# Lotus: Evolutionary Blind Regression over Noisy Crowdsourced Data

Chao Li*, Shan Chang*, Hongzi Zhu†, Hang Chen* and Ting Lu*
*School of Computer Science and Technology, Donghua University, Shanghai, China
†Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
chaoli@mail.dhu.edu.cn, changshan@dhu.edu.cn, hongzi@cs.sjtu.edu.cn, chenhang@mail.dhu.edu.cn, luting@dhu.edu.cn

***Abstract*—In mobile crowd sensing (MCS) applications, a public model of a system or phenomenon is expected to be derived from sensory data, i.e., observations, collected by mobile device users, through regression modeling. Unique features of MCS data bring the regression task new challenges. First, observations are error-prone and private, making it of great difficulty to derive an accurate model without acquiring raw data. Second, observations are non-stationary and opportunistically generated, calling for an evolutionary model updating mechanism. Last, mobile devices are resource-constrained, posing an urgent demand for lightweight regression schemes. In this paper, we propose an evolutionary blind regression scheme, called *Lotus*, in MCS settings. The core idea is first to select a 'maximum-safe-subset' of observations locally stored over all participants, which refers to finding a subset containing half of observations, such that the corresponding regression model has a minimum value of residual sum of squares. It implies the inconsistency between observations in the subset is minimized. Since such a maximum-safe-subset selection problem is NP-hard, a distributed greedy hill-climbing algorithm is proposed. Then, based on the resulted regression model, more observations are checked. Selected ones will be used to refine the model. With observations constantly coming, newly selected 'safe' observations are used to make the model evolved. To preserve data privacy, a *one-time pad masking* mechanism, and a *blocking scheme* are integrated into the process of regression estimation. Intensive theoretical analysis and extensive trace-driven simulations are conducted and the results demonstrate the efficacy of the Lotus design.**

***Index Terms*—mobile crowd sensing; blind regression; model evolution; outlier; opportunistic sensing; non-stationary**

## I. Introduction

The paradigm of mobile crowd sensing (MCS) empowers ordinary people to contribute data sensed or generated from their mobile devices, e.g., mobile phones, smart vehicles, wearable devices, etc. Such sensory data, called observations, can be aggregated and fused on a server for large-scale sensing or community intelligence mining, for example, smart home controlling [1], air pollution estimation [2], and biomedical data based health model forecasting [3], [4].

One of the commonly used statistical learning methods is regression, which can be used to estimate the relationship between a dependent variable and multiple independent variables based on collected MCS data. However, MCS application scenarios pose four rigid requirements to a practical regression estimator as follows. *1) Supreme reliability upon outliers:* untrained participants are enrolled in sensing tasks with Commercial-Off-The-Shelf (COTS) mobile devices equipped with low-end sensors, leading to untrustworthy low-quality or outlier observations due to unintentional mistakes (e.g., keystroke errors, misplaced decimal points, or wrong data representation) or device limitations. To make it worse, it is hard to know to what extend the sensory data are contaminated. The regression estimator should work reliably to derive accurate models with low-quality data. *2) Evolutionary mechanism:* observations, which are collected in an opportunistic fashion when cheap wireless communication is available or when some specific conditions occur, are naturally non-stationary, which means that both the distribution of observations, and the possibility an outlier occurs might change over time. The estimator should be able to deal with ever-coming and ever-changing observations and take an evolutionary methodology to develop models over time. *3) Strong privacy preservation:* as observations are often obtained through mobile devices, they are highly related with the private and sensitive information of mobile device users (e.g., current location, health status, etc.). The regression estimator should strongly preserve the privacy of MCS participants by keeping raw observations locally stored and processed. *4) Ultralow overheads:* mobile devices are key to MCS regression tasks, where they are not only involved in sensing tasks but also take part in regression modeling, but they are also resource constrained in terms of power, computational and communication capabilities. The estimator should take the limits of mobile devices into consideration while conducting regression modeling.

In the literature, regression problems with outliers and with privacy-preservation are investigated separately. For example, several secure regression methods [3], [5], [6] have been proposed for mining distributed datasets or for crowd-sourced systems, in which high-quality data are assumed. In contrast, a number of schemes [7], [8] have been proposed for outlier detection and diagnosis but without considering the data privacy issue. Recently, an outlier-tolerant blind regression scheme, PURE, is proposed [10]. However, PURE fails to consider the opportunistic and non-stationary features of MCS data. For PURE, well-estimated models cannot be updated with new coming data. In order to keep up-to-date, new models have to be re-estimated. As a result, to the best of our knowledge, there exists no successful solution, to tackling regression tasks in MCS settings.

In this paper, we consider typical MCS scenarios where

Shan Chang is the corresponding author.

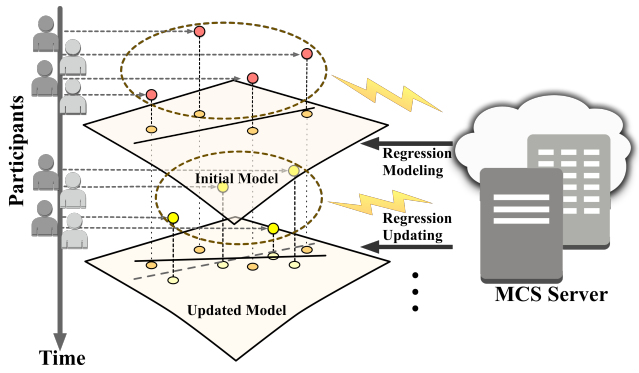



Fig. 1. Illustration of Lotus used in mobile crowd sensing applications.

sensory data are collected with mobile devices of unprofessional participants who can communicate with the MCS server via WiFi or 3G/4G. A regression estimator is proposed, called *Lotus*, that can be implemented as a set of protocols running between a server and mobile devices of participants. To deal with opportunistic and non-stationary observations, as illustrated in Figure 1, Lotus estimates a model which can evolve with newly collected observations periodically. More specifically, to estimate an initial model, we propose a *maximum-safe-subset selection problem*. By solving the problem, an optimal 'safe' (not-likely-to-be-outlier) subset of observations distributed over all participants can be determined for estimating a rough global regression model. Considering finding such a maximum-safe-subset is NP-hard, a distributed greedy hill-climbing algorithm is proposed, where a Mahalanobis distance-based selection protocol is carried out to decide a primary 'safe' subset of observations by which a model is estimated as the starting point of hill-climbing, and then this model is optimized as observations outside the 'safe' subset, which support the model more strongly, being swapped in the 'safe' subset iteratively, until no observations can be exchanged. After that, by examining the validity of local observations with this rough estimate, more valid observations can be identified, and are utilized to refine the global estimate. Moreover, the model evolves once enough new observations are available. During model evolution, a 'safe' subset of fresh observations is first determined in the same way as adopted in estimating the initial model, and then combined with the current model. With the new rough model, each participant re-examines the validity of both new observations and those observations having been involved in the initial model already, as well.

One main challenge is to achieve reliable model estimates on contaminated observations with the constraint of privacy-preservation. On one hand, preserving data privacy forbids raw data sharing with others including the server. On the other hand, without acquiring the raw data, it is hard to identify and to remove outliers, which makes the regression result unreliable. Lotus tackles this challenge through three key strategies: 1) participants in Lotus not only take part in collecting observations but also collaborate with each other and with the server in making decisions; 2) in Lotus, a one-time pad masking mechanism is introduced, such that secure aggregation can be achieved; 3) a blocking scheme is proposed to enable 'blind' optimization of 'safe' subset. Another challenge is to determine which observations are out-of-date and which are effective in updating the model. Furthermore, how to eliminate the impact of those out-of-date observations from the new model and how to affect the new model with new effective observations in an incremental way are of great difficulty. In Lotus, during model updating, the maximum-safe-subset of new observations will be included in the model, which guarantees that new observations indeed take effect on the new model. While out-of-data observations in the old model will be re-estimated according to the new model, such that those deviate from the new model will be excluded.

The main advantages of Lotus are four-fold. First, Lotus is resistant to high break-down outliers. Second, the confidentiality of raw observations is strongly protected. Third, regression model can be incrementally updated and can evolve over time. Last but not least, Lotus is a lightweight protocol tailored for mobile devices. We prove that Lotus can protect private data from being spied and can defend against collusion attacks theoretically. We evaluate the performance of Lotus through extensive trace-driven simulations using both real and synthetic datasets. The results demonstrate the effectiveness of Lotus to model updating, and the robustness even in the presence of 40% outliers.

The remainder of this paper is organized as follows. In Section II, we introduce problem formulation and preliminaries. In Section III, we elaborate the design of Lotus in detail. Section IV presents the analysis on defending against recovery and collusion attacks. In Section V, the accuracy and robustness of Lotus are examined. We review related work in Section VI. Finally, we conclude and outline the directions for future work in Section VII.

## II. Problem Formulation and Preliminaries

### A. Privacy and Threat Model

The main privacy leakage concerns during regression come from inside adversaries which take part in the estimating and updating procedures. More specifically, the MCS server and participants in collaborative regression are included. We characterize adversaries as follows.

- **Honest-but-curious:** both MCS server and participants are considered as 'passive' adversaries which follow the semi-honest model. It means that they execute the pre-designed protocols honestly but are curious about the private sensory data of others and attempt to learn or infer such information as much as possible.
- **Collusive:** participants may also mutual collude (or with the MCS server) to share knowledge in order to reveal more private information. However, we assume that the number of participants in collusion is limited.

### B. Multivariate Linear Regression

In MCS, the basic multivariate linear regression problem refers to a set of participants $\mathbb{N} = \{N_1, N_2, \ldots, N_m\}$ and a

sever $S$. $N_i (i = 1, 2, \ldots, m)$ collects a number of its own observations, each of them relates to $p$ ($p > 1$) independent variables $x_1, x_2, \ldots, x_p$ and a dependent variable $y$. The $j$-th observation $\boldsymbol{o}_j^{(i)}$ of $N_i$ is a vector of $[x_{j,1}^{(i)}, x_{j,2}^{(i)}, \ldots, x_{j,p}^{(i)}, y_j^{(i)}]$. $S$ gathers observations from participants in $\mathbb{N}$, to illuminate underlying association between variables, by fitting a model to observations. We have the definition as follows:

**Definition 1**. *A multivariate linear regression model* in MCS relates to observed independent and dependent variables, i.e., $\boldsymbol{x}^{(i)} = [\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \ldots, \boldsymbol{x}_{n_i}^{(i)}]^T$and $\boldsymbol{y}^{(i)} = [y_1^{(i)}, y_2^{(i)}, \ldots, y_{n_i}^{(i)}]^T$ from $N_i$ for $i = 1, 2, \ldots, m$, where $n_i$ represents the number of observations of $N_i$, and $\boldsymbol{x}_j^{(i)} = [1, x_{j,1}^{(i)}, x_{j,2}^{(i)}, \ldots, x_{j,p}^{(i)}]$, such that

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}]^T$, $\boldsymbol{Y} = [\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(m)}]^T$, $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_p]^T$ is the coefficient vector of regression model, $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_{\varkappa}]^T$,(where $\varkappa = \sum_{i=1}^{m} n_i$), represents random errors with *zero* expectation normal distribution.

A classic estimator is LS, which minimizes the sum of squared residuals, i.e., Residual Sum of Squares (RSS), and leads to the estimated value of unknown $\boldsymbol{\beta}$ as

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

$$\boldsymbol{u} = \sum_{i=1}^{m} (\boldsymbol{x}^{(i)})^T\boldsymbol{x}^{(i)}, \boldsymbol{v} = \sum_{i=1}^{m} (\boldsymbol{x}^{(i)})^T\boldsymbol{y}^{(i)}$$

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{(u)^{-1}v} \tag{2}$$

RSS is calculated as $R_{ss} = \sum_{i=1}^{m} (\boldsymbol{e}^{(i)})^T\boldsymbol{e}^{(i)}$, where $\boldsymbol{e}^{(i)} = [e_1^{(i)}, e_2^{(i)}, \ldots, e_{n_i}^{(i)}]^T$, and $e_j^{(i)}$, computed by $y_j^{(i)} - \boldsymbol{x}_j^{(i)}\widehat{\boldsymbol{\beta}}$, is the residual of $\boldsymbol{o}_j^{(i)}$ to $\widehat{\boldsymbol{\beta}}$.

Notice that both $(\boldsymbol{x}^{(i)})^T\boldsymbol{x}^{(i)}$ and $(\boldsymbol{x}^{(i)})^T\boldsymbol{y}^{(i)}$ can be computed by each $N_i$ locally and submitted to $S$ for calculating $\boldsymbol{\beta}$, without leaking the original $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$ to $S$.

Sharing aggregated results, however, should also be very carefully, since abuse of aggregated results is vulnerable to observations recovery attacks.

### C. Blind Regression with High Break-down Outlier Resistance

An observation is considered to be a *regression outlier* if it deviates from the relation followed by the majority of the data. We do not restrict the fraction of outliers in observations from certain individual. However, given a set of observations for regression modeling, the total outliers should be limited up to 50%; otherwise, it is impossible to achieve a reasonable regression model.

**Property 1 (Blind).** A regression modeling is *blind* if the raw observations cannot be obtained or inferred by any others (the server) during the regression.

**Property 2 (High Break-down Outlier Resistant).** A regression modeling is called *high break-down outlier resistant* method if the derived relation still fits the majority of data even if the portion of regression outliers reaches up to 50% of all observations.

**Definition 2.** The problem of *blind regression with high break-down outlier resistance* is referred to as, given the private observations, finding the optimal linear regression estimate so that it satisfies both Property 1 and 2.

### D. Evolutionary Regression with Non-stationary Observations

In MCS, a regression model is updated (evolved) periodically or once enough fresh observations are available. During model updating, fresh observations are added into the current model for improving model accuracy.

**Property 3 (Adaptive).** Model updating is called *adaptive* if it can accommodate to the gradual changes of dependency between the predictor and criterion variables, resulting from the non-stationarity of observations.

**Definition 3.** The problem of *evolutionary regression with non-stationary observations* is referred to as, given a group of new observations, updating a regression model (i.e., $\boldsymbol{\beta}$) without re-estimating from scratch, satisfying Property 3.

Remarks. *Model updating should be also blind and high break-down outlier resistant.*

### E. Mahalanobis Distance

The Mahalanobis distance [12] of an observation $\boldsymbol{o_i}$=$[x_{i,1}, \ldots, x_{i,p}, y_i]$, from a set of observations $\boldsymbol{O}$ with mean value $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_p, \mu_{p+1}]$ and covariance matrix $V$, is defined as:

$$d_M(\boldsymbol{o}_i) = \sqrt{(\boldsymbol{o}_i - \boldsymbol{\mu})^T V^{-1} (\boldsymbol{o}_i - \boldsymbol{\mu})} \tag{3}$$

$\boldsymbol{o}_i$ with greater $d_M(\boldsymbol{o}_i)$ from the rest of the observations is said to have higher leverage, and is suspected as an outlier, since it has a greater influence on the coefficients of the regression equation.

### F. Design Goals and Challenges

We aim to develop a practical method to address the problems of Definition 2 and 3, simultaneously, which is however very hard. The raw observations are not available due to the data confidentiality consideration, which obstructs outlier identification. To design a regression estimator satisfying privacy preservation and outlier resilience is nontrivial. Furthermore, model evolving should be adaptive to non-stationary observations. The current dependency between variables should be captured precisely. It implies the necessity to withdraw outliers or outdated observations from the new model. Without the prior knowledge on observation distributions, it's difficult to identify those 'bad' observations. Unfortunately, privacy concerns make the problem much harder.

Additionally, in MCS scenarios, typical mobile devices have relatively weak computation abilities and limited power. It is necessary that methodologies for evolutionary regression-estimating and privacy-preserving are lightweight. Especially, we desire an incremental model updating scheme, to maximize the use of existing computational results without re-estimating from scratch, and a privacy-preserving scheme free of complicated encryption schemes (e.g., homomorphic encryption), which induce high transmission and computation cost.

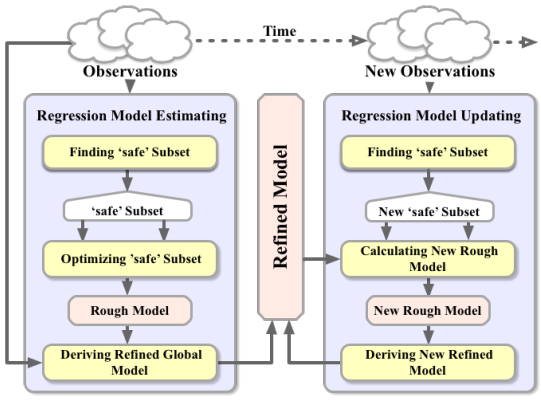



Fig. 2. Overview of Lotus

## III. Design of Lotus

### A. Overview

Lotus incorporates two techniques, i.e., blind regression model estimating and blind regression evolution, as illustrated in Figure 2.

**Blind Regression Model Estimating.** The basic idea to derive an initial model estimate has two steps. First, considering that outlier ratio is smaller than 50%, we wish to choose a 'safe' subset with half observations, according to which, a regression model can be estimated with minimum RSS, which implies that observations in selected 'safe' subset obey the same trend. In other words, a 'tight' and 'safe' regression model is estimated and used as a primary model. However, finding such an optimal subset is nontrivial, and proven NP-hard. Thus we develop a two-stage heuristic approach to achieve an approximation of optimal solution. *1) Initializing regression model over 'safe' subset:* the server collects and aggregates the information of some statistics (e.g., the mean) of local observations from all participants to get global statistics. Such global statistics are then distributed to all participants for data quality checking. Half of observations (determined in a distributed way) are considered as an initial 'safe' subset and used to conduct regression estimation. *2) Estimating rough global model with optimized 'safe' subset:* we design a greedy hill-climbing algorithm to optimize the 'safe' subset using the initial 'safe' observations as starting point. *3) Deriving refined global model:* by checking the data quality again using the rough global estimate, more valid local observations can be found and used to refine the rough model, resulting the ultimate initial global model.

**Blind Regression Evolution.** This technique is used for model updating, and has two functions. *1) Calculating new rough model:* when updating one model, a new 'safe' subset is formed according to a fresh observation group, which follows the same procedure as in finding the initial 'safe' subset. Then a new rough model can be built by including the new 'safe' subset into the current model. *2) Deriving new refined model:* after establishing the new rough model securely, each participant checks the quality of fresh observations, and non-outliers which had been used to build the current model, based on the new rough model locally, to decide which observations should be involved into or cut out from the new model, respectively.

In the above stages, in order to protect data privacy, each participant prepares its local aggregated results used for model estimating, and masks them with particular random values. Since those masks from different participants can cancel each other on the server side, upon receiving the masked local aggregations, the server can estimate a model precisely. Moreover, a blocking scheme is proposed to optimize the 'safe' subset in a secure manner.

### B. Blind Regression Model Estimating

*1) Initializing Regression Model over 'Safe' Subset:* Suppose $N_i$ $(1 \leqslant i \leqslant m)$ holds a set of observations $\boldsymbol{o}^{(i)} = \{\boldsymbol{o}_1^{(i)}, \boldsymbol{o}_2^{(i)}, \ldots, \boldsymbol{o}_n^{(i)}\}$, where $\boldsymbol{o}_j^{(i)}$ indicates the *j*-th observation of $N_i$. For the convenience of expression, we set the size of $\boldsymbol{o}^{(i)}$ as $n$ (which is known by $S$). Actually, Lotus can be easily extended to fit a general case where the size of each $\boldsymbol{o}^{(i)}$ might not be equal. Observations from the *m* participants are denoted by $\boldsymbol{O} = \bigcup_{i=1}^{m} \boldsymbol{o}^{(i)}$. We utilize $\lceil \frac{m \times n}{2} \rceil$ observations from $\boldsymbol{O}$ with smallest Mahalanobis distances, i.e., $d_M$, to form an initial 'safe' subset which is presumably free of outliers.

Taking into account privacy issues, each $d_M$ should be calculated by the corresponding observer locally, and the mean value $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p, \mu_{p+1})$ and covariance matrix $\boldsymbol{V}$ of $\boldsymbol{O}$ should be available to participants. To this end, the following protocol is performed between each $N_i$ and the server $S$ to select the 'safe' subset.

- *Calculating $\boldsymbol{\mu}$:* each $N_i$ computes the sum of each column in $\boldsymbol{o}^{(i)}$ locally, i.e., $\boldsymbol{s}^{(i)} = (\sum_{j=1}^{n} x_{j,1}^{(i)}, \ldots, \sum_{j=1}^{n} x_{j,p}^{(i)}, \sum_{j=1}^{n} y_j^{(i)})$, and then $N_i$ sends $\boldsymbol{s}^{(i)}$ to $S$. After gathering $\boldsymbol{s}^{(i)}$ from all participants, it's convenient for $S$ to calculate $\boldsymbol{\mu} = \frac{1}{m \times n} \sum_{i=1}^{m} \boldsymbol{s}^{(i)}$. Then $S$ sends $\boldsymbol{\mu}$ back to each $N_i$.
- *Calculating $\boldsymbol{V}$:* each $N_i$ computes $\boldsymbol{V}^{(i)} = \sum_{j=1}^{n} (\boldsymbol{o}_j^{(i)} - \boldsymbol{\mu})^T (\boldsymbol{o}_j^{(i)} - \boldsymbol{\mu})$ (which is a $(p+1) \times (p+1)$ symmetric matrix), and submits it to $S$. $S$ computes $\boldsymbol{V}$ by using

$$\boldsymbol{V} = E(\sum_{i=1}^{m} \boldsymbol{V}^{(i)}) = \frac{1}{m \times n} \sum_{i=1}^{m} \boldsymbol{V}^{(i)}$$

- *Calculating $\boldsymbol{V}^{-1}$:* $S$ calculates $\boldsymbol{V}^{-1}$, the inverse of $\boldsymbol{V}$, and sends it to all participants.
- *Calculating $d_M$:* $N_i$ computes $d_M(\boldsymbol{o}_j^{(i)})$ according to (3), and sends the median, i.e., $\mathcal{M}_{d_M}^{(i)}$, to $S$.
- *Estimating the global median of $d_M$:* $S$ sorts all $d_M(\boldsymbol{o}_j^{(i)})$ received, and broadcasts the median $\mathcal{M}_{d_M}$ (which is the global median) to participants.
- *Preparing local initial 'safe' subset:* $N_i$ selects those $\boldsymbol{o}_j^{(i)}$ whose $d_M$ are smaller than $\mathcal{M}_{d_M}$ to form a local 'safe' set $\boldsymbol{o}^{(c_i)} = \{\boldsymbol{o}_1^{(c_i)}, \boldsymbol{o}_2^{(c_i)}, \ldots, \boldsymbol{o}_{\xi_i}^{(c_i)}\}$. $N_i$ computes $(\boldsymbol{x}^{(c_i)})^T(\boldsymbol{x}^{(c_i)})$, $(\boldsymbol{y}^{(c_i)})^T(\boldsymbol{y}^{(c_i)})$ and $(\boldsymbol{x}^{(c_i)})^T\boldsymbol{y}^{(c_i)}$, and submits the results to $S$.
- *Regression over initial 'safe' subset:* $S$ computes $\boldsymbol{u}_0^{(c)} = \sum_{i=1}^{m} (\boldsymbol{x}^{(c_i)})^T(\boldsymbol{x}^{(c_i)})$, $\boldsymbol{v}_0^{(c)} = \sum_{i=1}^{m} (\boldsymbol{x}^{(c_i)})^T\boldsymbol{y}^{(c_i)}$, and

$\boldsymbol{w}_0^{(c)} = \sum_{i=1}^{m} (\boldsymbol{y}^{(c_i)})^T (\boldsymbol{y}^{(c_i)})$. Then, it's convenient for $S$ to estimate a regression model with coefficient vector $\widehat{\boldsymbol{\beta}}_0$ according to (2). Furthermore, $S$ calculates the corresponding Residual Sum of Squares, i.e., $Rss_{(0)}$, which equals to $\boldsymbol{w}_0^{(c)} - (\widehat{\boldsymbol{\beta}}_0)^T \boldsymbol{v}_0^{(c)}$.

**Theorem 1.** *In order to defend against $\boldsymbol{o}^{(\boldsymbol{c})}$ (we use $\boldsymbol{o}^{(\boldsymbol{c})} = \bigcup_{i=1}^{m} \boldsymbol{o}^{(c_i)}$) from been recovery attacks by $S$ or other attackers, the number of all observations collected in each period should be no less than $2p + 3$.*

*Proof.* An attacker aiming to recover $\boldsymbol{o}^{(c)}$ considers the problem as solving a set of equations, related to the corresponding $\lceil \frac{m \times n}{2} \rceil \times (p+1)$ unknown variables. According to the protocol, $S$ obtains $\sum_{i=1}^{m} (\boldsymbol{x}^{(c_i)})^T (\boldsymbol{x}^{(c_i)})$, $\sum_{i=1}^{m} (\boldsymbol{y}^{(c_i)})^T (\boldsymbol{y}^{(c_i)})$ and $\sum_{i=1}^{m} (\boldsymbol{x}^{(c_i)})^T \boldsymbol{y}^{(c_i)}$, which refer to $\frac{(p+1)(p+2)}{2} - 1$, 1 and $p+1$ equations related to $\boldsymbol{o}^c$, respectively. It is essential that the number of variables should be larger than that of the corresponding equations, otherwise, $\boldsymbol{o}^{(c)}$ could be recovered. Additionally, $\boldsymbol{o}^{(\boldsymbol{c})}$ have at least $p+1$ observations in order to estimate a regression. Thus, the following equality holds,

$$\lceil (m \times n)/2 \rceil \times (p+1) > (p+1) + (p+1)(p+2)/2 \quad (4)$$

$$\lceil (m \times n)/2 \rceil \geq p + 1 \quad (5)$$

So we set the number of all observations $m \times n \geq 2p+3$, which meet the above condition and this concludes the proof. ■

In this phase, $N_i$ needs to share $\mathcal{M}_{d_M}^{(i)}$, along with a number of aggregated results, i.e., $\boldsymbol{s}^{(i)}$, $\boldsymbol{V}^{(i)}$, $(\boldsymbol{x}_c^{(i)})^T (\boldsymbol{x}_c^{(i)})$ and $(\boldsymbol{x}_c^{(i)})^T \boldsymbol{y}_c^{(i)}$, relating to its raw observations $\boldsymbol{o}^{(i)}$, which may cause privacy problem if the number of variables in $\boldsymbol{o}_c^{(i)}$ is no more than that of the equations built from those aggregated results.

According to the protocol, the server adds up those aggregated values from all participants together for further processing, without knowing individual ones. It implies that the sum of certain kind of aggregate values should be computed in a secure fashion. We use a *one-time pad masking* technique for securely calculating the summation of aggregated values without disclosing individual ones. We explain the key idea by calculating $\sum_{i=1}^{m} \boldsymbol{s}^{(i)}$ as follows:

The basic idea of masking is that each $\boldsymbol{s}^{(i)}$ is masked with certain secret vector, such that all masks can be canceled when they are added on the server side. Suppose each pair of participants $(N_i, N_k)$ agrees on certain random vector $b_{i,k}$. If $N_i$ adds this to $\boldsymbol{s}^{(i)}$, while $N_k$ will subtract it from $\boldsymbol{s}^{(k)}$. In the design, each participant $N_i$ computes:

$$A_i = \boldsymbol{s}^{(i)} + \sum_{i<k} b_{i,k} - \sum_{i>k} b_{k,i}$$

and sends $A_i$ to the server, and the server computes:

$$A = \sum_{i=1}^{m} A_i = \sum_{i=1}^{m} (\boldsymbol{s}^{(i)} + \sum_{i<k} b_{i,k} - \sum_{i>k} b_{k,i}) = \sum_{i=1}^{m} \boldsymbol{s}^{(i)}$$

In order to avoid exchanging random vectors $b_{i,k}$ between participants, which requires quadratic communication overhead, each pair of participants $(N_i, N_k)$ shares a secret as the common seed in advance, such that the same pseudorandom can be generated by both parties.

To this end, we assume that each $N_i$ holds a pair of public and secret keys $(pk_i, sk_i)$, which can be achieved by commonly used Public Key Infrastructure (PKI). The server maintains a public key list of participants. Once $N_i$ registers with the server, $pk_i$ will be added into the list, and the updated list will be published to all participants. Then, for each $pk_k$ in the list (except $pk_i$), $N_i$ generates a secret $s_{i,k}$, and encrypts it with $pk_k$, and submits the ciphertext to $N_k$ (directly or relay by the server). $N_k$ decrypts the ciphertext using $sk_k$ and gets $s_{i,k}$. After that, $b_{i,k}$ can be generated based on $s_{i,k}$. A simple way is to apply a secure hash function $h(\cdot)$ (which is known to all participants) on $s_{i,k}$, i.e., $b_{i,k} = h(s_{i,k})$. In consideration of security, $b_{i,k}$ will only be used once. For updating $b_{i,k}$, new mask $b'_{i,k}$ is calculated as $h(b_{i,k})$.

*2) Estimating Rough Global model with Optimized 'Safe' Subset:* First, we give the following definitions:

**Definition 4.** The problem of *finding optimal 'safe' subset* is referred to as, given a set $\mathcal{N}$ of $n$ observations, select $\lceil \frac{n}{2} \rceil$ observations to fit a regression model, to minimize corresponding RSS.

However, above problem is NP-hard, since it can be easily translated to the *top-k nodes problem* [13] which has been proven to be NP-hard. Therefore, we use a greedy hill climbing algorithm to approximate the optimum solution.

**Definition 5.** Given $\widehat{\boldsymbol{\beta}}$ estimated by using $n$ observations, with corresponding RSS denoted as $Rss$, *Moderate Residual Expectation (MRE)* is referred to as $\sqrt{Rss}/n$.

We use $\boldsymbol{o}^{(\boldsymbol{c})} = \bigcup_{i=1}^{m} \boldsymbol{o}^{(c_i)}$, leading to $\widehat{\boldsymbol{\beta}}_0$, as the starting point of hill climbing. Then, we check the residuals of remaining observations. Those observations whose residuals are smaller than the MRE of current regression estimation will be included into the 'safe' subset while the same number of 'safe' observations with largest residuals will be removed from it, which helps to shrink the RSS. Thus, $\widehat{\boldsymbol{\beta}}_0$ is optimized towards minimizing RSS. 'Uncertain' and 'safe' observations are swapped in and out of the model iteratively until no observations can be swapped into the model. To this end, the following protocol is performed between each $N_i$ and the server $S$ to optimize the 'safe' subset.

First, $S$ calculates MRE, denoted as $Mre_{(0)}$, and broadcasts $\widehat{\boldsymbol{\beta}}_0$ and $Mre_{(0)}$ to all participants.

Second, $N_i$ calculates residual of each $\boldsymbol{o}_j^{(i)}$ to $\widehat{\boldsymbol{\beta}}_0$, i.e., $e_j^{(i)} = y_j^{(i)} - \boldsymbol{x}_j^{(i)} \widehat{\boldsymbol{\beta}}_0$, and compares those residuals with $Mre_{(0)}$. Notice that $\boldsymbol{o}^{(i)}$ is divided into local 'safe' subset $\boldsymbol{o}^{(c_i)}$ and remaining subset $\boldsymbol{o}^{(r_i)}$. Thus, those $\boldsymbol{o}_j^{(r_i)}$ whose residuals are smaller than $Mre_{(0)}$ is formed a swap-in subset $\boldsymbol{o}^{(in_i)}$, and submits the size of $\boldsymbol{o}^{(in_i)}$, i.e.,$|\boldsymbol{o}^{(in_i)}|$ to $S$ by using one-time pad masking. .

Third, in order to decide which observations should be removed from $\boldsymbol{o}^{(c_i)}$, a naive solution is to submit $\boldsymbol{e}^{(c_i)}$ to $S$.

After receiving all $\boldsymbol{e}^{(c_i)}$ and $|\boldsymbol{o}^{(in_i)}|$, $S$ computes $in_{num} = \sum_{i=1}^{m} |\boldsymbol{o}^{(in_i)}|$, the total number of observations should be added into 'safe' subset, and sorts all residuals of 'safe' observations, and notifies $N_i$ with threshold of residuals $e_{trsd}$ such that $in_{num}$ observations whose residuals exceed $e_{trsd}$ should be removed from the 'safe' subset.

Fourth, on the side of $N_i$, according to $e_{trsd}$, a swap-out subset $\boldsymbol{o}^{(out_i)}$ can be formed, which is composed of those observations satisfying $\boldsymbol{e}_j^{(c_i)} > e_{trsd}$. Then $N_i$ calculates $(\boldsymbol{x}^{(in_i)})^T(\boldsymbol{x}^{(in_i)}) - (\boldsymbol{x}^{(out_i)})^T(\boldsymbol{x}^{(out_i)})$, $(\boldsymbol{y}^{(in_i)})^T(\boldsymbol{y}^{(in_i)}) - (\boldsymbol{y}^{(out_i)})^T(\boldsymbol{y}^{(out_i)})$ and $(\boldsymbol{x}^{(in_i)})^T\boldsymbol{y}^{(in_i)} - (\boldsymbol{x}^{(out_i)})^T\boldsymbol{y}^{(out_i)}$, and submits the results, i.e., $|\boldsymbol{o}^{(in_i)}|$, to $S$, by using one-time pad masking.

Consider that

$$\begin{aligned}\boldsymbol{u}_1^{(c)} &= \boldsymbol{u}^{(c)} + \sum_{i=1}^{m}((\boldsymbol{x}^{(in_i)})^T(\boldsymbol{x}^{(in_i)}) - (\boldsymbol{x}^{(out_i)})^T(\boldsymbol{x}^{(out_i)})) \\ &= \boldsymbol{u}^{(c)} + \sum_{i=1}^{m}(\boldsymbol{x}^{(in_i)})^T(\boldsymbol{x}^{(in_i)}) - \sum_{i=1}^{m}(\boldsymbol{x}^{(out_i)})^T(\boldsymbol{x}^{(out_i)})\end{aligned}$$

Similarly, $\boldsymbol{v}_1^{(c)}$ and $\boldsymbol{w}_1^{(c)}$ are computed. Thus $\widehat{\boldsymbol{\beta}}_1 = (\boldsymbol{u}_1^{(c)})^{-1}\boldsymbol{v}_1^{(c)}$, and $Rss_{(1)} = \boldsymbol{w}_1^{(c)} - (\widehat{\boldsymbol{\beta}}_1)^T\boldsymbol{v}_1^{(c)}$.

Repeat the above steps until no observations can be swapped. Then regression estimate $\widehat{\boldsymbol{\beta}}_{safe}$ obtained in the last round is considered as a rough global regression model.

However, publishing residuals of certain observations to different models gives the chance for attackers to launch observation recovery attacks. One solution to solve the problem is to conduct sorting on ciphertext, e.g. Garble Circuit, which however induces high computation and communication costs. We design a light-weight and effective scheme by loosening the accuracy of $\boldsymbol{o}^{(out)}$, i.e. $\bigcup_{i=1}^{m}\boldsymbol{o}^{(out_i)}$, slightly.

$S$ publishes a set of bins such that a residual can fall into one and only one bin. Furthermore, the size of the bins are different, which satisfies that bigger residuals will be thrown into smaller bins. Each $N_i$ has the knowledge of those bins, hence can decide the number of observations in $\boldsymbol{o}^{(out_i)}$ falling into each bin locally, and submits those numbers to $S$ by using masking. Thus $S$ can calculate the total number of observations falling into each bin. Since $S$ knows the $in_{num}$, which is equal to $out_{num}$, and follows the rule of small-bin-first, i.e. observations in smaller bins well be removed first. $S$ decides the threshold bin $bin_{trsd}$ such that all observations in smaller and bigger bins will be removed and kept, respectively. While partial observations in $bin_{trsd}$ will be removed. Then $S$ broadcasts the $bin_{trsd}$ and the ratio of the observations $rmo_{ratio}$ in it. According to this, $N_i$ selects all observations whose bins are smaller than $bin_{trsd}$, and random selects observations in $bin_{trsd}$ with a probability of $rmo_{ratio}$ to form local swap-out subset $\boldsymbol{o}^{(out_i)}$.

*3) Deriving Refined Global Model:* After achieving the rough model $\widehat{\boldsymbol{\beta}}_{safe}$, $S$ broadcasts $\widehat{\boldsymbol{\beta}}_{safe}$ and root mean squared error $RMse$ ($RMse = \sqrt{\frac{Rss}{(n\times m-p-1)}}$) to all participants. Then the following refining protocol is carried out between $S$ and each $N_i$, in which the outlyingness of observations in $\boldsymbol{o}^{(r_i)}$ (i.e., 'uncertain') is tested in accordance with $\widehat{\boldsymbol{\beta}}_{safe}$. Those observations fitting $\widehat{\boldsymbol{\beta}}_{safe}$ well will be added into achieve an initial estimate $\widehat{\boldsymbol{\beta}}_{ini}$.

$N_i$ calculates standardized residual $z_j^{(r_i)}$ of $\boldsymbol{o}_j^{(r_i)}$

$$z_j^{(r_i)} = |e_j^{(r_i)}|/RMse \tag{6}$$

$N_i$ goes through $\boldsymbol{o}^{(r_i)}$, and labels observations with $z_j^{(r_i)}$ exceeding 1.69 according to [11], which implies inconsistency with $\widehat{\boldsymbol{\beta}}_{safe}$, as outliers. Then, observations passing the test are formed as a refining subset $\boldsymbol{o}^{(f_i)}$, which will be added into the initial regression model. Specifically, $(\boldsymbol{x}^{(f_i)})^T(\boldsymbol{x}^{(f_i)})$ and $(\boldsymbol{x}^{(f_i)})^T\boldsymbol{y}^{(f_i)}$ are computed locally, and are submitted to $S$ by using masking.

$S$ computes $\boldsymbol{u}^{(ini)} = \boldsymbol{u}^{(c)} + \sum_{i=1}^{m}(\boldsymbol{x}^{(f_i)})^T(\boldsymbol{x}^{(f_i)})$ and $\boldsymbol{v}^{(ini)} = \boldsymbol{v}^{(c)} + \sum_{i=1}^{m}(\boldsymbol{x}^{(f_i)})^T\boldsymbol{y}^{(f_i)}$, $S$ estimates $\widehat{\boldsymbol{\beta}}_{ini} = (\boldsymbol{u}^{(ini)})^{-1}\boldsymbol{v}^{(ini)}$.

### C. Blind Regression Evolution

Under the current estimate $\widehat{\boldsymbol{\beta}}_{ini}$, assume that there exists a group of participants $\widetilde{\mathbb{N}} = \{\widetilde{N}_1, \widetilde{N}_2, \ldots, \widetilde{N}_q\}$, and each $\widetilde{N}_l$ $(l = 1, 2, \ldots, q)$ holds a set of new observations, i.e., $\widetilde{\boldsymbol{o}}^{(l)} = \{\widetilde{\boldsymbol{o}}_1^{(l)}, \widetilde{\boldsymbol{o}}_2^{(l)}, \ldots, \widetilde{\boldsymbol{o}}_n^{(l)}\}$. In order to update $\widehat{\boldsymbol{\beta}}_{ini}$ by utilizing new observations, the server $S$ first calculates a new rough model $\widetilde{\boldsymbol{\beta}}_{rgh}$ according to $\widehat{\boldsymbol{\beta}}_{rgh}$ and the new 'safe' subset, and then refines $\widetilde{\boldsymbol{\beta}}_{rgh}$ to achieve the new estimate.

*1) Calculating New Rough Model:* In this procedure, a new 'safe' subset is formed and included into $\widehat{\boldsymbol{\beta}}_{ini}$ to construct the new rough model $\widetilde{\boldsymbol{\beta}}_{rgh}$.

Specifically, $S$ and $\widetilde{N}_i$ cooperate with each other to select the new 'safe' subset (with a size of $\lceil\frac{q\times n}{2}\rceil$) from $\widetilde{\boldsymbol{o}}^{(l)}$, following the protocols which have been introduced in subsections III-B1 and III-B2. We denote the new 'safe' observation subset of $\widetilde{N}_l$ as $\widetilde{\boldsymbol{o}}^{(c_l)}$. Each $\widetilde{N}_l$ submits $\sum_{l=1}^{q}(\widetilde{\boldsymbol{x}}^{(c_l)})^T(\widetilde{\boldsymbol{x}}^{(c_l)})$ and $\sum_{l=1}^{q}(\widetilde{\boldsymbol{x}}^{(c_l)})^T\widetilde{\boldsymbol{y}}^{(c_l)}$with masking to $S$ to estimate a new rough model $\widetilde{\boldsymbol{\beta}}_{rgh}$.

*2) Deriving New Refined Model:* After updating the new rough model $\widetilde{\boldsymbol{\beta}}_{rgh}$, $S$ broadcasts it to participants in $\mathbb{U} = \widetilde{\mathbb{N}}\bigcup\mathbb{N}$. Then the following procedures are performed to establish a new model $\widetilde{\boldsymbol{\beta}}_{new}$.

Specifically, $S$ communicates with each participant in $\mathbb{U}$, to test the outlyingness of observations in $\boldsymbol{o}^{(c_i)}\bigcup\widetilde{\boldsymbol{o}}^{(\boldsymbol{r_l})}$, in accordance to $\widetilde{\boldsymbol{\beta}}_{rgh}$, which is alike the procedure obtaining $\boldsymbol{o}^{(f_i)}$ (described in subsection III-B3). Those observations in $\boldsymbol{o}^{(c_i)}$ which unfit $\widetilde{\boldsymbol{\beta}}_{rgh}$ will be removed from the new model. On the other side, observations in $\widetilde{\boldsymbol{o}}^{(\boldsymbol{r_l})}$ which pass the test will be added into the new model.

## IV. Security Analysis

### A. Observation Recovery Attacks

In the phase of initializing regression model over 'safe' subset, $S$ obtains $\boldsymbol{\mu}$ and $\boldsymbol{V}$ of all $m \times n$ observations from all participants for finding a 'safe' subset, which refer to $\frac{(p+1)(p+2)}{2} - 1$, and $p+1$ equations related to $\boldsymbol{O}$, respectively. Note that the size of $\boldsymbol{O}$ is $m \times n \geq 2p+3$ (which means the 'safe' subset have enough data to estimate a regression model).

Therefore, the number of variables is at least $(2p+3)(p+1)$. As $(2p+3)(p+1) > \frac{(p+1)(p+2)}{2} + p$ always holds, it means that $S$ does not have enough equations to recover $\boldsymbol{O}$. Similar method can prove that no one can recover the original data of the participant during estimating regression model over initial 'safe' subset. Additionally, privacy security during estimating a rough regression model and deriving refined global model could be ensured, since that observation swapped could be viewed as add and minus operations on a matrix to an existing matrix, it is impossible for $S$ to deduce any privacy information by the differences in two received matrices..

### B. Collusion Attacks

Malicious participants can collude to collect slices generated by $N_i$ in order to rebuild $(\boldsymbol{x}^{(i)})^T(\boldsymbol{x}^{(i)})$ and $(\boldsymbol{x}^{(i)})^T\boldsymbol{y}^{(i)}$, which can be used to recover the local observations of $N_i$. Using $(\boldsymbol{x}^{(i)})^T(\boldsymbol{x}^{(i)})$ as an example, with proposed masking technology, both in initial model estimating and updating, to recover $(\boldsymbol{x}^{(i)})^T(\boldsymbol{x}^{(i)})$, malicious participants have to know all $m-1$ masks hold by $N_j(j \neq i)$. But, the number of participants in collusion is limited in our assumption of threat model. So it is hard to recovery the raw observations from the aggregated data.

## V. Performance Evaluation

### A. Methodology

We examine the performance of Lotus via both real and synthetic datasets. We use four datasets, three are well-known and one is generated, which are described as follows:

1) *Airfoil self-noise* [14]: this dataset includes 1503 observations. We use attributes of *frequency*, *angle of attack*, *free-stream velocity* and *suction side displacement thicknessto* build the influence function of scaled sound pressure level.
2) *Concrete compressive strength* [15]: the dataset contains 1030 observations. We use attributes of *cement*, *blast furnace slag*, *fly ash* and *age* to build the influence function of the concrete compressive strength.
3) *Synthetic*: we generate 1400 observations by using $y = 5+5\sum_{i=1}^{9} x_i+\epsilon$, where $\epsilon \sim N(0,1)$ and $x_i \sim N(0,1)$.

Based on above datasets, we generate two kinds of noises. All the noises are random vectors and each dimension has independent and identical distributions.

1) *Random noise:* variables obey uniform distribution within the range $[0, v_{max} - v_{min}]$, where $v_{max}$ and $v_{min}$ refer to the maximum and the minimum measures of one certain attribute in certain dataset, respectively.
2) *Normal distributed noise with $N(\mu, \sigma^2)$:* variables obey normal distribution with a mean of $\mu$ and a standard deviation of $\sigma$, where $\mu$ and $\sigma$ are the estimates of the mean and standard deviation of one given attribute of certain dataset.

For each dataset, we generate outliers by adding certain noise on original observations under predetermined ratio.

We compare the performance of Lotus with the state-of-art LS estimator, and WLS estimator in which observations with larger residuals will receive smaller weight in order to reduce the influence of the suspected outliers in modeling. WLS leads to the estimated of $\boldsymbol{\beta}$ as

$$\widehat{\boldsymbol{\beta}}^W = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{Y}$$

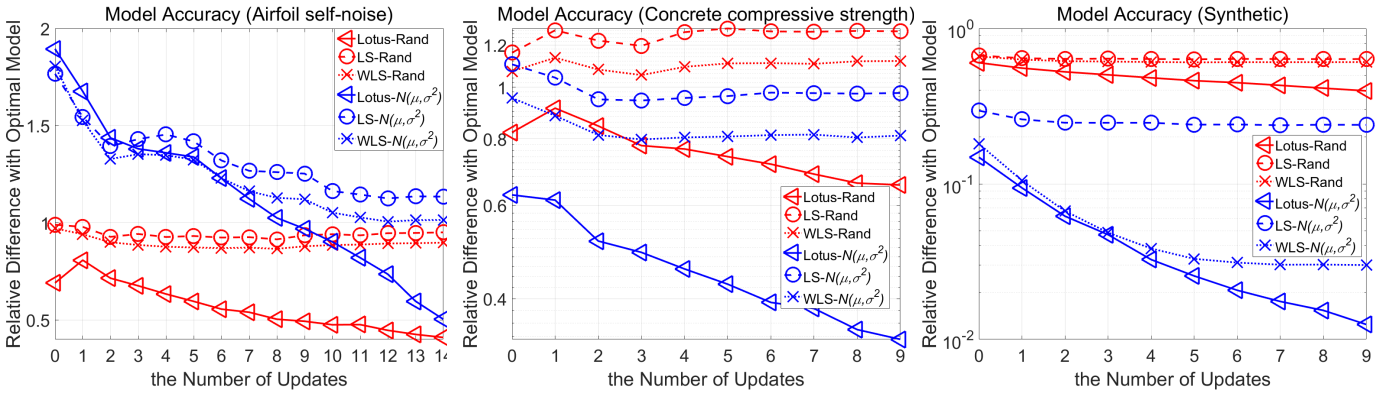
Weight matrix $\boldsymbol{W}$ is diagonal, where the $i$-th diagonal element refers to the influence of the $i$-th observations. In specific, $\boldsymbol{W}$ is initialized as an identity matrix, and is updated according to the current $\widehat{\boldsymbol{\beta}}^W$, and will be used for new model estimating.

By utilizing LS estimator, we can obtain the models to the original datasets, which are high quality, as the ground truths. We evaluate the accuracy of an estimate by calculating the relative difference of model coefficients between an estimated $\widehat{\boldsymbol{\beta}}$ and the ground truth $\boldsymbol{\beta}_*$, defined as $\frac{\|\boldsymbol{\beta}_* - \widehat{\boldsymbol{\beta}}\|}{\|\boldsymbol{\beta}_*\|}$. For each dataset, noise type and each simulation configuration, we run the simulation 20 times and get the average.

### B. Accuracy with Model Updating

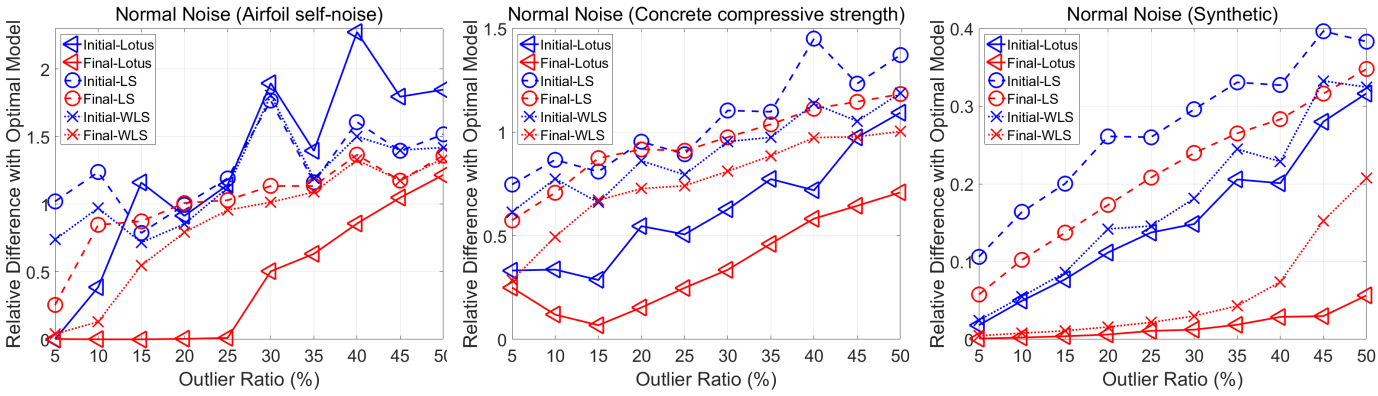
In this experiment, we examine the performance of Lotus on the four datasets, with more observations facilitating model updating. In order to simulate the periodical regression model updating in MCS, for each dataset, we randomly divide the observations into groups, each of them satisfies that at least $n \geq 50 + 8p$ observations are included, such that the number of observations for estimating a regression model is sufficient according to the suggestion from Green [16]. For *Airfoil self-noise*, *Energy efficiency*, *Concrete compressive strength* and *Synthetic*, the number of groups are 15, 6, 10 and 10, respectively. We then randomly pick up one unused group from one certain dataset for model estimating or updating. We compare Lotus with LS and WLS, under both the random and normal distributed noise settings with the outlier ratio $\varepsilon$ equal to $30\%$ in each observation group, which is considered to be practical.

Fig. 3 plots the relative difference of model coefficients between the estimated models and the global optimal models, estimated by LS according to all original observation groups (without noises) obtained from the beginning to the current updating period. The horizontal axis indicates the number of updates, where zero refers to the initial model without updating. We can see that, the accuracy of the model estimated by Lotus is continuously improved with new group of observations being included in model updating. It can also be seen that Lotus outwits LS and WLS in all settings of three real datasets. In the synthetic dataset, WLS can achieving a similar performance Lotus in the early rounds of updating, but Lotus refines the model much faster than WLS in the following rounds, and achieves a better estimate in the end. Moreover, with the model being updated, the performance gaps between Lotus and other two estimators become even larger.




(a) Airfoil self-noise (b) Concrete compressive strength (c) Synthetic

Fig. 3. Difference from the optimal model vs. the number of updates.




(a) Airfoil self-noise (b) Concrete compressive strength (c) Synthetic

Fig. 4. Difference from the optimal model vs. outlier ratio with normal noise under $N(\mu, \sigma^2)$.

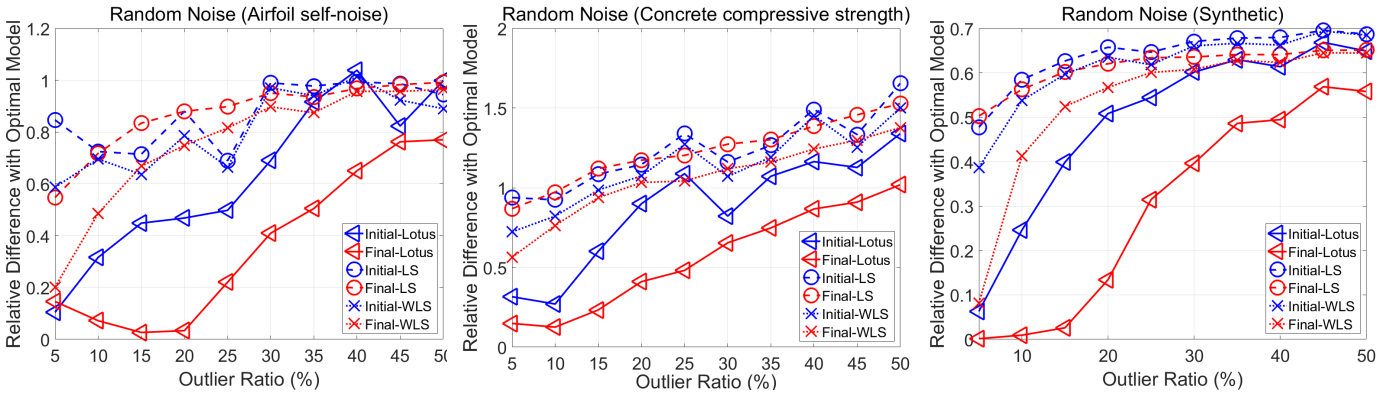
### C. Robustness under Different Outlier Ratios

In this experiment, we further examine the robustness of Lotus against outliers under different outlier ratios $\varepsilon$, and of different types. In specific, we check both the initial estimates, which are established based on the first groups of observations without any updating, and the final models which are updated for several times based on all observations in the corresponding datasets. For each setting of noise, the noise ratio $\varepsilon$ varies from $5\%$ to $50\%$ at an interval of $5\%$. We randomly divide the observations into groups as described before, for a group of $n$ observations, we generate $n*\varepsilon$ outliers according to given noise type, and random distribute them to observations in the group.

Fig. 4 and Fig. 5 plot the relative difference of model coefficients between the estimated models and the corresponding global optimal models as a function of outlier ratio under random and normal distribution noise settings, respectively. It can be seen that Lotus outwits LS and WLS in all settings. LS is very sensitive to outliers, even under a low outlier ratio of $5\%$, LS gets a bad performance, whereas, Lotus can hold good accuracy even when $\varepsilon$ increases to $40\%$, especially for the final estimates. In some cases, particularly in synthetic dataset, WLS can achieve a high accuracy of final models under low outlier ratio, however, the performance degrades dramatically as the outlier ratio increasing to $10\%$. Oppositely, Lotus is much more tolerant to outliers, and model updating can significantly improve the accuracy of estimates. Overall, we find that Lotus is robust to not only the number but also the randomness of outliers.

## VI. Related Work

Privacy-preserving data aggregation problems have been widely investigated in the literature. Most privacy-preserving aggregation methods, however, only focus on calculating simple aggregation functions such as *sum*, *mean* and *max/min*, which cannot be used for privacy-preserving regression estimation directly. Privacy-preserving outlier detection has been investigated in distributed systems [9]. Most methods deal with distanced-based outliers. However, the definition of outliers in regression is much more complex. Perturbation technologies are widely used for privacy protecting, where random noises are used to cover up private data. PoolView [17] protects privacy of stream data in participatory sensing. The core idea is to add the random noise with known distribution to raw data, after which a reconstruction algorithm is used to estimate the distribution of original data. The disadvantage of perturbation-based schemes is that additional noises introduced make regression estimates inaccurate.




(a) Airfoil self-noise (b) Concrete compressive strength (c) Synthetic

Fig. 5. Difference from the optimal model vs. outlier ratio with random noise.

Studies on constructing an accurate regression model with privacy-preserving are more related to this work. M-PERM is a mutual privacy-preserving regression modeling approach, where a series of data transformation and aggregation operations are operated at the participatory nodes to preserve data privacy [5]. However, it should be emphasized that these methods are all based on the LS estimator. The correctness of them relies on the assumption that original data are collected correctly without gross errors, i.e., outliers. Any incorrect observations may breakdown the estimation, since LS estimator is very sensitive to outliers. PURE, an outlier tolerant privacy-preserving regression scheme, proposed by S. Chang et al. is most relevant to our work [10]. However, PURE does not notice the opportunistic and non-stationary features of sensory data, and cannot deal with model updating. Furthermore, in PURE, model estimation requires a number of iterations, which is undesirable on energy-constrained mobile devices, due to high communication and computation cost.

## VII. Conclusion

In this paper, we have introduced Lotus, a scheme for regression and model updating with low-quality, private, opportunistic and non-stationary sensory data in MCS. Lotus provides strong protection on the confidentiality of raw sensory data by masking aggregated result with cancellable one-time pads, and by blind optimizing of 'safe' subset with a blocking scheme, and achieves good model accuracy under very low quality and non-stationary data, through identifying suspected outliers and withdrawing outdated observations during model updating. Both analysis and extensive simulation results demonstrate the efficacy of Lotus. In future work, we will build a prototype system of MCS in our campus, and further examine the feasibility of Lotus under real deployment.

## VIII. Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61672151, 61370205, 61772340, 61472255, 61420106010); by the Fundamental Research Funds for the Central Universities (Grant No. EG2018028); by Shanghai Rising-Star Program (Grant No.17QA1400100), by DHU Distinguished Young Professor Program.

## References

[1] J. Vanus, P. Valicek, T. Novak, R. Martinek, P. Bilik, and J. Zidek, "Utilization of regression analysis to increase the control accuracy of dimmer lighting systems in the Smart Home," in *IFAC-Papers OnLine*, 2016, pp. 517-522.

[2] H. J. Lee, R. B. Chatfield, and A. W. Strawa , "Enhancing the applicability of satellite remote sensing for PM2.5 estimation using MODIS deep blue AOD and land use regression in california, united states," in *Environmental Science & Technology*, 2016, pp. 6546-6555.

[3] H. Kikuchi, C. Hamanaga, H. Yasunaga, H. Matsui, and H. Hashimoto, "Privacy-preserving multiple linear regression of vertically partitioned real medical datasets," in *Proc. IEEE AINA*, 2017, pp. 1042-1049.

[4] Y. Gong, Y. Fang, and Y. Guo, "Private data analytics on biomedical sensing data via distributed computation," in *IEEE/ACM Trans. Comp. Bio. Bioinfo*, 2016, pp. 431-444.

[5] K. Xing, Z. Wan, P. Hu, H. Zhu, Y. Wang, X. Chen, Y.Wang, and L. Huang, "Mutual privacy-preserving regression modeling in participatory sensing," in *Proc. IEEE INFOCOM*, 2013, pp. 3039-3047.

[6] A. F. Karr, X. Lin, J. Reiter, and A. P. Sanil, "Secure regression on distributed databases," in *Computational and Graphical Statistics*, 2004, pp. 263-279.

[7] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine," in *Ad Hoc Networks*, 2013, pp. 1062-1074.

[8] A. D. Paola, S. Gaglio, G. L. Re, F. Milazzo, and M. Ortolani, "Adaptive Distributed Outlier Detection for WSNs," in *IEEE Trans. Cyb*, 2015, pp. 902-913.

[9] J. Vaidya and C. Clifton, "Privacy-Preserving Outlier Detection," in *Proc. IEEE ICDM*, 2004, pp. 233-240.

[10] S. Chang, H. Zhu, W. Zhang, L. Lu, and Y. Zhu, "PURE: blind regression modeling for low quality data with participatory sensing," in *IEEE Trans. Para. Distr. Sys*, 2016, pp. 1199-1211.

[11] M. Kutner, C. Nachtsheim, J. Neter, and W. Li, "Applied linear statistical models," in *McGraw-Hill*, 2005 .

[12] P. Mahalanobis, "On the generalised distance in statistics," in *Proc. NISI*, 1936, pp. 49C55.

[13] N. R. Suri and Y. Narahari, "Determining the top-k nodes in social networks using the Shapley value," inProc. IFAAMAS, 2008, pp. 1509-1512.

[14] The Airfoil self-noise dataset. [Online]. Availiable: https://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise

[15] Concrete compressive strength dataset. [Online]. Availiable: https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength.

[16] S. Green, "How many subjects does it take to do a regression analysis?," in *Multivariate Behavioral Research*, 1991, pp. 499-510.

[17] R. Ganti, N. Pham, Y. Tsai, and T. Abdelzaher, "PoolView: stream privacy for grassorts participatory sensing," in *Proc. ACM ENCC*, 2008, pp. 281-294.