mathcal{O}_A^U)$ of problem $(\mathcal{O}_A)$ is formulated as

$$(\mathcal{O}_A^U):\max_{\widetilde{X}\sim\mathcal{F}_a}\sum_{o_n\in O}\left(E[x_{b,n}^{truth}]-E[x_{a,n}^{truth}]\right)^2, \tag{30a}$$

$$\text{s.t.}\sum_{o_n\in O}D_{KL}(F_{a,n}\|F_{b,n})\leq\eta, \tag{30b}$$

$$E[\wedge]\leq\zeta, \tag{30c}$$

$$\int_{-\infty}^{+\infty}F_{a,n}(y)dy=1,\ \forall o_n\in O, \tag{30d}$$

$$E[x_{b,n}^{truth}]=\int_{-\infty}^{+\infty}y\cdot F_{b,n}(y)dy, \tag{30e}$$

$$E[x_{a,n}^{truth}]=\int_{-\infty}^{+\infty}y\cdot F_{a,n}(y)dy, \tag{30f}$$

$$x_{a,n}^{truth}=\left(\sum_{\widehat{u}_m\in\widehat{U}}\widehat{w}_m\cdot\widehat{x}_n^m+\sum_{\widetilde{u}_m\in\widetilde{U}}\widetilde{w}_m\cdot\widetilde{x}_n^m\right)\Big/\left(\sum_{\widehat{u}_m\in\widehat{U}}\widehat{w}_m+\sum_{\widetilde{u}_m\in\widetilde{U}}\widetilde{w}_m\right), \tag{30g}$$

where the truths $x_{a,n}^{truth}$ can be estimated by (30g) if the worker weights $W=\widehat{W}\cup\widetilde{W}$ are fixed.

Algorithm 2: Attack Strategy Algorithm.

Input : $Y = \{y_n^m \mid \forall o_n \in O, \forall u_m \in U\}$: the collected sensory data set; $\mathcal{F}_b = \{F_{b,n} \mid \forall o_n \in O\}$: the distribution followed by the perturbed sensory data; $\mathcal{F}_a^* = \{F_{a,n}^* \mid \forall o_n \in O\}$: the current defense strategy; κ: the threshold for convergence;

Output: $\mathcal{F}_a = \{F_{a,n} \mid \forall o_n \in O\}$: the current attack strategy.

$\mathcal{F}_a = \{F_{a,n} \mid \forall o_n \in O\} \leftarrow$ The attack strategy is initialized to normal behavior, i.e., $F_{a,n} = F_{b,n}, \forall o_n \in O$;

$i \leftarrow 0$;

$\{X_a^{truth,i}, W\} \leftarrow$ Solve the lower-level problem Eq. (29g) using current attack strategy $\mathcal{F}_a$ by iterating Eq. (38) and Eqs.(39)-(40);

while *True* **do**

  $\mathcal{F}_a = \{F_{a,n} \mid \forall o_n \in O\} \leftarrow$ Solve the upper-level problem $(\mathcal{O}_A^U)$ using current worker weights W by Theorem 3;

  $\{X_a^{truth,i+1}, W\} \leftarrow$ Solve the lower-level problem Eq. (29g) using current attack strategy $\mathcal{F}_a$ by iterating Eq. (38) and Eqs.(39)-(40);

  Calculate the expectations $E[X_a^{truth,i+1}]$ and $E[X_a^{truth,i}]$ by constraints (29e) and (29f);

  if $|E[X_a^{truth,i+1}] - E[X_a^{truth,i}]| < \kappa$ **then**

    break;

  end

  $i \leftarrow i + 1$;

end

To overcome the challenge of directly solving the sub-problem $(\mathcal{O}_A^U)$ posed by dimension, i.e., the decision variable $\mathcal{F}_a = \{F_{a,n} \mid \forall o_n \in O\}$ is a $|O|$-dimension function set, we relax the problem $(\mathcal{O}_A^U)$ into $|O|$ optimization problems over $\forall F_{a,n} \in \mathcal{F}_a$ respectively. The object (30a) can be reformulated as

$$\sum_{o_n \in O} \max_{\widetilde{x}_n^m \sim F_a^n} \left(E[x_{b,n}^{truth}] - E[x_{a,n}^{truth}]\right)^2. \tag{31}$$

We respectively assign the disguise level and the stealth level to each object o_n, i.e., $\sum_{o_n \in O} \eta_n = \eta$ (η_n is the disguise level hidden behind DP noise in object o_n) and $\sum_{o_n \in O} \zeta_n = \zeta$ (ζ_n denotes the stealth level of avoiding defenders in object o_n). Thus, the constraints (30b) and (30c) can be relaxed to

$$D_{KL}(F_{a,n} \| F_{b,n}) \le \eta_n, \tag{32}$$

$$\int_{-\infty}^{+\infty} F_{a,n}(y) \cdot \log \frac{F_{a,n}^*(y)}{F_{b,n}(y)} dy \le \zeta_n. \tag{33}$$

According to (31), (32), and (33), the problem $(\mathcal{O}_A^U)$ can be relaxed as

$$(\mathcal{O}_A^{U,o_n}): \max_{\widetilde{x}_n^m \sim F_{a,n}} \left(E[x_{b,n}^{truth}] - E[x_{a,n}^{truth}]\right)^2, \tag{34a}$$

$$\text{s.t. } D_{KL}(F_{a,n} \| F_{b,n}) \le \eta_n, \tag{34b}$$

$$\int_{-\infty}^{+\infty} F_{a,n}(y) \cdot \log \frac{F_{a,n}^*(y)}{F_{b,n}(y)} dy \le \zeta_n, \tag{34c}$$

$$\int_{-\infty}^{+\infty} F_{a,n}(y) dy = 1, \tag{34d}$$

$$E[x_{b,n}^{truth}] = \int_{-\infty}^{+\infty} y \cdot F_{b,n}(y) dy, \tag{34e}$$

$$E[x_{a,n}^{truth}] = \int_{-\infty}^{+\infty} y \cdot F_{a,n}(y) dy, \tag{34f}$$

$$x_{a,n}^{truth} = \Big(\sum_{\widehat{u}_m \in \widehat{U}} \widehat{w}_m \cdot \widehat{x}_n^m + \sum_{\widetilde{u}_m \in \widetilde{U}} \widetilde{w}_m \cdot \widetilde{x}_n^m \Big) / \Big(\sum_{\widehat{u}_m \in \widehat{U}} \widehat{w}_m + \sum_{\widetilde{u}_m \in \widetilde{U}} \widetilde{w}_m \Big). \tag{34g}$$

Due to the problem $(\mathcal{O}_A^{U,o_n})$ is a functional optimization problem [46], i.e., $F_{a,n}$ is a density function, we use variational methods [47] to solve it and present the main result in the Theorem 3.

Theorem 3: Given the current defense strategy $F_{a,n}^*$ and the normal behavior $F_{b,n}$, the optimal solution of problem $(\mathcal{O}_A^{U,o_n})$ (i.e., the attack strategy for object o_n) is

$$F_{a,n} := \arg\max_{F_{a,n} \in \{F_{a,n}^{\min}, F_{a,n}^{\max}\}} (E[x_{b,n}^{truth}] - E[x_{a,n}^{truth}])^2, \tag{35}$$

where $x_{a,n}^{truth}$ is estimated by (34g), $E[x_{a,n}^{truth}]$ is calculated by (34f), and

$$F_{a,n}^{\min} = \frac{F_{b,n}(y) \cdot \left(\frac{F_{b,n}(y)}{F_{a,n}^*(y)}\right)^{\frac{\lambda_2}{\lambda_1}} \cdot \exp(-\frac{\theta_1}{\lambda_1} y - F_{b,n}(y))}{\int_{-\infty}^{+\infty} F_{b,n}(y) \cdot \left(\frac{F_{b,n}(y)}{F_{a,n}^*(y)}\right)^{\frac{\lambda_2}{\lambda_1}} \cdot \exp(-\frac{\theta_1}{\lambda_1} y - F_{b,n}(y)) dy}, \tag{36}$$

$$F_{a,n}^{\max} = \frac{F_{b,n}(y) \cdot \left(\frac{F_{b,n}(y)}{F_{a,n}^*(y)}\right)^{\frac{\lambda_2}{\lambda_1}} \cdot \exp(\frac{\theta_1}{\lambda_1} y - F_{b,n}(y))}{\int_{-\infty}^{+\infty} F_{b,n}(y) \cdot \left(\frac{F_{b,n}(y)}{F_{a,n}^*(y)}\right)^{\frac{\lambda_2}{\lambda_1}} \cdot \exp(\frac{\theta_1}{\lambda_1} y - F_{b,n}(y)) dy}. \tag{37}$$

The Lagrange multipliers λ_1 and λ_2 are the solutions of $D_{KL}(F_{a,n} \| F_{b,n}) = \eta_n$ and $\int_{-\infty}^{+\infty} F_{a,n}(y) \cdot \log \frac{F_{a,n}^*(y)}{F_{b,n}(y)} dy = \zeta_n$.

Proof: See supplemental file. □

2) Lower-Level Sub-Problem: When the attack strategy $\mathcal{F}_a^* = \{F_{a,n}^* \mid \forall o_n \in O\}$ is fixed, as detailed in Section III-C, the sub-problem (29g) can be solved by sequentially iterating the following two steps:

1) Fixing the normal worker weights $\widehat{W} = \{\widehat{w}_m \mid \forall \widehat{u}_m \in \widehat{U}\}$ and the malicious worker weights $\widetilde{W} = \{\widetilde{w}_m \mid \forall \widetilde{u}_m \in$

Algorithm 3: Local Minimax Point Calculating Algorithm.

Input : ϖ: the threshold for convergence; $\vartheta > 0$: the constant defined in Lemma 1; $0 < \rho < 1$: the penalty constant;
Output: $(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star})$: the local minimax point.

1 **for** $k = 0, 1$ **do**
2 $\quad \mathcal{F}_a^{*,k} = \{F_{a,n}^{*,k} \mid \forall o_n \in O\} \leftarrow$ Obtain current defense strategy through Algorithm 1;
3 $\quad \mathcal{F}_a^{k} = \{F_{a,n}^{k} \mid \forall o_n \in O\} \leftarrow$ Obtain current attack strategy using $\mathcal{F}_a^{*,k}$ through Algorithm 2;
4 **end**
5 **while** $\|\mathcal{F}_a^{*,k} - \mathcal{F}_a^{*,k-1}\| + \|\mathcal{F}_a^{k} - \mathcal{F}_a^{k-1}\| > \varpi$ **do**
6 $\quad$ **while** $\|\mathcal{F}_a^{k} - \mathcal{F}_a^{k-1}\| > \vartheta$ **do**
7 $\quad\quad \mathcal{F}_a^{*,k+1} \leftarrow$ Obtain current defense strategy through Algorithm 1;
8 $\quad\quad \mathcal{F}_a^{k+1} \leftarrow$ Obtain current attack strategy using $\mathcal{F}_a^{*,k+1}$ through Algorithm 2;
9 $\quad\quad k \leftarrow k + 1$;
10 $\quad$ **end**
11 $\quad \vartheta \leftarrow \rho \cdot \vartheta$;
12 $\quad k \leftarrow k + 1$;
13 **end**
14 $\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star} \leftarrow \mathcal{F}_a^{*,k}$; $\quad \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star} \leftarrow \mathcal{F}_a^{k}$;

$\widetilde{U}\}$, the truths $X_a^{truth} = \{x_{a,n}^{truth} \mid \forall o_n \in O\}$ are calculated by

$$x_{a,n}^{truth} = \frac{\sum_{\widehat{u}_m \in \widehat{U}} \widehat{w}_m \cdot \widehat{x}_n^m + \sum_{\widetilde{u}_m \in \widetilde{U}} \widetilde{w}_m \cdot \widetilde{x}_n^m}{\sum_{\widehat{u}_m \in \widehat{U}} \widehat{w}_m + \sum_{\widetilde{u}_m \in \widetilde{U}} \widetilde{w}_m}; \quad (38)$$

2) After obtaining the truths $x_{a,n}^{truth}$, the normal worker weights $\widehat{W}$ and the malicious worker weights $\widetilde{W}$ are estimated by

$$\widehat{w}_m = \log\Bigg\{ \sum_{o_n \in O} \Bigg(\sum_{\widehat{u}_m \in \widehat{U}} (\widehat{x}_n^m - x_{a,n}^{truth})^2 + \sum_{\widetilde{u}_m \in \widetilde{U}} (\widetilde{x}_n^m - x_{a,n}^{truth})^2 \Bigg) \Big/ \sum_{o_n \in O} (\widehat{x}_n^m - x_{a,n}^{truth})^2 \Bigg\}, \quad (39)$$

$$\widetilde{w}_m = \log\Bigg\{ \sum_{o_n \in O} \Bigg(\sum_{\widehat{u}_m \in \widehat{U}} (\widehat{x}_n^m - x_{a,n}^{truth})^2 + \sum_{\widetilde{u}_m \in \widetilde{U}} (\widetilde{x}_n^m - x_{a,n}^{truth})^2 \Bigg) \Big/ \sum_{o_n \in O} (\widetilde{x}_n^m - x_{a,n}^{truth})^2 \Bigg\}. \quad (40)$$

According to the above discussion, as shown in Algorithm 2, we propose an attack strategy algorithm that combines the variational method with the alternating optimization algorithm for solving the problem $(\mathcal{O}_A)$. We initialize the poisoning attack strategy $\mathcal{F}_a$ as $\mathcal{F}_b$, i.e., $F_{a,n} = F_{b,n}, \forall o_n \in O$ (line 1). At each step in the iteration, we alternately solve the upper-level problem $(\mathcal{O}_A^U)$ and the lower-level problem (29g), until obtaining the current attack strategy $\mathcal{F}_a$. In addition, the computational complexity of Algorithm 2 (i.e., attack strategy algorithm) is $O(|O| \times |U|)$.

C. Local Minimax Point of Defenders-Attackers Interaction

The problems $(\mathcal{O}_A)$ and $(\mathcal{O}_D)$ are bi-level optimizations [45], so finding global minimax point of the problem (16) is NP-hard in general [48]. With this concern, we aim to find the local (rather than global) minimax point within the sequential interaction between defenders and attackers. To this end, we introduce the fundamental definition of local minimax points [48], a critical concept in the minimax optimization problem, and then develop an effective iterative algorithm to find it.

Definition 10 (Local Minimax Points): A point $(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star})$ is said to be a local minimax point of the problem (16), if there exists $\vartheta_0 > 0$ and a function h satisfying $h(\vartheta) \to 0$ as $\vartheta \to 0$, such that for any $\vartheta \in (0, \vartheta_0]$ and any $(\mathcal{F}_a^*, \mathcal{F}_a)$ satisfying $\|\mathcal{F}_a^* - \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}\| \le \vartheta$ and $\|\mathcal{F}_a - \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}\| \le \vartheta$, we have

$$U(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \mathcal{F}_a) \le U(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}) \le \max_{\mathcal{F}_a : \|\mathcal{F}_a - \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}\| \le h(\vartheta)} U(\mathcal{F}_a^*, \mathcal{F}_a). \quad (41)$$

According to Definition 10, Jin et al. [48] introduce a crucial lemma (i.e., Lamma 1) for acquiring local minimax point, and its detailed lemma is given below.

Lemma 1: A point $(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star})$ is a local minimax point of the problem (16) if and only if $\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}$ is a local maximum of function $U(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \cdot)$, and there exists $\vartheta_0 > 0$ such that $\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}$ is a local minimum of function K_ϑ for all $\vartheta \in (0, \vartheta_0]$ where $K_\vartheta(\mathcal{F}_a^*) := \max_{\mathcal{F}_a : \|\mathcal{F}_a - \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}\| \le \vartheta} U(\mathcal{F}_a^*, \mathcal{F}_a)$.

To further illustrate the relationship between local minimax point and Nash equilibrium, we give the definition of local Nash equilibrium and then showcase an important lemma (i.e., Lemma 2) proposed by Jin et al. in [48].

Definition 11 (Local Nash Equilibrium): A point $(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star})$ is said to be a local Nash equilibrium of the problem (16) (or the game defined in Section V-A) if and only if there exists $\vartheta > 0$ such that for any $(\mathcal{F}_a^*, \mathcal{F}_a)$ satisfying $\|\mathcal{F}_a^* - \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}\| \le \vartheta$ and $\|\mathcal{F}_a - \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}\| \le \vartheta$, we have

$$U(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \mathcal{F}_a) \le U(\boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{*,\star}, \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}) \le U(\mathcal{F}_a^*, \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}). \quad (42)$$

Lemma 2: Any local Nash equilibrium is a local minimax point.

According to the Lemma 1, we propose a local minimax point calculating algorithm for finding the local minimax point of the game described in Section V-A as shown in Algorithm 3. At each step in the iteration, we alternately solve problems $(\mathcal{O}_D)$ and $(\mathcal{O}_A)$, i.e., perform Algorithm 1 (line 7) and Algorithm 2 (line 8) alternately. Then, we judge whether the condition $\|\mathcal{F}_a - \boldsymbol{\mathcal{F}}_{\boldsymbol{a}}^{\star}\| \le \vartheta$ (defined in Lamma 1) is satisfied (line 6). If so, we reduce the constant ϑ (line 11), and then evaluate whether the convergence condition is satisfied (line 5). Through iterating these steps, we can find the local minimax point of defenders-attackers interaction. Furthermore, according to the computational complexity of Algorithm 1 and Algorithm 2 are $O(G_{\max} \times SN \times |O| \times |U|)$ and $O(|O| \times |U|)$, respectively, we can obtain the computational complexity of Algorithm 3 (i.e., the proposed Stackelberg-game-based defense approach) is $O(G_{\max} \times SN \times |O| \times |U|)$.

TABLE II
SAMPLE RECORDS OF "ANGER" IN THE EMOTION DATASET

WorkerID	ObjectID	Response	Ground-Truth
A1AVJRFM6L0RN8	594	25	37
ADAGUJNWMEPT6	594	80	37

VII. PERFORMANCE EVALUATION

The proposed game-based defense approach enables us to find an effective defense strategy to defend against the powerful data poisoning attack. In this section, we evaluate the attack performance of the proposed powerful data poisoning attack and some existing data poisoning attacks against the DP-based privacy-preserving crowdsensing systems, and then evaluate the defense performance of the proposed defense strategy to resist these attacks.

A. Experiment Setup

1) Datasets: We use two datasets in our experiments: SynData (a synthetic dataset) and Emotion (a real-world dataset). Below, we present an overview of these two datasets.

SynData Dataset: The synthetic dataset, named SynData, contains 200,000 records generated by 500 workers for 400 objects. These real sensory data are drawn from normal distributions, i.e., $\widehat{x}_n^m \sim N(\widehat{\mu}_n, \widehat{\sigma}_m^2), \forall o_n \in O, \forall \widehat{u}_m \in \widehat{U}$ ($\widehat{\mu}_n$ denotes the truth of object o_n and $\widehat{\sigma}_m^2$ is the reliability of worker $\widehat{u}_m$). Similar to [5], [15], we sample $\widehat{\mu}_n$ and $\widehat{\sigma}_m^2$ from uniform distributions $U(20, 30)$ and $U(0, 30)$, respectively. That is, $\widehat{\mu}_n \sim U(20, 30)$ and $\widehat{\sigma}_m^2 \sim U(0, 30)$.

Emotion Dataset [49]: The real-world dataset, named Emotion, consists of 7,000 records produced by 30 workers for 700 objects. Each record represents the degree of emotion (e.g., anger, disgust, and fear) in a text and ranges from -100 to 100. Table II showcases two randomly selected records from the Emotion dataset.

2) Comparisons of Data Poisoning Attacks: For the experimental evaluation of the proposed game-based defense approach, five different data poisoning attacks were applied. The experiments were performed on a laptop with an Intel i7-8750H 2.20GHz CPU and 16GB memory. Next, we outline these attacks.

1) *Disguise-based Data Poisoning Attack (DDPA) [5]:* DDPA is a state-of-the-art data poisoning attack against DP-based privacy-preserving crowdsensing systems, leveraging DP noise to disguise malicious behavior.
2) *Optimization-based Attack (OA) [15]:* This attack is formulated as an optimization problem that maximizes the introduced deviation by attackers. In addition, OA is designed for crowdsensing systems without privacy protection.
3) *Substitution-based approach for black-box data poisoning attack (SubPac) [50]:* This attack is formulated as a bi-level min-max optimization problem where the outer problem is to find the optimal attack strategies and the inner problem is to optimize data aggregation.
4) *Random Attack (RA):* In RA, for object $o_n \in O$, each malicious worker selects a value from the uniform distribution $U(x_n^{\min}, x_n^{\max})$ as malicious sensory data, where $x_n^{\min} = \arg\min\{\widehat{x}_n^m \mid \forall \widehat{u}_m \in \widehat{U}\}$ (or $x_n^{\max} = \arg\max\{\widehat{x}_n^m \mid \forall \widehat{u}_m \in \widehat{U}\}$) is the minimum (or maximum) sensory data provided by normal workers for object o_n. That is, $\widetilde{x}_n^m \sim U(x_n^{\min}, x_n^{\max}), \forall o_n \in O, \forall \widetilde{u}_m \in \widetilde{U}$.
5) *Maximum Gain Attack (MGA):* In MGA, for object $o_n \in O$, the malicious sensory data shared by malicious workers is the optimal solution of problem $\max_{\widetilde{x}_n^m} (\widehat{x}_n^m - x_{b,n}^{truth})^2$. That is, for object $o_n \in O$, the malicious sensory data $\widetilde{x}_n^m$ satisfies $(\widetilde{x}_n^m - x_{b,n}^{truth})^2 \geq (\widehat{x}_n^m - x_{b,n}^{truth})^2, \forall \widehat{x}_n^m \in \widehat{X}$.

3) Comparisons of Defenses: We evaluate our proposed defense approach by comparing it with two classical defense mechanisms. Next, we detail MWA [15] and RobTD [22].

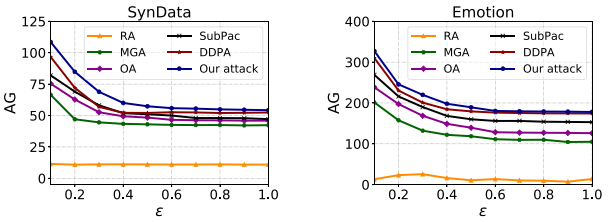
1) MWA [15] utilizes (9) to update worker weights, and the truths can be updated through the following steps. First, MWA sorts the sensory data uploaded by all workers and partitions them into L groups. Second, MWA estimates the truths for each group according to (8), obtaining L truths. Finally, MWA takes the median of these L truths as the inferred truth.
2) RobTD [22] estimates the error bias and variance of each worker, and then removes workers with high error bias and variance to mitigate the influence of malicious workers. After filtering out the malicious workers, RobTD updates the worker weights and the truths utilizing (9) and (8).

4) Evaluation Metrics: Referring to Definition 6, we propose the attack gain metric to quantify the damage caused by malicious workers to crowdsensing systems. Since the attackers aim to maximize the attack gain (i.e., skew from the outputs of truth discovery methods as much as possible), higher attack gain indicates more effective attacks. Conversely, since the attack gain for attackers is the defense loss for defenders, a smaller attack gain (defense loss) corresponds to a more successful defense strategy. Therefore, we use the attack gain (AG) to evaluate the effectiveness of our data poisoning attack and defense.

B. Effectiveness of Our Attack Strategy

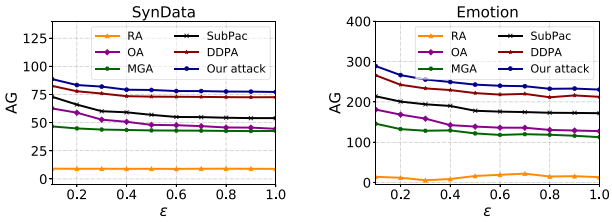
We study the performance of both our and other existing data poisoning attacks against DP-based privacy-preserving crowdsensing systems without defenders. Furthermore, we evaluate the effectiveness of these attacks by varying the percentage of known workers (defined in Definition 7) and the percentage of malicious workers (defined in Definition 8). By conducting these comparisons, we can evaluate the effectiveness of our attacks and the potential risks DP poses to crowdsensing.

1) Impacts of the Privacy Budget: In Fig. 2, we show the AG obtained by the proposed data poisoning attack and the competitor attacks against the DP-based privacy-preserving crowdsensing without defenders, where the Laplace mechanism is employed to achieve DP. First, we observe that our attack could effectively employ DP noise injected by the Laplace mechanism



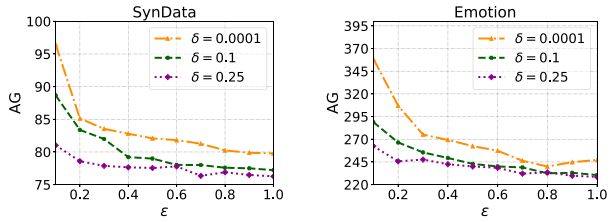
(a) Impact of ε on attack gain (b) Impact of ε on attack gain

Fig. 2. The attack gain (AG) vs. the privacy budget ε of our data poisoning attack and competitor attacks against the ε-DP-based privacy-preserving crowdsensing with Laplace mechanism.



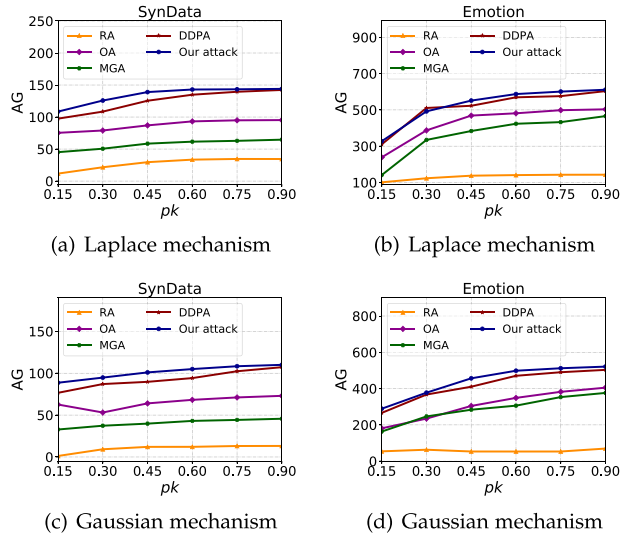
(a) AG vs. ε, where $\delta = 0.1$ (b) AG vs. ε, where $\delta = 0.1$

Fig. 3. The attack gain (AG) varying with the privacy budget ε of our data poisoning attack and competitor attacks against the $(\varepsilon, 0.1)$-DP-based privacy-preserving crowdsensing with Gaussian mechanism.



(a) Impact of (ε, δ) on attack gain (b) Impact of (ε, δ) on attack gain

Fig. 4. The attack gain (AG) vs. the privacy budget ε of our data poisoning attack against the (ε, δ)-DP-based privacy-preserving crowdsensing with Gaussian mechanism, where the relaxation parameter $\delta = 0.0001, 0.1, 0.25$.



(a) Laplace mechanism (b) Laplace mechanism

(c) Gaussian mechanism (d) Gaussian mechanism

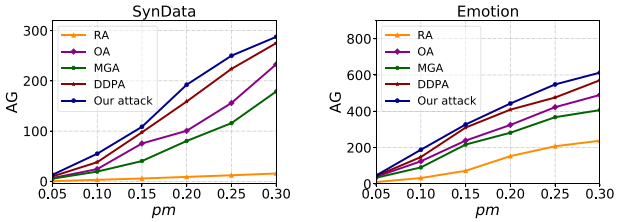
Fig. 5. The attack gain (AG) vs. the percentage of known workers (pk) of our data poisoning attack and competitor attacks against the DP-based privacy-preserving crowdsensing, where DP achieved by (a)(b) the Laplace mechanism and (c)(d) the Gaussian mechanism.

to bypass truth discovery methods. For instance, on the SynData, our attack increases AG to 108.52 when $\varepsilon = 0.1$ and improves upon 99.9% on AG as ε decreases from 1.0 to 0.1. This advantage is attributed to the fact that our attack could leverage DP noise to cloak its malicious behaviors. Second, our attack performs roughly equal to the DDPA: AG of our attack is approximately equivalent to that of DDPA for the given privacy budget ε. This situation arises due to the absence of defenders within the DP-based privacy-preserving crowdsensing systems, enabling both our attack and DDPA to avoid detection.

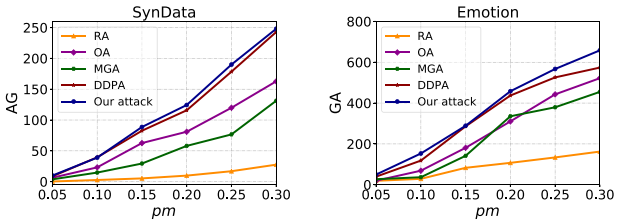
Fig. 3 showcases the superior performance of our attack against the DP-based privacy-preserving crowdsensing systems on both SynData and Emotion, utilizing the Gaussian mechanism for achieving DP: For given privacy budget ε and relaxation parameter δ, AG obtained by our attack is higher than that of SubPac, OA, RA, and MGA. For instance, when $\varepsilon = 1.0, \delta = 0.1$, our attack improves AG by 73.50% on the SynData (80.76% on the Emotion). As expected, AG obtained by our attack and DDPA become roughly equal when no defenders are deployed in crowdsensing. Moreover, Fig. 4 indicates that the performance of our attack improves with a decrease in the relaxation parameter δ, i.e., the improvement on AG reaches upon 19.07% on the SynData (36.67% on the Emotion) as δ decreases from 0.25 to 0.0001 when $\varepsilon = 0.1$. This is because reducing δ strengthens privacy protection, implying that more DP noise is added to the sensory data, which makes it easier for attackers to hide behind the noise.

2) Impacts of the Percentage of Known Workers: Fig. 5 shows AG obtained by our attack and competitor attacks varying with the percentage of known workers pk on SynData and Emotion. Fig. 5 indicates that, for any pk, our attack significantly boosts AG by 96.81 on the SynData (226.2936 on the Emotion) against the Laplace mechanism, and boosts AG by 87.4503 on the SynData (234.8072 on the Emotion) against the Gaussian mechanism. Furthermore, Fig. 5 also indicates that the performance of all attacks improves with an increase in pk. That is, the AG of all attacks increases with the rising value of pk. This occurs because higher values of the percentage of known workers pk enable attackers to more accurately infer the truths before attacks X_b^{truth}, thereby improving their attack strategy.

3) Impacts of the Percentage of Malicious Workers: Fig. 6 displays AG achieved by our attack and competitor attacks as the percentage of malicious workers pm is varied. As expected, the performance of all attacks improves as parameter pm increases: With pm ranging from 0.05 to 0.3, AG exhibits a minimum 274.5865in the Laplace mechanism (238.1736 in the Gaussian mechanism) increase on both datasets SynData and Emotion. Besides, as pm increases, we observe a more pronounced improvement in the performance of both our attack and DDPA. This arises from the fact that, as attackers' capabilities increase (i.e., manipulating more malicious workers), the probability of successful attacks rises.

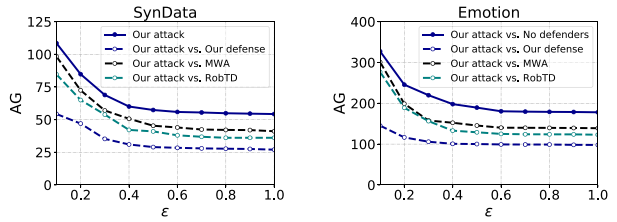


(a) Laplace mechanism (b) Laplace mechanism

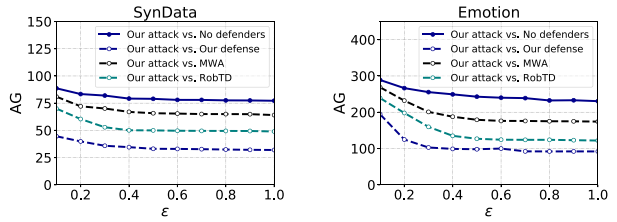


(c) Gaussian mechanism (d) Gaussian mechanism

Fig. 6. The attack gain (AG) vs. the percentage of malicious workers (pm) of our data poisoning attack and competitor attacks against the DP-based privacy-preserving crowdsensing, where DP achieved by (a)(b) the Laplace mechanism and (c)(d) the Gaussian mechanism.



(a) Laplace mechanism (b) Laplace mechanism



(c) Gaussian mechanism (d) Gaussian mechanism

Fig. 7. The attack gain (AG) vs. the privacy budget ε of our defense approach and competitor defenses resist our data poisoning attack.

C. *Effectiveness of Our Defense Strategy*

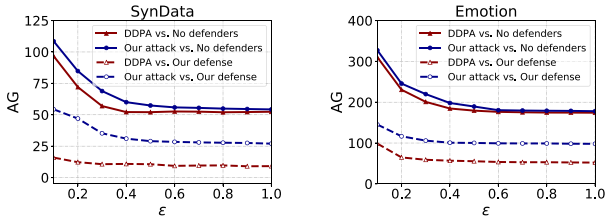
In Fig. 7, we show the defense performance of the proposed defense approach and the competitor defense mechanisms. As expected, the proposed defense approach outperforms MWA and RobTD. For the same privacy budget ε, the reduction in AG achieved by the proposed defense approach is significantly greater than that achieved by MWA and RobTD. For example, when the privacy budget $\varepsilon = 0.1$, our defense reduces AG by 181.9455 on the SynData against the Laplace mechanism, while MWA and RobTD reduce AG by 26.3168 and 50.5390, respectively. The superiority of our defense approach is attributed to the fact that MWA and RobTD cannot detect malicious workers hidden behind the DP noise. In addition, the superiority of our defense approach increases as the privacy budget ε decreases. For instance, our defense reduces AG by 79.9844 on the SynData against the Laplace mechanism when $\varepsilon = 1.0$, and reduces AG by 181.9455 when $\varepsilon = 0.1$. This is because a smaller ε results in a greater amount of DP noise being added, allowing malicious workers to better disguise themselves. Furthermore, we compared the computational complexity between the proposed defense approach and the competitor defense mechanisms, as shown in Table III. Particularly, the computational complexity of the proposed defense is slightly higher than that of MWA and RobTD when $\log |U| < G_{\max} \times SN$. However, this disadvantage becomes negligible with the increase of $|U|$.

TABLE III
COMPUTATIONAL COMPLEXITY OF DEFENSES

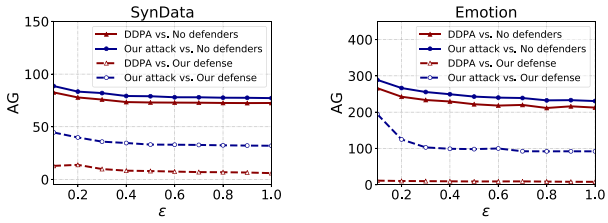
Defense Strategy	Computational Complexity
MWA	$O(\|O\| \times \|U\| \times \log \|U\|)$
RobTD	$O(\|O\| \times \|U\| \times \log \|U\|)$
Our	$O(G_{max} \times SN \times \|O\| \times \|U\|)$

In this part, we also consider four combinations to evaluate the robustness of our proposed defense approach: (DDPA vs. No defenders), (Our attack vs. No defenders), (DDPA vs. Our defense), and (Our attack vs. Our defense). The combinations (DDPA vs. No defenders) and (Our attack vs. No defenders) represent DDPA (or our data poisoning attack) against the DP-based privacy-preserving crowdsensing without defenders, which are the attackers' favorite settings due to the absence of defenders. Inversely, the defenders prefer the combination (DDPA vs. Our defense), i.e., DDPA against the DP-based privacy-preserving crowdsensing equipped with our defense strategy, as the attackers are unaware of the defense strategy and inability to bypass the defenders. However, neither of these settings represents a stable state. In the combinations (DDPA vs. No defenders) and (Our attack vs. No defenders), the attackers can employ the powerful data poisoning attack (or DDPA) to achieve the maximum AG. In the combination (DDPA vs. Our defense), the attackers obtain minimum AG since their malicious behaviors can be detected by the defenders. Hence, the combination (Our attack vs. Our defense), i.e., the game-based defense approach helps us find an effective defense strategy to mitigate the powerful data poisoning attack, is a stable equilibrium for both defenders and attacks. By comparing these four combinations, we evaluate the effectiveness of our defense strategy in mitigating data poisoning attacks.

1) Impacts of the Privacy Budget: Fig. 8 illustrates the utilization of attack and defense strategies by both defenders and attackers in their interactions. First, we observe that our defense strategy can effectively resist DDPA by comparing combinations (DDPA vs. No defenders) and (DDPA vs. Our defense): AG experiences a significant reduction for any specified privacy budget ε. For instance, when $\varepsilon = 0.1$, AG reduces 80.7819 in the Laplace mechanism (69.7213in the Gaussian mechanism) on the dataset Syndata and decreases 211.6281 in the Laplace mechanism (254.6977in the Gaussian mechanism) on the dataset Emotion. Second, our defense strategy can mitigate our data poisoning attack by comparing combinations (Our attack vs.

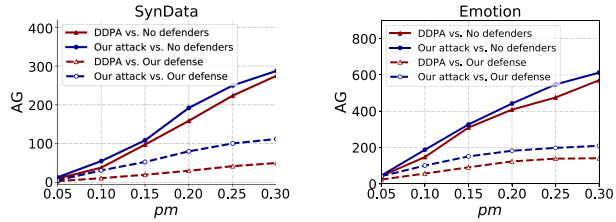


(a) Laplace mechanism (b) Laplace mechanism

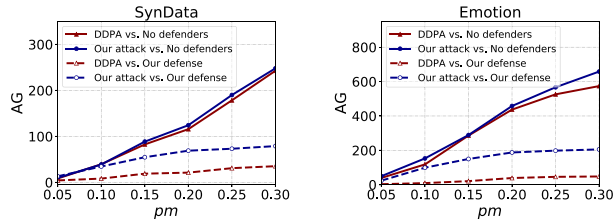


(c) Gaussian mechanism (d) Gaussian mechanism

Fig. 8. The attack gain (AG) vs. the privacy budget ε for the attackers. The various lines depict combinations of our ($\circ$) and other existing ($\triangle$) data poisoning attacks against the DP-based privacy-preserving crowdsensing with ($--$) and without ($-$) our defense strategy.

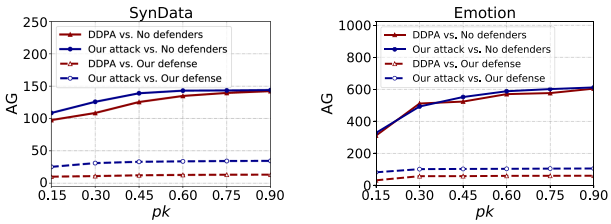


(a) Laplace mechanism (b) Laplace mechanism

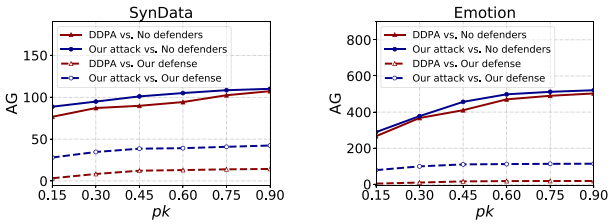


(c) Gaussian mechanism (d) Gaussian mechanism

Fig. 10. The attack gain (AG) vs. the percentage of malicious workers (pm). The various lines depict combinations of our ($\circ$) and other existing ($\triangle$) data poisoning attacks against the DP-based privacy-preserving crowdsensing with ($--$) and without ($-$) our defense strategy.



(a) Laplace mechanism (b) Laplace mechanism



(c) Gaussian mechanism (d) Gaussian mechanism

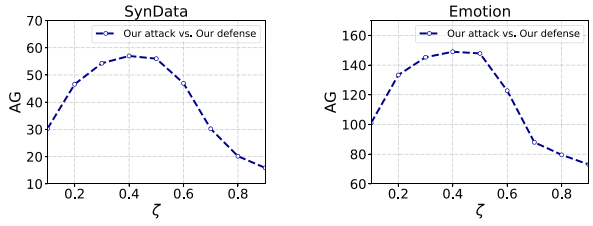
Fig. 9. The attack gain (AG) vs. the percentage of known workers (pk). The various lines depict combinations of our ($\circ$) and other existing ($\triangle$) data poisoning attacks against the DP-based privacy-preserving crowdsensing with ($--$) and without ($-$) our defense strategy.

No defenders) and (Our attack vs. Our defense): AG reduces 54.2027 in the Laplace mechanism (44.0708 in the Gaussian mechanism) on the dataset SynData and decreases 181.9455 in the Laplace mechanism (94.5856 in the Gaussian mechanism) on the dataset Emotion when $\varepsilon = 0.1$. Third, we also notice that our data poisoning attack can partially bypass our defense strategy by comparing combinations (Our attack vs. Our defense) and (DDPA vs. Our defense): Our attack improves AG by a minimum of 19.9095 on the Syndata (45.8921 on the Emotion) compared to DDPA.

2) Impacts of the Percentage of Known Workers: Fig. 9 shows the effectiveness of our defense strategy varies with the percentage of known workers, denoted as pk. First, we observe that our defense strategy can effectively defend against DDPA for any given pk by comparing the combinations (DDPA vs. No defenders) and (DDPA vs. Our defense): AG significantly decreases for any given pk . For example, AG reduces 87.5258 in the Laplace mechanism on the dataset SynData when $pk = 0.15$. Second, our defense strategy can defend against our poisoning attacks for any given pk by comparing the combinations (Our attack vs. No defenders) and (Our attack vs. Our defense): AG reduces 83.3500 in the Laplace mechanism on the dataset SynData when $pk = 0.15$. Third, the effectiveness of our defense strategy is robust by observing the combination (DDPA vs. Our defense) and the combination (Our attack vs. Our defense): AG does not undergo significant changes with variations in pk.

3) Impacts of the Percentage of Malicious Workers: Fig. 10 illustrates the effectiveness of our defense strategy varies with the percentage of malicious workers, denoted as pm. First, our defense strategy can defend against DDPA, since the AG of the combination (DDPA vs. No defenders) is significantly smaller than the AG of the combination (DDPA vs. Our defense) for any given pm. Second, our defense strategy can defend against our poisoning attack, as the AG of the combination (Our attack vs. No defenders) is significantly smaller than the AG of the combination (Our attack vs. Our defense) for any given pm. Third, the effectiveness of our defense strategy is robust by observing the combinations (DDPA vs. Our defense) and (Our attack vs. Our defense): Although the value of AG increases slowly with the increase of pm, it still can significantly reduce the bias caused by DDPA and our poisoning attack.

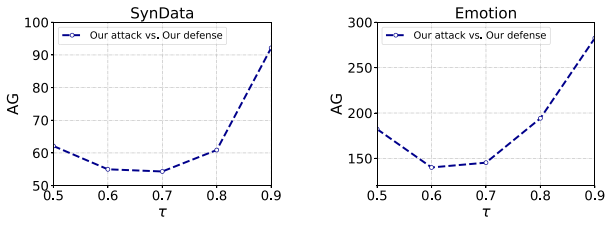
4) Impacts of the Stealth Level: Fig. 11 displays AG obtained through our data poisoning attack against the DP-based privacy-preserving crowdsensing with our defense, as it varies with the stealth level ζ. The parameter ζ exclusively influences the effectiveness of our attack in evading defenders and is absent



(a) Laplace mechanism (b) Laplace mechanism

(c) Gaussian mechanism (d) Gaussian mechanism

Fig. 11. The effectiveness of our defense strategy varies with the stealth level (ζ). The attack gain (AG) vs. the stealth level (ζ) of our data poisoning attacks against the DP-based privacy-preserving crowdsensing with defenders, where DP is achieved by (a)(b) the Laplace mechanism and (c)(d) Gaussian mechanism.



(a) Laplace mechanism (b) Laplace mechanism

(c) Gaussian mechanism (d) Gaussian mechanism

Fig. 12. The effectiveness of our defense strategy varies with the parameter of hypothesis testing (τ). The attack gain (AG) vs. the parameter (ζ) of our data poisoning attack against the DP-based privacy-preserving crowdsensing with defenders, where DP is achieved by (a)(b) the Laplace mechanism and (c)(d) Gaussian mechanism.

in other attacks, thereby Fig. 11 only shows the results for the combination (Our attack vs. Our defense). We notice that AG initially rises and then declines with varying the stealth level ζ. That is, the performance of our defense initially decreases and subsequently improves as ζ adds. This phenomenon arises from the fact that a larger ζ implies that malicious workers are readily detected by defenders, thereby reducing the damage caused by our attack. Inversely, a smaller ζ implies the enhanced capability to evade the defenders, i.e., reducing the expected log-likelihood ration test, but it can mitigate the damage caused by our attack when the value of ζ is too small.

5) Impacts of the Parameter of Hypothesis Testing: Fig. 12 illustrates the performance of our defense strategy as it varies with the hypothesis testing parameter, denoted as τ, which quantifies the likelihood of the worker being malicious. Fig. 12 presents the results for the combination (Our attack vs. Our defense) due to the influence of parameter τ on the effectiveness of our defense strategy. As expected, AG initially decreases and subsequently increases with the increasing value of parameter τ. That is, the performance of our defense initially improves and then decreases with the increase in τ. This occurs because higher τ could lead the defenders to mistake some malicious workers for normal, while lower τ could result in misidentifying normal workers as malicious.

VIII. Conclusion

We have proposed a Stackelberg-game-based defense approach, for resisting powerful data poisoning attack in the DP-based privacy-preserving crowdsensing systems. It reveals that even with the attackers being aware of the defense strategy, the defenders can still effectively resist the data poisoning attack launched by those powerful attackers. They can be easily adapted to other DP-based privacy-preserving scenarios, such as location-based services, to mitigate the damage caused by DP. For the future work, we will investigate a stronger data poisoning attack that disguises itself within the noise introduced by shuffle DP. This is different compared to hiding behind DP noise because shuffle DP can provide strict privacy protection by adding a small amount of noise.

References

[1] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu, "CitySee: Urban CO_2 monitoring with sensors," in *Proc. 2012 IEEE Int. Conf. Comput. Commun.*, 2012, pp. 1611–1619.

[2] F.-Y. Wang, "Scanning the issue and beyond: Crowdsourcing for field transportation studies and services," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 1–8, Feb. 2015.

[3] Z. Wang et al., "Towards privacy-driven truthful incentives for mobile crowdsensing under untrusted platform," *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 1198–1212, Feb. 2023.

[4] X. Pang, Z. Wang, D. Liu, J. C. S. Lui, Q. Wang, and J. Ren, "Towards personalized privacy-preserving truth discovery over crowdsourced data streams," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, pp. 327–340, Feb. 2022.

[5] Z. Li, Z. Zheng, S. Guo, B. Guo, F. Xiao, and K. Ren, "Disguised as privacy: Data poisoning attacks against differentially private crowdsensing systems," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5155–5169, Sep. 2023.

[6] Z. Zheng, Z. Li, H. Jiang, L. Y. Zhang, and D. Tu, "Semantic-aware privacy-preserving online location trajectory data sharing," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2256–2271, 2022.

[7] C. Huang, D. Liu, A. Yang, R. Lu, and X. Shen, "Multi-client secure and efficient DPF-based keyword search for cloud storage," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 1, pp. 353–371, Jan./Feb. 2024.

[8] J. Hou et al., "Data protection: Privacy-preserving data collection with validation," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 4, pp. 3422–3438, Jul./Aug. 2024.

[9] C. Huang, W. Wang, D. Liu, R. Lu, and X. Shen, "Blockchain-assisted personalized car insurance with privacy preservation and fraud resistance," *IEEE Trans. Veh. Technol*, vol. 72, no. 3, pp. 3777–3792, Mar. 2023.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Theory Cryptogr. Conf.*, Springer, 2006, pp. 265–284.

[11] Y. Li et al., "Towards differentially private truth discovery for crowd sensing systems," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 1156–1166.

[12] G. Xu et al., "Catch you if you deceive me: Verifiable and privacy-aware truth discovery in crowdsensing systems," in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, 2020, pp. 178–192.
[13] P. Sun et al., "Towards personalized privacy-preserving incentive for truth discovery in mobile crowdsensing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 352–365, Jan. 2022.
[14] Y. Li et al., "An efficient two-layer mechanism for privacy-preserving truth discovery," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1705–1714.
[15] M. Fang, M. Sun, Q. Li, N. Z. Gong, J. Tian, and J. Liu, "Data poisoning attacks and defenses to crowdsourcing systems," in *Proc. Web Conf.*, 2021, pp. 969–980.
[16] C. Miao, Q. Li, L. Su, M. Huai, W. Jiang, and J. Gao, "Attack under disguise: An intelligent data poisoning attack mechanism in crowdsourcing," in *Proc. 2018 World Wide Web Conf.*, 2018, pp. 13–22.
[17] H. Zhang and M. Li, "Multi-round data poisoning attack and defense against truth discovery in crowdsensing systems," in *Proc. 23rd IEEE Int. Conf. Mobile Data Manage.*, 2022, pp. 109–118.
[18] Y. Zhao, X. Gong, F. Lin, and X. Chen, "Data poisoning attacks and defenses in dynamic crowdsourcing with online data quality learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 5, pp. 2569–2581, May 2023.
[19] C. Miao, Q. Li, H. Xiao, W. Jiang, M. Huai, and L. Su, "Towards data poisoning attacks in crowd sensing systems," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2018, pp. 111–120.
[20] R. Fujimoto and N. Kamiyama, "Poisoning attacks in crowdsensing over multiple areas," in *Proc. 2022 IEEE Glob. Commun. Conf.*, 2022, pp. 68–73.
[21] Z. Zheng, Z. Li, C. Huang, S. Long, M. Li, and X. Shen, "Data poisoning attacks and defenses to LDP-based privacy-preserving crowdsensing," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 5, pp. 4861–4878, Sep./Oct. 2024.
[22] Z. Huang, M. Pan, and Y. Gong, "Robust truth discovery against data poisoning in mobile crowdsensing," in *Proc. 2019 IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.
[23] H. Zhang, M. Li, Y. Sun, and G. Qu, "Robust truth discovery against multi-round data poisoning attacks," in *Wireless Algorithms, Systems, and Applications*. Berlin, Germany: Springer, 2022, pp. 258–270.
[24] J. Giraldo, A. Cardenas, M. Kantarcioglu, and J. Katz, "Adversarial classification under differential privacy," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020.
[25] X. Cao, J. Jia, and N. Z. Gong, "Data poisoning attacks to local differential privacy protocols," in *Proc. 30th {USENIX} Secur. Symp.*, 2021, pp. 947–964.
[26] A. Cheu, A. Smith, and J. Ullman, "Manipulation attacks in local differential privacy," in *Proc. 2021 IEEE Symp. Secur. Privacy*, 2021, pp. 883–900.
[27] Y. Wu, X. Cao, J. Jia, and N. Z. Gong, "Poisoning attacks to local differential privacy protocols for key-value data," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 519–536.
[28] X. Li, N. Z. Gong, N. Li, W. Sun, and H. Li, "Fine-grained poisoning attacks to local differential privacy protocols for mean and variance estimation," in *Proc. 32nd {USENIX} Secur. Symp.*, 2023, pp. 1739–1756.
[29] A. Cheu and M. Zhilyaev, "Differentially private histograms in the shuffle model from fake users," in *Proc. 2022 IEEE Symp. Secur. Privacy*, 2022, pp. 440–457.
[30] J. Imola, A. R. Chowdhury, and K. Chaudhuri, "Robustness of locally differentially private graph analysis against poisoning," 2022, *arXiv:2210.14376*.
[31] J. Giraldo, A. A. Cárdenas, M. Kantarcioglu, and J. Katz, "Adversarial classification under differential privacy," in *Proc. 27th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–18.
[32] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Theory Cryptogr. Conf.*, 2006, pp. 265–284.
[33] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014.
[34] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. 2014 ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 1187–1198.
[35] Y. Li, H. Sun, and W. H. Wang, "Towards fair truth discovery from biased crowdsourced answers," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 599–607.
[36] Y. Li et al., "Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 1986–1999, Aug. 2016.
[37] S. Giulini and M. Sanguineti, "Approximation schemes for functional optimization problems," *J. Optim. Theory Appl.*, vol. 140, pp. 33–54, 2009.
[38] V. Kurkova and M. Sanguineti, "Comparison of worst case errors in linear and neural network approximation," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 264–275, Jan. 2002.
[39] H. E. Salzer, "Hermite polynomials'," *J. Res. Nat. Bur. Standards*, vol. 48, no. 2, 1952, Art. no. 111.
[40] H. HasanÖrkcü, "Subset selection in multiple linear regression models: A hybrid of genetic and simulated annealing algorithms," *Appl. Math. Comput.*, vol. 219, no. 23, pp. 11 018–11 028, 2013.
[41] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statist. Sci.*, vol. 8, no. 1, pp. 10–15, 1993.
[42] S. Mirjalili and S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Berlin, Germany: Springer, 2019, pp. 43–55.
[43] B. L. Miller et al., "Genetic algorithms, tournament selection, and the effects of noise," *Complex Syst.*, vol. 9, no. 3, pp. 193–212, 1995.
[44] H. Boche and S. Stanczak, "The kullback–leibler divergence and nonnegative matrices," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5539–5545, Dec. 2006.
[45] J. Nie, L. Wang, and J. J. Ye, "Bilevel polynomial programs and semidefinite relaxation methods," *SIAM J. Optim.*, vol. 27, no. 3, pp. 1728–1757, 2017.
[46] A. Alessandri, C. Cervellera, and M. Sanguineti, "Functional optimal estimation problems and their solution by nonlinear approximation schemes," *J. Optim. Theory Appl.*, vol. 134, pp. 445–466, 2007.
[47] M. Struwe and M. Struwe, *Variational Methods*, vol. 991. Berlin, Germany: Springer, 2000.
[48] C. Jin, P. Netrapalli, and M. Jordan, "What is local optimality in nonconvex-nonconcave minimax optimization?," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 4880–4889.
[49] R. Snow, B. O'connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast–but is it good? Evaluating non-expert annotations for natural language tasks," in *Proc. 2008 Conf. Empirical Methods Natural Lang. Process.*, 2008, pp. 254–263.
[50] P. Chen, Y. Yang, D. Yang, H. Sun, Z. Chen, and P. Lin, "Black-box data poisoning attacks on crowdsourcing," in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, 2023, pp. 2975–2983.